Cypherium: A Scalable and Permissionless Smart Contract Platform

Draft v1.1
Sky Guo
contact@cypherium.io

Abstract. This paper presents consensus methods, including Proof-of-Work, fail to achieve sufficient transaction throughput to satisfy the requirements of a public blockchain, or an application that leverages one. This also includes the mainstream exchange of the digital tokens that secure them. Additionally, transactions cannot be considered irreversible until several blocks have been mined. We present Cypherium, which proposes a hybrid consensus mechanism wherein a dynamic group of replica nodes acts as validator committee to determine the validity and order of transactions within a Byzantine fault tolerance-based blockchain system. In place of a certificate authority, the system adopts Proof-of-Work to establish node identities and enable its open participation. Leader election and transaction validation are decoupled into two separate chains to eliminate transaction confirmation time. Transactions are permanently recorded once verified by more than two-thirds of the majority of members of the validator committee.

1. Introduction

Bitcoin revolutionized cryptocurrency, and Ethereum, the smart contract platform. We are using our later mover advantage to learn from the mistakes and growing pains other coins have experienced.

Both Bitcoin and Ethereum blockchains are based on a mathematical process called Proof-of-Work[1]. Proof-of-work serves both as a consensus mechanism within a decentralized network and as an incentive for nodes running it. While Proof-of-Work is a decentralized way of distributing tokens, it is not without its shortcomings. The most significant drawback is lack of scalability. As of June 2017, there are more than a quarter million unconfirmed transactions within the Bitcoin network mempool. Executing a single Bitcoin transaction can take between a few hours to a few days, with transaction fees averaging several dollars and rising with the increasing demand of the block-size market. Because Bitcoin's parameters are hard-coded in its client, the Bitcoin network will split, or hard-fork, unless nodes upgrade their Bitcoin Client at once to, e.g., SegWit2X.

Ethereum faces similar structural problems and will eventually follow the same fate of Bitcoin unless Ethereum adopts Proof-of-Stake. However, Proof-of-Stake is not without its drawbacks, including increased incentives for centralization and the nothing-at-stake problem. These shortcomings have become more evident in recent months, as Ethereum smart contract execution fees (gas) recently rose to more than 5 US cents from less than one cent in 2016. Other consensus models such as Delegated Proof-of-Stake are either unsubstantiated or inevitably introduce one or more parties that must be trusted.

The Basic Attention Token (BAT) crowd sale raised over 35 million US dollars within just two Ethereum blocks, or about 30 seconds. More than 90% percent of its coins went to just a few people. For initial distribution, Cypherium will make every reasonable effort to ensure the distributed ownership of tokens, consistent with our belief in decentralization. High throughput, scalability, fast settlement and low fees -- while retaining decentralization, security, trustlessness, pseudonymity, open membership and immutability -- have become essential for the next generation of blockchain platforms to truly enable enterprise level applications. A public blockchain must be able to upgrade itself without the involvement of a centralized party, such as an official release of a newer client.

Security is another major issue of current public blockchains. Ethereum is the first smart contract platform to see widespread adoption. The DAO attack in June 2016 raised questions regarding whether blockchain history should be re-writeable to serve the interests of a group of people. While the developers of the DAO were working on the solution to fix its bugs, an attacker had already begun to take advantage of those bugs by transferring over 3.6m ethers to his own account. In addition, Ethereum lacks a decentralized, open and transparent governance mechanism that amends itself when it is necessary to apply mitigation or upgrade.

In this white paper, we present a new blockchain, which addresses present shortcomings of current public blockchain infrastructures. Key features of our design are double-chain consensus, federated architecture and separate sandboxes for test and production smart contracts.

2. Cryptocurrencies

Cryptocurrencies are a series of secure, immutable transaction records. The mechanism is similar to issuing a check in real life. When a person wishes to transfer his money to another party, he 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 such that the transaction becomes valid.

Cryptocurrencies have no physical form and only exist within a computer network. Cypherium's native token, Cypher (CPH) is the infrastructure-level cryptocurrency that is required by the platform for the execution of transactions. CPH acts as the incentive to secure the blockchain they are transacted over and preventive measure against Denial-of-Service attacks.

3. Blockchain

Because there is trivial 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', Bitcoin introduced the concept of blockchain. A blockchain is a cryptographic enhanced immutable database. 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 the current block's timestamp, the 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 access its data. So long as a group of nodes maintains a distributed network to broadcast, validate, and store transactions, it will grow indefinitely. The number that each block corresponds to is called block height. In terms of cryptocurrency, due to the possibility of tampering during data transmission and storage process, a public blockchain must be tamper resistant to protect its data. To eliminate dependence on a centralized issuing authority, blockchain uses a decentralized 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 to 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.

4. Consensus

The consensus is the most crucial part of decentralized peer-to-peer network and is imperative for a permission-less blockchain. Due to physical hardware and human factors, latency, downtime and malicious attack are prevalent among distributed systems. When two nodes have inconsistent block records, a fork occurs. This can cause 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 exact 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 greater adoption in Bitcoin[1] and Ethereum[2]. It is the 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 that coincide with mainstream adoption. 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. In contrast, VisaNet is capable of processing 2000 transactions per second on average and may handle over 10000 transactions per second during its peak. However, nearly transactions are validated in one-of-two secure facilities in the U.S. 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 lightning networks, both of which represent off-chain solutions and inevitably bring centralization and security problems.

Because Proof-of-Work blockchains currently consume a relatively large amount of energy, there have also been proposals of consensus on the ownership of stakes[3], or, 'coins'. In this schema, nodes express their acceptance of a transaction by locking up a portion of their coins as security deposit. These nodes, also called stakeholders, serve as validators for all incoming transactions. If a stakeholder commits fraud, its deposit is forfeited as punishment. However, since developers control the initial distribution of coins, Proof-of-Stake is inherently more centralized than Proof-of-Work. Stakeholders tend to store more tokens, reducing the supply of tokens in circulation. Moreover, current implementations of

Proof-of-Stake have proven far less secure than Proof-of-Work because any stakeholder can initiate a nothing-at-stake attack by signing multiple blockchain histories. The centralized stakeholders also create single-point-of-failure. If stakeholders collude together or are taken down by hackers or authority, the entire network goes down as well. Although Proof-of-Stake has been proposed as early as 2012 by Peercoin[4], it has not seen as wide of an 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 $3f+1$ nodes must exist in the system. Within the peer-to-peer network of the blockchain, nodes can be disconnected, shutdown or broadcast malicious information. It is impossible to predict their behavior. In 1999, researchers at MIT for the first time published the Practical Byzantine Fault Tolerance algorithm[5], which reduced the complexity of Byzantine algorithm from exponential to polynomial. This algorithm chooses one node as primary and a group of nodes as backup. The consensus is achieved after multiple rounds of voting amongst nodes. This method was originally designed for internal closed systems, such as flight control system of airplanes and data center clusters. It is not suitable for decentralized, peer-to-peer applications such as cryptocurrencies. Better-known examples of Byzantine fault tolerance based blockchains are Hyperledger, Tendermint[6] and Stellar[7]. In these examples, traditional mining is abandoned and transaction speeds of up to 100,000 tx/s are possible. Similarly, PBFT does not support nodes to join and exit the network at will. And when the total number of nodes go beyond 20, the entire network drastically slows down [8].

5. Design goal

Because an attacker can easily create hundreds of 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 administrators of the blockchain tightly hold the governance of nodes, centralizing the process. A cryptocurrency is effectively worthless if it is not built on top of a trustless mechanism because of its lack of check and balance, and becomes no different from a fiat currency 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 Cypherium.

The purpose of Bitcoin mining is essentially the process of leader election within its peer-to-peer network, and in the meantime makes it more expensive to modify the

blockchain. The winning miner becomes the leader, which is responsible for establishing a unanimously agreed order of operations, more specifically, generating the next block. This process, however, is not related to 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 proposed by Bitcoin-NG[9]. However, in Bitcoin-NG the leader node has complete control over the transaction processing, and if it goes down, the network is stale until someone mines the next key block. ByzCoin[10] is another proposal that combines PBFT with a group of validator nodes dynamically elected by Proof-of-Work. If the leader fails, another node will immediately replace it. The idea of windowed consensus group is also adopted by Hybrid Consensus[10] and ALGORAND[11]. We present a new blockchain consensus mechanism that is inspired by concepts presented in both Bitcoin-NG and ByzCoin. From Bitcoin-NG, we adopt the idea of decoupling key block mining from micro blocks for faster transaction processing. From ByzCoin, we adopt the method of using windows of recent key block miners to elect a group of validators for Byzantine consensus and allow those groups to commit transactions collectively.

Next, we draw an analogy between the voting process of United Nations and the consensus mechanism of Cypherium. 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. N is the number of total voters. If the draft receives more than $\frac{2(N-1)}{3}$ 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 term of the voters and allow all nodes to have an equal opportunity to vote. Moreover, since the Cypherium blockchain is permissionless, and all nodes may join and exit without authorization, nodes must maintain the integrity of the data. By doing so, if a malicious actor produces numerous nodes via a botnet in order to gain more than $\frac{2(N-1)}{3}$ 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.

6. Election chain

Nodes identify themselves by their public key hash and discover one another through gossiping, in which case a node establishes TCP or UDP connection randomly with one or more of its peers. For a fair competition, Cypherium program 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 known to the network, then all competing nodes mine the next block. Each node is a state machine and the rank of the nodes is determined by the value of their Proof-of-Work. When a node mines a Proof-of-Work, it is broadcast to the network and verified by other nodes. The time required for verification is considered trivial, comparing to that for finding the Proof-of-Work. Before timeout, the first n nodes to find the Proof-of-Work hash satisfying current difficulty then become member $M_{1\ldots n}$ of the validator committee for the next validation **term**. If an attacker attempts to modify an election block B at height H_e, he must find all n Proof-of-Work and outrun honest validators. The election block hash is generated with a collective Proof-of-Work of all validators.

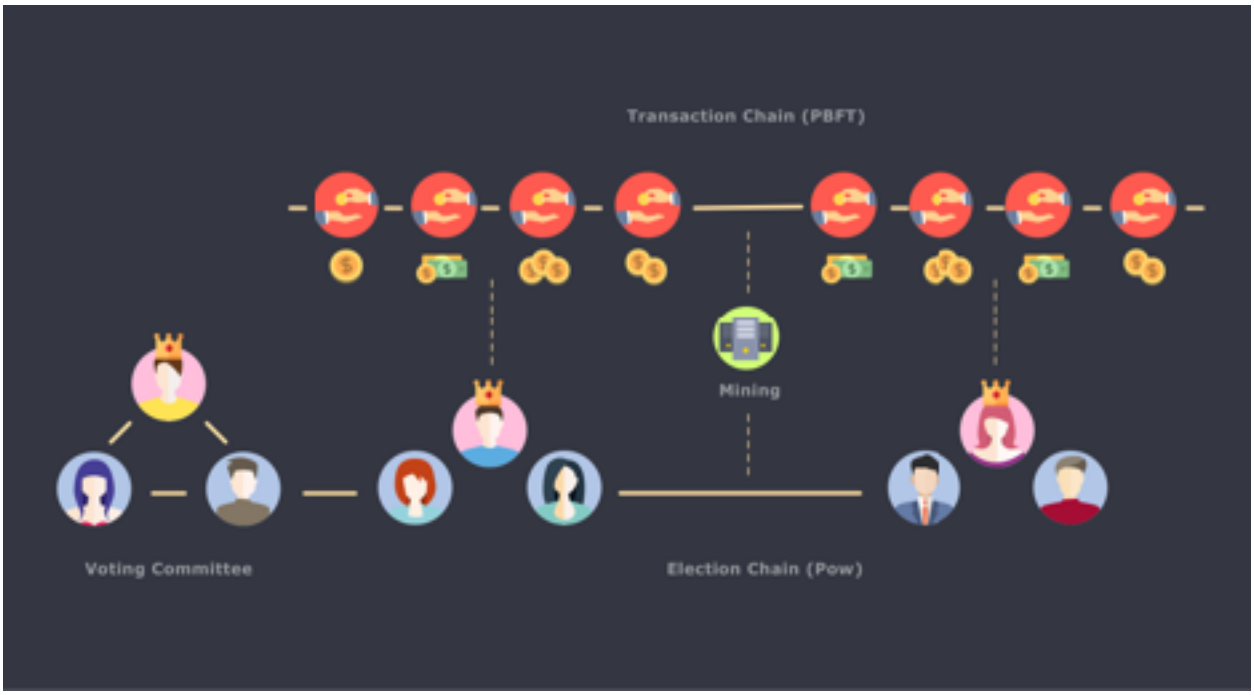
$$B(H_e) = \text{Hash}(M_1(H_e) \,||\, M_2(H_e) \,||\, \ldots \,||\, M_n(H_e))$$

After the mining is finished, the successful competing nodes become **validator** nodes. Classic PBFT determines the **leader** by $p = v \bmod |R|$, where v is the number of total validators and R the set of replicas. The leader is responsible for maintaining a total ordering of client requests. In our version of PBFT, the validator that carries minimum Proof-of-Work becomes the leader of the validator committee. The leader collects transactions into transaction block as well as coordinate other committee members. Each transaction block must be approved by more than n*2/3 committee members before being accepted to the transaction chain. After the term 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 term 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 broadcast simultaneously for the nth position. However, this situation may be addressed by a simple deterministic function that picks only one node as result.

7. 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 configuration of the validator committee is already achieved before voting starts, all transaction blocks are verified instantly without mining and confirmation time. The consensus of transaction blocks is achieved via Byzantine fault tolerance. A transaction block includes transaction records, Merkle root, UNIX timestamp, hashes of the previous transaction block and the correspondent election block. The mining-independent transaction chain allows itself to have dynamic block time and size. Notably, because Cypherium has two distinct chains serving different purposes, we denote the transaction block height as $H_{t.}$ and election block height as height $H_{e,}$ respectively.

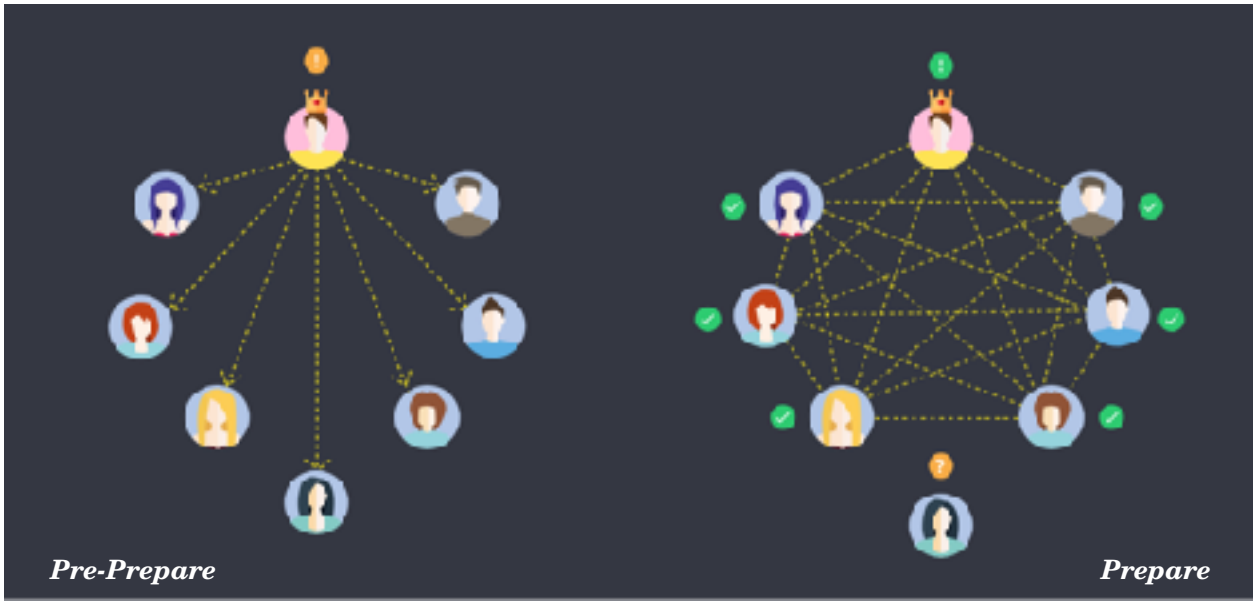


8. Voting process

The Cypherium consensus is de facto the process of state machine replication. Nodes communicate each other through messaging, e.g. gRPC, but they may receive messages in different order. The blockchain requires sequential consistency, which means that the order of execution must be consistent in every replica. To maintain the total order of messages, each **round** of validation consists of a three-phase communication: pre-prepare, prepare and commit.

Pre-prepare

When receiving a request from the client, the leader node multicast a PRE-PREPARE message containing with the client message to all incumbent validators and waits for their reply.



Prepare

An incumbent validator validates the message and multicast a *PREPARE* message to all validators. If 2/3 majority of a quorum is achieved, then the leader broadcasts an announcement to proceed to the next phase. During the voting stage, a validator 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 validators are routinely excluded from the validator committee. Therefore, the voting process continues in the event of more than 1/3 validators become unreachable.

Commit

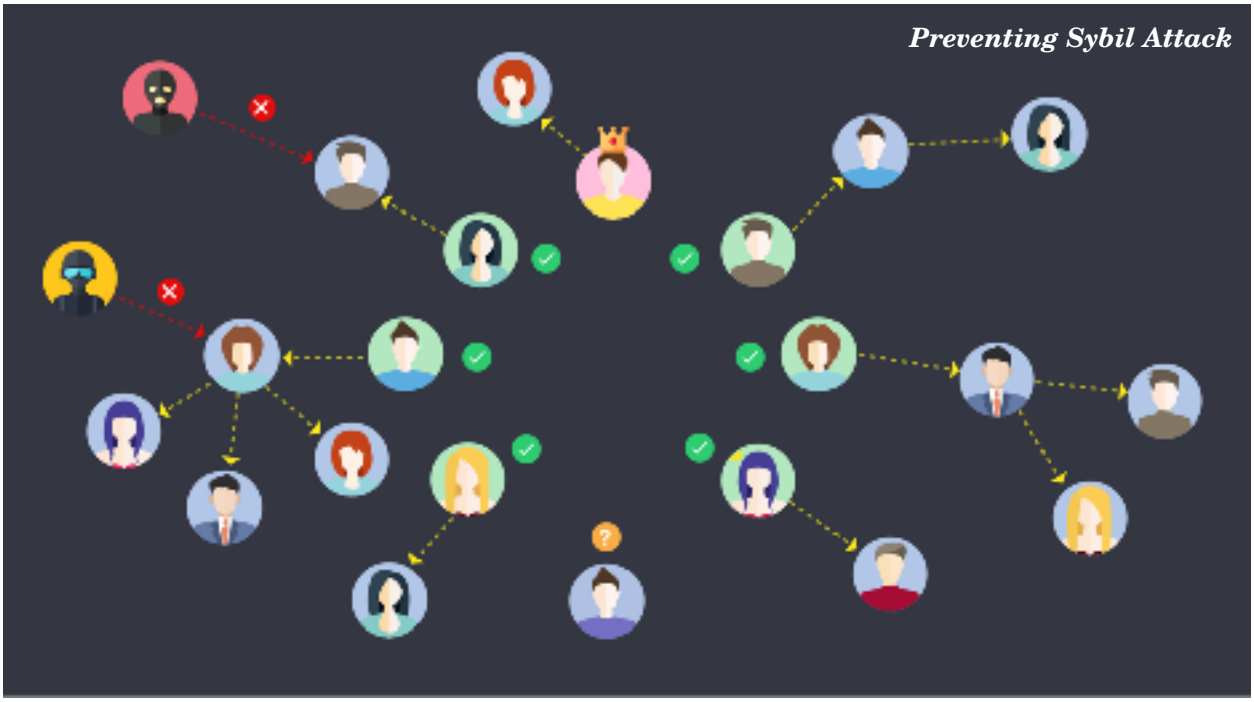
After the prepare phase, validator multicast a *COMMIT* message to all validators. Again, a 2/3 majority of a quorum is required. If a block receives more than 2/3 approval from incumbent validators, it will be committed to the blockchain and broadcast to all nodes

in the network. After successful completion of all the above steps, the current state **s** is

transitioned to the next state **s**'.

View-change

If the leader node commits malicious behavior or fails to respond within a set time, other validators can impeach the leader by multicasting a *VIEW-CHANGE* message to all validators to replace the leader with the next candidate.



Preventing Sybil attack

An attacker cannot outvote honest nodes by creating a large number of fake nodes to control the 2/3 of majority because all validators must provide Proof-of-Work before the network accepts them.

Committee-change

When a new committee is elected, a validator sends COMMITTEE-CHANGE messages to all members of the current committee $v_{c,}$ which stops accepting messages from clients, and transfers the job to the next committee v_n. The last committed transaction block becomes the checkpoint, from which all incoming transactions are handled by the new committee.

9. Multi-sig voting

To illustrate the voting process, we draw an analogy between the voting process of United Nations and Cypherium. 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 N^2 times of check against signatures. When the number of node grows, it could become considerably expensive to verify all signatures. ByzCoin proposed CoSi[12], a collective signing protocol. It can produce an aggregated signature within a group of decentralized nodes so that each node only needs to verify this signature once. The original ByzCoin implemented CoSi over a tree multicast protocol. In our implementation, we adopt gossip as the underlying communication protocol for improved robustness and decentralization.

10. Federated architecture

The biggest challenge of a dynamic membership BFT system is that all replicas must maintain the same state. Otherwise, an out-of-sync validator may produce a false positive on a valid transaction. If a node just starts up after a long period during which it was offline, it must catch up all blocks it is missing before entering the voting committee. To mitigate this issue, we introduce federated architecture. A federated model could ensure synchronization of the network that a validator could fetch missing transaction blocks from its non-validating peers. Multiple nodes can form one virtual node in order to guarantee the completeness of data. Federated architecture can effectively reduce the resource

requirements such as load and balance, and facilitate performance for participants of the network. It is also possible to support sharding by hash partitioning with this architecture.

11. Sharding

Recent research advances have shown the promise of sharding in an open, byzantine environment. Elastico is the first of this proposal. Cypherium will adopt sharding at sometime of its roadmap.

12. Smart contract

To support more advanced and flexible transactions, Cypherium implements a Turing-complete, stack-based scripting system, also known as smart contract. The smart contract runs inside a Cypherium Virtual Machine (CVM). Smart contract enables users to create more complex decentralized apps, including deferred payment, advanced access management, voting applications and user-defined digital assets. The high throughput of Cypherium 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. The DAO attack of Ethereum led to a controversial hard-fork. To prevent flawed code having detrimental effect on the network, such as in the case of the DAO attack, CVM have two separate sandboxes, one for testing and the other for production.

In a **test deployment**, the smart contract can read from the main net, but not write to. Instead, all executions will happen inside an isolated environment, which is essentially a mini fork of the main chain, which only extends the state of the test smart contract. The fork can be discarded upon the completion of testing. This can give developers sufficient time to discover any bugs preceding the **production deployment**.

The account balance model has its benefit of being stateful, which makes it useful to program advanced applications. Cypherium builds an abstract layer on top of the transaction layer, which associates an account with transactions belong to it. To make this architecture more understandable, consider the UTXO model as the lower level stateless HTTP requests, while the account balance model would be the sessions, which hold states.

13. Cypherium Virtual Machine (CVM)

To support more advanced and flexible transactions, Cypherium implements a Turing-complete scripting system, also known as smart contract. The smart contract runs inside a Cypherium Virtual Machine (CVM). Smart contract enables users to create more complex

decentralized apps, including deferred payment, advanced access management, voting applications and user-defined digital assets. The high throughput of Cypherium is capable of satisfying large-scale requirements of enterprise applications.

The Ethereum Virtual Machine (EVM) is the first and most widely adopted virtual machine for smart contracts. It is stack-based and only the data at top of the stack is accessible by the memory. Cypherium Virtual Machine (CVM) is based on the Dalvik architecture, which is register-based. Register machine allows arbitrary access to data and requires less operation steps. The Dalvik is a lightweight Java Virtual Machine originally adopted by Android operating system. Therefore, Cypherium is possible to execute smart contracts on mobile devices. The CVM has total 65536 registers.

Here's a simple arithmetic calculation performed by CVM:

```
INC %1            (Register 1 value +1,  Reg1 = 1)
INC %2            (Register 2 value +1,  Reg2 = 1)
INC %1            (Register 1 value +1, Reg1 = 2)
ADD %2 %1 %2      (Reg2 = Reg1 + Reg2 = 3)
```

This code calculates the mathematical expression 1 + 2, and the result is 3. The equivalent EVM code is:

```
PUSH4 1           (Push 1 on stack)
PUSH4 1           (Push 1 on stack)
MSTORE            (Save data to memory)
POP               (Remove data from stack)
PUSH4 1           (Push 1 on stack)
ADD               (Addition operation)
MLOAD             (Load data from memory)
ADD               (Addition operation)
```

EVM requires twice steps as CVM. As the complexity of smart contracts grows, the performance degradation of EVM shall grow exponentials versus that of CVM. The gas cost is also proportional to the number of operations.

EVM does not check overflow and underflow by default; developers must rely on external libraries to implement safemath check. However, developers not always remember to implement safemath, leaving many multi-million dollar smart contracts vulnerable to overflow attacks. CVM performs safe check at the virtual machine level, completely eliminating the need for developers to implement their own safe check. The runtime will throw an exception if it encounters an overflow.

The EVM does not support fixed-point representation. A fixed-point number is an expression of fractional numbers using fixed number of bits. This feature is significant in financial applications and embedded chips for its low cost of computing resource.

All EVM's instructions operate on its basic data type, the 256-bit words. Most modern CPUs are using either 32-bit or 64-bit integers as basic type. To perform 256-bit operation on a 32-bit or 64-bit processor require excessive steps, severely dragging down the computation speed and consuming substantial amount of computing resources. CVM comes with native support for 64-bit integer, which can boost its performance by an order of magnitude. If a number exceeds this boundary, it will be converted to BigNum type, which supports arbitrary data size.

CVM supports all equivalent operations of the EVM. Like the EVM, the CVM uses gas to ensure the smart contract execution terminates. The instruction set is optimized so that only smart contract related instructions are kept. Contracts may communicate with one another through messages.

Cypherium smart contract is based on Java syntax. The Java code is first compiled into CVM bytecode, then executed in the CVM. CVM will have Just-In-Time compilation in the future to further improve its performance.

14. Use cases

Alongside value transfer, a Cypherium smart contract can support storage for various formats of data. The distributed storage of the Cypherium blockchain aims to provide better stability and is ideal for anyone seeking high availability and security of sensitive information. Below is an incomplete list of use case scenarios for Cypherium.

Finance

Stock, bonds and other financial instruments can be implemented using Cypherium blockchain. Comparing with traditional exchange centered system, blockchain based solution is more effective and costs less. Cypherium smart contract supports conversion and transfer of any currencies, which settle using CPH as an intermediary currency.

Digital Contract

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

Messaging

Cypherium's built-in transaction fee mechanism can effectively prevent spamming. This allows users access to secure information exchange. Receivers can verify the authenticity of the message by checking its signature against the public key that the sender claims to own.

Voting

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

Notarization

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

Secure data storage

The nature of blockchain is an append-only log database secured by cryptographic algorithms. Cypherium's dynamic node election mechanism ensures that all nodes are replaced after a certain period, and therefore minimizes the possibility of single point of failures. This feature makes it ideal to store high-value critical data. Cypherium can also serve as a registry of hashes that point to files or data records on external storages eg. IPFS.

Internet of Things

Cypherium is capable of registering billions of IoT devices and enhancing their security through cryptographic proofs. Nodes can verify each other's authenticity via information registered on the Cypherium blockchain.

Artificial intelligence

Finite State Machine has been long and widely used in modeling artificial intelligence. CVM enables AI driven applications that involves frequent state transitioning and deep logic steps.

15. Protocol upgrade

In order to extend its performance and capacity without hard-forking, Cypherium allows participants to adaptively adjust its protocol. The committee can decide at certain intervals and vote on whether to approve proposed upgrades. To prevent instability, the new

value cannot offset more than one percent of its previous value. Upon approval, all nodes will switch to the new protocol. The protocol begins with its version number, along with one or more of the following parameters:

1. Transaction block size
2. Minimum smart contract execution fee
3. Virtual machine stack depth
4. Network connection timeout

16. 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 contribute to centralization within a permission-less system. Cypherium proposes the use of an ASIC-resistant CPU mining algorithm which is bound to device memory.

Address generation

Cypherium uses Ed25519, an implementation of Schnorr signature algorithm, for its address generation. Ed25519 is faster than ECDSA and has a 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. An 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

17. Transaction

Traditional centralized payment-processing works by maintaining a list, which contains account, balances of all users. A transfer in such system is simply the addition of one account and subtraction of the other. All transactions of an account can be easily linked. To protect user privacy and decentralization, Cypherium 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. A 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. If a previous input is missing from one validator's blockchain data, the validator can walk through the Merkle tree and retrieve the transaction block containing the missing input from its peers.

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 by the transaction protocol. This reward is determined by the current reward of the network. Unlike Bitcoin, the coinbase transaction of Cypherium does not occur immediately after an election block is mined, but after the term of current validator committee has ended. This mechanism effectively ensures the honesty of nodes and suppresses the selfish-mining exhibited in Bitcoin.

Transaction fees

Because nodes consume resources such as electricity, bandwidth, storage 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. Nodes calculate transaction fees based on their data size and complexity. The difference between all transaction inputs and output is the transaction fee paid to the nodes.

Multi-signature transaction

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 Cypherium 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 incumbent validators handle the verification of a single transaction. If one of the following conditions is true, then the transaction is considered invalid:

1. The signature of the transaction does not match the owner of the coins
2. The total output exceeds total input
3. 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.

18. 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 blockchain 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 of 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, an 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 do 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 throughout their term.

19. Conclusion

To enable anonymous participation of anyone while being resistant to Sybil attack, we introduced Proof-of-Work as the node election mechanism. Elected nodes will only hold temporary position to verify transactions, and thus the network will retain its decentralization. Confirmation time is no longer required since 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.

Acknowledgement

We extend special thanks to the people who offered their insight and assistance during the composure of this white paper: Emin Gün Sirer and Bryan Ford.

References

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

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

[3] BitFury Group, Proof of Stake versus Proof of Work White Paper, http://bitfury .com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf, 2015

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

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

[6] Jae Kwon, Tendermint: Consensus without mining, http://tendermint.com/docs/tendermint.pdf, 2014.

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

[8] 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.

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

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

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

[12] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, Bryan Ford, Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing, *25th USENIX Security Symposium*, August 2016

[13] 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.

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

[15] 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.

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