Improved Hybrid Consensus Scheme with Privacy-preserving Property

February 25, 2017

Abstract

Proof-of-work-based consensus, adopted in Bitcoin, has already drawn much attention from cryptocurrency and block chain community. Despite its nice decentralization property, it has significant limitation in terms of efficiency since transactions can not be confirmed within seconds. In 2016, hybrid consensus was proposed to partially deal with this issue by introducing committee responsible for validating transactions. However, there still exists some issues with respect to this hybrid consensus such as selfish mining, fairness to the election of committee member, incentives for the consensus scheme, and so on.

To improve the hybrid consensus further, we first present a substitution for proof-of-work, named as fair-proof-of-work (fPoW), to solve the issues related to selfish mining and fair committee election. We also demonstrate the incentives for our improved hybrid consensus. Then, based on this consensus, we build privacy-preserving constructions (including identity and content privacy preserving) to make the consensus more applicable and powerful. Finally, we give formal security proof for our newly-proposed consensus scheme. It is expected that this novel consensus scheme could be adopted in block chains which require decentralization, high efficiency, as well as privacy-preserving.

1 Introduction

Bitcoin, firstly proposed in [Nak08], utilized blockchain (we may refer to this in term *Nakamoto Chain*) as building block, to achieve decentralized transaction based on a gossip network. Soundness of Nakatomo chain is guaranteed as long as majority of total computing power is at hands of honest nodes. For years it has been attracting much eyesight (see [BMC+15]).

Blockchain requires miners to solve a hash problem, that is to find a nonce nc, so that $H(\mathsf{rec}, \mathsf{nc}) \in \mathsf{target}$, where rec is all records in this block (except for this nonce), and $H(\cdot)$ is a collision-resistant hash function.

However, traditional block chain is facing two tough problems. Firstly, it has no guarantee on privacy of identity and transaction amount, although pseudo-identity might be utilized to achieve anonymity, some previous works have shown that such anonymity is not dependable (see [**?**]) and that no much users could manage too many keys well. Secondly, traditional block chain grows at unsatisfactory speed, and it takes 10 minutes on average to create a new block (at least before the year this paper is written). What's worse, to make sure one block will eventually be on-chain, one has to wait till λ new blocks appended to that block.

Works have been done in attempt to achieve real anonymity. *ZeroCoin* and *ZeroCash* proposed in [MGGR13] and [BCG+14], provided us a novel vision of transaction anonymity by implementing zero-knowledge proof. However, though it is a smart extension of traditional block chain, it makes no significant improvement on efficiency.

Also, works have been done to achieve better efficiency by replacing *Proof-of-work* by *Proof-of-stake* (see [BGM16]) or PBFT (see *Hybrid Consensus*), which help achieve better efficiency, but still without guarantee on privacy of transactions.

Hybrid consensus proposed in [PS16b] gave an outline of new utilization of *proof-of-work* by taking Nakamoto Chain or FruitChain (see [PS16a]), which are named snailchain, as generator of a rotating committee, and all transactions are validated by the committee through a Daily BFT protocol.

Committee members of each round are miners of last csize blocks, that is to say, committee members will wait till generation of csize new blocks to perform a switchover. Soundness of this construction is guaranteed as long as over $3/4$ (or $2/3$ for Fruit Chain) computing power is at hands of honest nodes.

Here by term *round*, it means time interval for creation of csize blocks in snailchain, which is also the term of service of current committee. It denotes each round with consequent natural numbers starting from 1. It denotes validated transaction log of round R as rec_R, and $\mathsf{rec}_R[l]$ means l-th transaction in daily log (l is called a sequence number). Moreover, it uses notation $\mathsf{CM}_R = [\mathsf{ID}_1, \mathsf{ID}_2, \ldots, \mathsf{ID}_{\mathsf{csize}}]$ to denote set of committee members for round R.

In [PS16b], Permissioned Byzantine Fault-tolerance(PBFT), proposed in [CL99], is recommended to perform daily BFT. PBFT guarantees that all honest nodes would come to same consensus proposed by an honest node when it terminates.

Here are **Some drawbacks of original *Hybrid Consensus* :**

Privacy. In original work of *Hybrid Consensus* , transactions are open to public.

Motivation for honesty. In *Hybrid Consensus* , honesty of committee members are guaranteed from block reward and transaction fee of traditional blockchain. However, it merely guaranteed honesty and hard-working of nodes as committee candidates, not as committee members.

Existence of forking In *Hybrid Consensus* , forking of underlying snailchain exists, wasting great amount of time and energy, making selfish mining possible (that is the reason why it needs $3/4$ overall honesty for *Nakamoto Chain*).

In this paper, we will propose a novel construction, with real privacy-preserving property, as well as a satisfactory efficiency, and its soundness will be robust against active attackers. Demand on fraction of honest computing power will be roughly $2/3$, same to FruitChain case in *Hybrid Consensus* .

1.1 Our contribution

In our construction, we use *Hybrid Consensus* as underlying protocol. Of course, we have made some modification on original *Hybrid Consensus* protocol, so that we can achieve following properties.

- **Transaction Privacy.** In this construction, all transactions are only accessible to members of rotating committee, with techniques in **Section 4**. Even for committee members, identity of sender and receipt are blurred out with special techniques listed in **Section ??**.
- **Permissionless model with excellent Efficiency.** This is a permissionless model, where nodes can join and leave dynamically. In traditional constructions, a permissionless model means terrible performance in efficiency of transaction validation. However, with rotating committee elected from underlying snailchain (see *Hybrid Consensus*), we can validate transactions through BFT network among committee members. In such way, satisfactory efficiency can be achieved.

  Inherited from *Hybrid Consensus* , our confirmation time is bounded by actual delay, instead of theoretical upper-bound of delay.
- **Forking-free.** In traditional blockchain, forking has to be implemented to tackle with ambiguous. However, forking is waste of time and energy, causing damage to fairness of committee election. Users have to wait for generation of sufficiently many new blocks to conform a transaction. Power consumed by miners who followed "wrong" block has to be wasted in vain, for the same reason, fairness is in hazard. Also, existence of forking leads to existence of selfish mining. In *Hybrid Consensus* , forking still exists in underlying snailchain. While in our construction from **Section 3**, forking can be prevented.
- **Security.** Compared with related works (including *Hybrid Consensus*), our construction is entitled with following security properties.

  Tolerated corruption. In this work, we require roughly $2/3$ overall honest to achieve $2/3$-chain quality, so as to assure $2/3$ BFT committee members are honest.

  Looser assumption against mildly agile corruption. In *Hybrid Consensus* and our work, adversary is allowed to perform mildly agile corruption, i.e., they can choose nodes to corrupt according to the configuration of environment. τ-agility, which means an adversary has to wait for τ time to corrupt a honest node, is defined to describe security against adaptive corruption. In our work, assumption on τ can be much looser than that of *Hybrid Consensus* .

  Preventing transaction pool and selfish mining. In traditional block chain, with existence of transaction pool, selfish mining may happen (see [ES14]). However, in our forking-free construction, selfish mining has no reason to exist, and for this reason.

  Defending against retroactive attack. In our construction in **Section 3**, hash values of each round's transaction history are timestamped through underlying block mining.
- **Fairness in competition.** Without existence of forking, selfish mining is prevented. Also, with our fPoW, better fairness to committee candidates can be guaranteed in face of delay.
- **Arbitrary Combination between** PoW **and** PoS **.** In [POA16] (which is an adaptive version of fPoW), we will present one method to arbitrarily combine PoW and PoS .
- **Chain quality and tolerated corruption.** Our demand for chain quality and tolerated corruption follows that of *Hybrid Consensus* (with FruitChain as underlying block chain). That is to say, we need roughly $2/3$ overall honest to achieve $2/3$-chain quality, so as to perform PBFT safely.

Remark: In this paper, we will use *Hybrid Consensus* as underlying protocol. All security guarantees proved in *Hybrid Consensus* will not be repeated in this paper.

1.2 Paper organization

In **Section 2**, we will present notations and preliminaries for our construction. Followed by **Section ??**, where we will introduce parts of related works to form comparison and present our designing goal.

Nextly, in **Section 3**, we are going to present our forking-free framework of committee election, followed by **Section 4**, in which we present our privacy-preserving consensus on transaction validation.

After that, in **Section ??**, we present methods to blur out identities to committee members, and technical details on log storage. And in **Section 5**, we will show formal proof for security of our construction. Proofs regarding to underlying *Hybrid Consensus* that has already shown in [PS16b] will not be repeated in this paper.

Finally, in last few sections, we will do qualitative description on more detailed techniques and analysis. And in **Section ??**, we will show figures to conclude our construction in a nutshell.

2 Notations and Preliminaries

In this section, we present notations for our work, as well as some preliminaries.

2.1 Notations for consensus

Notation	Description
κ, λ	security parameters for encryption and consensus
ID	in this work, by term identity, we mean a pseudo-identity named by node itself (collision can be prevented)
PID	PID is blurred out ID, which will be shown in **Section ??**
R	round number
ℓ	sequence number
CM_R	committee members for round R
csize	size of rotating committee csize $:= \Theta(\lambda)$
(PK,SK)	public and private key for public key encryption scheme
ek	key for symmetric encryption
(pk, sk)	public and private key for digital signature scheme
(pk, sk)	public and private key for hybrid encryption scheme
(k, ψ)	session key k and its encapsulation ψ in hybrid encryption scheme
$in = (R, l)$	in is index of one transaction, R is its round number and l is its sequence number of that round
$\mathsf{round}(in)$	$\mathsf{round}((R, l)) = R$
$\mathsf{seq}(in)$	$\mathsf{seq}((R, l)) = l$
$\langle \mathsf{tx}, \mathsf{tx}' \rangle$	pair of payment transaction and corresponding charge transaction
$\langle \mathsf{TX}, \mathsf{TX}' \rangle$	pair of payment transaction and corresponding charge transaction after encryption
txc, TXC	transaction command txc and its encryption TXC
txp, TXP	transaction proof txp and its encryption TXP
val(tx)	transaction amount for tx
$\mathsf{val}_{\mathsf{ek}}(\mathsf{TX})$	transaction amount for TX, which requires symmetric key ek to be disclosed
$(\mathsf{PK}_{com}, \mathsf{SK}_{com})$	public and private key for committee in public key encryption scheme
(pk_{com}, sk_{com})	public and private key for committee in hybrid encryption scheme

Table 1: Notations for construction

For convenience, we denote transactions in lowercase tx as transaction before encryption, and encrypted transaction in uppercase TX, and regard its corresponding symmetric key as ek. Additionally, since indexes of all income transactions of current transaction have to be made public (so that committee of different rounds could check whether double-spending happens), TX should include all income indexes.

Remark: Similarly to that of bitcoin, we assume all transactions may have multiple incomes, but only two outcomes: one for payment, one for charge (sent to itself or another account held by itself).

We use notation $\mathsf{val}(\mathsf{tx})$ to denote quantity of currency in transaction tx, $\mathsf{val}_{\mathsf{ek}}(\mathsf{TX})$ for transaction TX encrypted with ek.

For example, for Alice to transmit m SHCoins to Bob, she may not use all SHCoins from income transactions, so she may need to return charge of amount m' to herself. More formally, she makes such transaction pair:

$$\langle \mathsf{tx}, \mathsf{tx}' \rangle = \langle (\mathsf{PID}_{Alice}, \mathsf{PID}_{Bob}, m), (\mathsf{PID}_{Alice}, \mathsf{PID}'_{Alice}, \mathsf{m}') \rangle$$

so that once this transaction goes through consensus, *Bob* later on will be able to spend the m SHCoins in later transaction tx' when he needs, he can include round and sequence number this transaction as source reference of his later transaction. Of course, it must guarantee that

$$\sum_{i=1}^{t} [\mathsf{val}_{\mathsf{ek}_{in_i}}(\mathsf{TX}_{in_i})] \geq m + m'$$

We note

$$\begin{aligned} \mathsf{TX} &= (\mathsf{SE.Enc}(\mathsf{ek}, \mathsf{tx}), t, in_1, in_2, \ldots, in_t) \\ \mathsf{TX}' &= (\mathsf{SE.Enc}(\mathsf{ek}, \mathsf{tx}')) \end{aligned}$$

Within an index of income source $in = (R, l)$, R is the round and l is one sequence number of that round.

Transaction command

In our construction, transactions command includes transaction $\langle \mathsf{TX}, \mathsf{TX}' \rangle$, symmetric key ek to get $\mathsf{tx} \leftarrow \mathsf{SE.Dnc}(\mathsf{ek}, \mathsf{TX})$, as well as symmetric keys $(\mathsf{ek}_{in_1}, \mathsf{ek}_{in_2}, \ldots, \mathsf{ek}_{in_t})$ to open up all previous transactions which serve as income of the current transaction.

For convenience, we note each transaction command as

$$\begin{aligned} &\mathsf{txc} = (\langle \mathsf{TX}, \mathsf{TX}' \rangle, \mathsf{ek}, t,\ \mathsf{ek}_{in_1}, \mathsf{ek}_{in_2}, \ldots, \mathsf{ek}_{in_t}) \\ &(k', \psi) \leftarrow \mathsf{KEM.Enc}(pk_{com}) \\ &\mathsf{TXC} = (\psi, \mathsf{DEM.Enc}(k', \mathsf{txc})) \end{aligned}$$

Detailed illustration on notations for transaction will be presented in following sections.

2.2 Proof-of-work

Proof-of-work, firstly proposed to prevent e-mail spamming, has been introduced to bitcoin system in order to make sure any newly generated block is mined by an honest node with probability equal to fraction of total honest computing power.

In detail: suppose H is a cryptographic hash function, $\boldsymbol{B}_i$ is i-th block on chain, and target is a target range to adjust difficulty of puzzle. Miners of block prove their computing power by trying to solve hash-puzzles $H(\boldsymbol{B}_{i-1}, \mathsf{nc}_i) \in \mathsf{target}$ after block (say, $\boldsymbol{B}_{i-1}$) on chain, so that (s)he could propose $\boldsymbol{B}_i$ with nc_i (along with reward transaction to reward itself) appended onto it.

2.3 Blockchain

Block Chain (for short, *blockchain*), was digital currency system firstly proposed by *Satoshi Nakamoto* in [Nak08]. In bitcoin system, network links transactions by time-stamping technique, and hashing them into a chain of hash-based proof-of-work. In such way, history that cannot be tampered without redoing the proof-of-work is formed.

2.4 Hybrid consensus

Hybrid Consensus, proposed in [PS16b], was a brand-new cryptographic scheme utilizing Nakamoto blockchain or *FruitChain* (see [PS16a]) as underlying $\mathsf{snailchain}$, so as to dynamically maintaining a rotating committee. And all transactions are verified by a BFT network among committee members.

2.5 Permissioned model

Permissionless models, where nodes are allowed to join and leave dynamically, often lacks effciency in practice. On the other hand, Permissioned models, where nodes are pre-determined, can achieve satisfactory efficiency. In *Hybrid Consensus* , a rotating committee is elected from a permissionless environment, so as to perform permissioned BFT among committee members. With this technique, efficiency of transaction validation is guaranteed in a permissionless environment.

2.6 Mildly agile corruption

In fully adaptive corruption model, we assume that adversary can perform any corruption without any cost of time. This assumption is too strong in practice, since adversary has to spend long time locating a node, when adversary in fact only know its pseudo-identity.

τ-agile corruption is assumption that adversary has to spend time τ to corrupt a node. That is to say, time interval between adversary's deciding to corrupt a node and getting finally corrupted should be at least τ.

Remark: Although we write pseudo-identity of nodes as $(\mathsf{ID}_1, \mathsf{ID}_2, \ldots)$, ID here is only a pseudo-identity named by node itself.

3 Improved Hybrid Consensus with Fair-proof-of-work

In *Hybrid Consensus* , for each round (we denote R as round number), transactions are validated through PBFT network among committee members, who are the miners of last csize on-chain blocks at end of previous round, of underlining $\mathsf{snailchain}$.

However, *Hybrid Consensus* is facing some challenges. Firstly, committee election in *Hybrid Consensus* is not forking-free (which we will unfold in next subsection). Secondly, original *Hybrid Consensus* is not adaptive to the case in which we hope to do arbitrary combination between PoW and PoS . Our construction *fair-proof-of-work* (in short, fPoW) will be forking-free, and that, *enhanced-proof-of-activity* (in short, ePoA), which is adapted from fPoW, will achieve arbitrary combination between PoW and PoS . We will unfold ePoA in [POA16]. In fact, fPoW is a special case of ePoA .

3.1 Demand for forking-free construction

When multiple nodes mine a block at almost same time, how to make sure all nodes concede to ”miner of next block”? In traditional block chain, forking of chain is introduced to solve this issue. Forking is necessary in traditional Bitcoin, because forking also helps tackle with problem of misbehaviour of miners, but it is not the case in the existence of rotating committee since validation of transactions have nothing to do with non-committee miners.

Demand for forking-free construction arises mainly for three reasons. Firstly, forking is waste of energy, as much computation power would be wasted in vain merely in order to tackle some ambiguity. Secondly, forking is waste of time, since that in order to believe one block will be finally on-chain, one has to wait till λ new blocks are created following that block. Also, long interval of waiting means that committee members might be more vulnerable under target corruption. Thirdly, forking-free helps to prevent *selfish-mining* (since selfish mining exists only when forking is possible, see [ES14]).

3.2 Fair-proof-of-work

In *Hybrid Consensus*, where committee members are selected from traditional *Proof-of-work*, controversial problems arises. Compared with original *Hybrid Consensus* , fPoW mainly has following advantages:

Forking-free. The greatest problem of *Hybrid Consensus* is that, forking is still needed in competition for block mining. However, forking is what we want to prevent. To avoid forking, we can revise the rule and stipulate that blocks should be appended with broadcast time and the *first* rightful one was broadcast should be the next block, it is possible in theory but not in practice, since it is not practical for all nodes to share the same clock, and nodes have no reason to behave honestly when appending broadcast time.

In our construction, we need neither guarantee on time synchronization nor honesty of nodes, to achieve a fair competition for "*first* miner of next block" without existence of forking.

Fairness in competition. In original *Hybrid Consensus* , even if we can stipulate that *first* miner of next block should be next lucky candidate, so as to achieve forking-free, and that time synchronization and honesty can be guaranteed, miners constantly suffering network delay may still feel unfair in competition for "*first* miner of next block".

Our construction guarantees on more fairness on candidates suffering network delay. In **Appendix C**, we will give a formal proof of why our newly proposed construction excels ordinary PoW considering the existence of network delaying, in competition for "*first* miner of next block", even if time synchronization and honesty can be guaranteed.

Robust against target corruption. Secondly, committee switchover in *Hybrid Consensus* happens every creation of csize blocks, and due to the existence of forking, nodes have to wait for creation of at least λ blocks to finally become a committee member. This is too long an interval, as well as a great risk for them to be corrupted in this process.

Preventing selfish mining. In our forking-free construction, selfish mining has no reason to work. In *Hybrid Consensus* , $3/4$ overall honesty has to be guaranteed (if underlying snailchain is *Nakamoto Chain*) to achieve $2/3$-chain quality, due to existence of selfish mining. However, we have no such concern in this work.

Adaptive to combination of PoW **and** PoS **.** Thirdly, *Hybrid Consensus* utilizing traditional block chain as building block is not adaptive to fit in the case in which we hope to do arbitrary combination between *Proof-of-work* and *Proof-of-stake*. This issue will be briefly introduced in **Appendix D**, and full paper of arbitrary combination between PoW and PoS , which we call it *proof-of-liveness* (ePoA), will be released soon after publication of this paper.

We propose *fair-proof-of-work* (fPoW), which is new vision of *Proof-of-work*, which benefits from the existence of rotating committee. We denote it as fPoW in order to distinguish it from traditional PoW. In traditional construction, for each candidate, probability of mining a nonce for each block is roughly proportional to its computing power, similarly, in our construction, we lower done difficulty of mining puzzle to make expected number of nonce found proportional to its computing power.

In our construction, hardness of nonce-puzzle is smaller than that of PoW. In each round, each candidate u finds some nonce solutions, say, $\mathsf{nc}_{u,1}, \mathsf{nc}_{u,2}, \ldots, \mathsf{nc}_{u,P_u}$. At the end of this round, candidate submits all solutions it found to the committee, then committee members arrange all received solutions in an array L in a certain order, and decides one random number $1 \leq r \leq |L| = \sum_u P_u$ and announce new committee member of next round, who is the miner corresponding to the r-th item of L. We will give a more detailed description in next part.

3.3 Committee from fPoW

In our construction, one node enters committee and one leaves for each round. Hence our round interval is much shorter than that of *Hybrid Consensus* (which is called ”`Day`” in *Hybrid Consensus*).

Here we give a summary introduction to fPoW, more details will be shown in [POA16].

3.3.1 On side of candidates

In round R, one candidate, say, Tom, collects all transactions of round $R-1$ (signed by at least $2/3$ committee members) and arrange them according to sequence order into rec_{R-1}, then finds as much as possible nonces $\mathsf{nc}_{tom,1}, \mathsf{nc}_{tom,2}, \ldots, \mathsf{nc}_{tom,P_{tom}}$ such that

$$H(\boldsymbol{B}_{R-1}||\mathsf{ID}_{tom}||\mathsf{nc}_{tom,i}) \in \mathsf{target} \qquad (\text{for all } 1 \leq i \leq P_{tom})$$

where block content $\boldsymbol{B}_{R-1} := \{\mathsf{rec}_{R-1}, H(\boldsymbol{B}_{R-2}), \mathsf{CM}_{R-1}\}$ is block of previous round. Note that differently from traditional Bitcoin block chain, rec_{R-1} here includes users' transactions handled by previous round's committee, reward transactions for previous round's committee. CM_i is identity list of previous round's committee members.

Tom arranges all nonces found along with signatures into L_{tom}:

$$L_{tom} = \begin{bmatrix} \mathsf{nc}_{tom,1} & \mathsf{ID}_{tom} \\ \mathsf{nc}_{tom,2} & \mathsf{ID}_{tom} \\ \vdots & \vdots \\ \mathsf{nc}_{tom,P_{tom}} & \mathsf{ID}_{tom} \end{bmatrix}$$

and submit all items in L_{tom} to the rotating committee before the end of round R.

Remark: In practice, it is not good idea to submit nonces to all committee members and assume they will all receive same number of nonces during whole interval of round R. We have shown a practical submission protocol in [POA16]. However, in this paper, we merely assume that a safe submission protocol exists, to simplify representation.

Also, in this paper, we assume all committee members will know when each round begins and stops. This assumption is not practical but solution to this problem is too subtle to be mentioned in this paper, we will present it in [POA16] (since fPoW is in fact special case of ePoA). At end of each round, each member declares (`"Terminate"`, R) along with its signature.

3.3.2 On side of current committee member

For simplicity, we order all committee members in $1, 2, \ldots, \mathsf{csize}$.

Each honest committee member receives L_u from all candidates, putting all L_u into L, and sorting all items in the same order, to get

$$L = \begin{bmatrix} \mathsf{nc}_{A,1} & \mathsf{ID}_A \\ \mathsf{nc}_{A,2} & \mathsf{ID}_A \\ \mathsf{nc}_{B,1} & \mathsf{ID}_B \\ \vdots & \vdots \\ \mathsf{nc}_{tom,1} & \mathsf{ID}_{tom} \\ \mathsf{nc}_{tom,2} & \mathsf{ID}_{tom} \\ \vdots & \vdots \\ \mathsf{nc}_{tom,P_{tom}} & \mathsf{ID}_{tom} \\ \vdots & \vdots \end{bmatrix}_{|L|\times 2}$$

Before beginning of next round, committee members in $\mathsf{CM}_R = [\mathsf{ID}_1, \mathsf{ID}_2, \ldots, , \mathsf{ID}_{\mathsf{csize}}]$ produce a random number $1 \le r \le |L| = \sum_u P_u$ by the procedures in Figure 1. That is, all members firstly find random number r_j (for j-th one) and broadcast g^{r_j}, where g is generator of a multiplicative group of order of a large prime number greater than 2^λ. After that, all members broadcast r_j. In such way, adversary can control nothing about generated random number, as long as any one committee member is honest.

Generating random number $1 \le r \le |L|$ on round R
(for member of identity ID_i , $1 \le i \le$ csize)

```
$\boxed{\mathsf{CM}_R = [\mathsf{ID}_1, \mathsf{ID}_2, \ldots, , \mathsf{ID}_{\mathsf{csize}}];}$
Prime number $p > 2^\kappa$, and $g$ be one generator of $\mathbb{Z}_p^*$;
Choose randomly $r_i \leftarrow \{0,1\}^\kappa$;
Broadcast commitment $\mathsf{commit}_i := g^{r_i}$ (and its signature) through PBFT;
Keep pending till all other members $j \ne i$ broadcast $\mathsf{commit}_j := g^{r_j}$;
Broadcast $r_i$ (and its signature) through PBFT towards all other members;
Receive all $r_j$ $(j \ne i)$ from other members through PBFT network;
For $j \ne i$: if $g^{r_j} \ne \mathsf{commit}_j$, then $\{$set $r_j \leftarrow 0$; Accuse $j$;$\}$
$r \leftarrow \lceil \Big(1 + \mathsf{PRF}(\bigoplus_{j=1}^{\mathsf{csize}} r_j, R)\Big) \cdot \frac{|L|}{2^\kappa} \rceil$
```

Figure 1: Generating random number for committee election

By term `"Accuse"`, we mean to vote denial during final voting process (which will be sketched in **Section** 5.2).

Finally, committee members declare (`"Enter"`, ID') along with their signatures, where ID' is pseudo-identity of miner of r-th nonce. This lucky candidate is enrolled into committee if this declaration is signed by at least $2/3$ current committee members. After that, this round ends and next round begins.

4 Privacy-preserving Constructions built on Our Improved Hybrid Consensus

In *Hybrid Consensus* , encryption on transaction is not introduced, hence are open to all publicity. In this section, we propose framework of a privacy-preserving consensus, in which all transactions are encrypted by public key negotiated by committee members. Hence all transactions can be blurred out to publicity.

In our construction, one transaction (except income index) will be encrypted with a symmetric encryption key, held by the payer, payee and committee members on the round of transaction.

4.1 Committee side

Before starting each round of daily BFT, committee members of current round determine two pairs of keys $(\mathsf{PK}_{com}, \mathsf{SK}_{com}), (pk_{com}, sk_{com})$ from negotiation, description of this negotiation process will be sketched in **Appendix A**.

Upon receiving a transaction command TXC, it verifies TXC and checks whether double-spending of income indexes exists, and perform PBFT within the committee.

To make judgement on one transaction command, each honest committee member gets $k' \leftarrow \mathsf{KEM.Decap}$ (sk_{com}, ψ), and then $\mathsf{txc} \leftarrow \mathsf{DEM.Dec}(k', \mathsf{TXC})$, after that, gets $b \in \{0, 1\} \leftarrow \mathsf{judge(txc)}$, where judge(txc) is to take validation procedure of committee as a black box, including checking of transaction identities, double-spending detection, and amount verification.

```
judge(txc)
----------------------------------------------------------------
where $\mathsf{txc} = (\langle \mathsf{TX}, \mathsf{TX}' \rangle, \mathsf{ek}, t, \mathsf{ek}_{in_1}, \mathsf{ek}_{in_2}, \ldots, \mathsf{ek}_{in_t})$
$\mathsf{TX} = (\mathsf{SE.Enc}(\mathsf{ek}, \mathsf{tx}), t, in_1, in_2, \ldots, in_t)$
$\mathsf{TX}' = (\mathsf{SE.Enc}(\mathsf{ek}, \mathsf{tx}'))$
----------------------------------------------------------------
$\mathsf{tx} \leftarrow \mathsf{SE.Dec}(\mathsf{ek}, \mathsf{SE.Enc}(\mathsf{ek}, \mathsf{tx}))$
$\mathsf{tx}' \leftarrow \mathsf{SE.Dec}(\mathsf{ek}, \mathsf{SE.Enc}(\mathsf{ek}, \mathsf{tx}'))$

For $i$ from $1$ to $t$:
    Search $in_i$ in all record of income reference of previous transactions,
    (that is why $in_i$ not encrypted in TX) if successfully find one,
    then consider this transaction as double-spending attempt:
    { Return 0; HALT; }

Calculate $\mathsf{sum} \leftarrow \sum_{i=1}^{t} [\mathsf{val}_{\mathsf{ek}_{in_i}}(\mathsf{TX}_{in_i})]$
If any receipt of $in_i$ is not $\mathsf{PID}_A$: { Return 0; HALT; }

If $\mathsf{val}(\mathsf{tx}) + \mathsf{val}(\mathsf{tx}') > \mathsf{sum}$: { Return 0; HALT; }

Return 1
```

Figure 2: Committee validation as black box

4.2 User side

```
$\mathsf{txc} = (\langle \mathsf{TX}, \mathsf{TX}' \rangle, \mathsf{ek}, t, \mathsf{ek}_{in_1}, \mathsf{ek}_{in_2}, \ldots, \mathsf{ek}_{in_t})$
$(k', \psi) \leftarrow \mathsf{KEM.Enc}(pk_{com})$
$\mathsf{TXC} = (\psi, \mathsf{DEM.Enc}(k', \mathsf{txc}))$
----------------------------------------------------------------
$\mathsf{txp} = (\mathsf{tx}, \mathsf{ek})$
$(k_2', \psi_2) \leftarrow \mathsf{KEM.Enc}(pk_{com})$
$\mathsf{TXP} = (\psi_2, \mathsf{DEM.Enc}(k_2', \mathsf{txp}))$
```

Figure 3: Transaction command to committee and transaction proof to receipt

For Alice to pay Bob, it makes a transaction command txc and encrypts it with session key generated from committee public key pk_{com} in order to get TXC, then delivers it to the committee through a gossip network. At the same time, in order to prove to Bob, she makes transaction proof txp and sends $\mathsf{TXP} = (\psi_2, \mathsf{DEM.Enc}(k_2', \mathsf{txp}))$ to Bob, where pk_{bob} is public key of Bob.

After transaction goes through consensus of committee, Alice finds the index of this encrypted transaction pair $\langle \mathsf{TX}, \mathsf{TX}' \rangle$, which are denoted as $ix_1 = (R, l)$, $ix_2 = (R, l+1)$ (where R is the round of transaction, l and $l+1$ are adjacent sequence number). Then Alice sends ix_1 to Bob.

For Bob to check whether Alice has payed or not, it waits till receiving ix_1 from Alice, then he finds TX on chain according to the index.

Finally, he checks $\mathsf{tx} = \mathsf{SE.Dec}(\mathsf{ek}, \mathsf{TX})$ and finish confirmation.

5 Security Analysis of the Newly-proposed Consensus Scheme

Remark: In this paper, notation H might be a random oracle or a distribution, and when H denotes a distribution, then $\mathcal{O}_H$ denotes an oracle machine that returns a random element according to distribution H for each query.

Computational Indistinguishable. For any two distribution H_1, H_2, given oracle $\mathcal{O}_{H_1}, \mathcal{O}_{H_2}$ that return to each query one element from H_1,H_2, respectively. For any PPT distinguisher $\mathcal{A}$, there exists a negligible function $\epsilon(\kappa)$ that

$$\left| \Pr\left[b \in \{0,1\} \leftarrow \mathcal{A}^{\mathcal{O}_{H_1}}(1^\kappa) \middle| b = 1 \right] - \Pr\left[b \in \{0,1\} \leftarrow \mathcal{A}^{\mathcal{O}_{H_2}}(1^\kappa) \middle| b = 1 \right] \right| \leq \epsilon(\kappa).$$

Statistical Distance. For any set $\mathbb{S}$, any two distributions H_1, H_2 defined on $\mathbb{S}$, statistical distance of H_1, H_2, denoted as $\Delta[H_1; H_2]$, is defined as follows:

$$\Delta[H_1; H_2] := \frac{1}{2} \sum_{i \in \mathbb{S}} |\Pr[H_1 = i] - \Pr[H_2 = i]|$$

Lemma 5.1. *For any two distributions H_1, H_2, if statistical distance $\Delta(H_1; H_2) = 0$, then H_1 must be computationally indistinguishable from H_2, i.e. $H_1 \approx_c H_2$.*

Proof. From definition of statistical distance, if $\Delta(H_1; H_2) = 0$, then $H_1 = H_2$, since they are indistinguishable in sense of information theory, certainly $H_1 \approx_c H_2$. □

5.1 Safety of random number negotiation

5.1.1 Safety of commitment

Lemma 5.2. *For any cyclic group $\mathbb{G}$, and any generator $g \xleftarrow{\$} \mathbb{G}$, and any $k_1, k_2 \leftarrow \mathbb{Z}_p$, then $g^{k_1} = g^{k_2}$ if and only if $k_1 = k_2$.*

Proof. Suppose $g^{k_1} = g^{k_2}$, then we have $g^{k_1 - k_2} = 1_{\mathbb{G}}$, from property of cyclic group, we can get $k_1 = k_2$. Since sufficiency is obvious, we have proved this lemma. □

5.1.2 Randomness of generated number

Lemma 5.3. *For any cyclic group $\mathbb{G}$, and any generator $g \xleftarrow{\$} \mathbb{G}$, random number $r \xleftarrow{\$} \mathbb{Z}_p$, no committee member could get r from g^r in PPT time.*

This can be obviously derived from hardness of discrete logarithm.

Lemma 5.4. *As long as one of $\{r_1, r_2, \ldots, r_{\mathsf{csize}}\}$, say, r_1 is random number* ***independent*** *from choice of adversary, and PRF is pseudorandom function from $\{0,1\}^\kappa$ to $\{0,1\}^\kappa$, for any $R \in \{0,1\}^\kappa$, distribution of PRF$(\bigoplus_{j=1}^{\mathsf{csize}} r_j, R)$ will be computationally indistinguishable from random number in range $\{0,1\}^\kappa$ (regarding to random number r_1).*

Proof. Given $r' = \bigoplus_{j=2}^{\mathsf{csize}} r_j$, and random number r_1, we define random variables $H_{r',R}$ as output of $\mathsf{PRF}(r' \oplus r_1, R)$, H_R as output of $\mathsf{PRF}(r_1, R)$, where r_1 is random number over $\{0,1\}^\kappa$. And we define U as uniform distribution over $\{0,1\}^\kappa$.

From basic property of PRF,

$$H_R \approx_c U$$

And that,

$$\begin{aligned}
\Delta[H_{r',R}; H_R] &= \frac{1}{2} \sum_{s \leftarrow \{0,1\}^\kappa} |\Pr[H_{r',R} = s] - \Pr[H_R = s]| \\
&= \frac{1}{2} \sum_{s \leftarrow \{0,1\}^\kappa} \left| \Pr_{r_2 \leftarrow \{0,1\}^\kappa} [\mathsf{PRF}(r_2 \oplus r', R) = s] - \Pr_{r_1 \leftarrow \{0,1\}^\kappa} [\mathsf{PRF}(r_1, R) = s] \right| \\
&= \frac{1}{2} \sum_{s \leftarrow \{0,1\}^\kappa} \left| \Pr_{r_2 \leftarrow \{0,1\}^\kappa \oplus r'} [\mathsf{PRF}(r_2, R) = s] - \Pr_{r_1 \leftarrow \{0,1\}^\kappa} [\mathsf{PRF}(r_1, R) = s] \right| \\
&= \frac{1}{2} \sum_{s \leftarrow \{0,1\}^\kappa} \left| \Pr_{r_2 \leftarrow \{0,1\}^\kappa} [\mathsf{PRF}(r_2, R) = s] - \Pr_{r_1 \leftarrow \{0,1\}^\kappa} [\mathsf{PRF}(r_1, R) = s] \right| \\
&= 0
\end{aligned}$$

Hence, $H_{r',R} \approx_c H_R$;

Finally, it comes to $H_{r',R} \approx_c U$. □

In fact, security of committee key negotiation can be proved in the same way.

5.2 Motivation for Honesty

For committee members

In *Hybrid Consensus* , honesty of committee members is inherited from that of *Bitcoin*. However, *Hybrid Consensus* made the underlining assumption that most computing power must be in hands of *permanently honest* and *diligent* nodes. That is to say, *Hybrid Consensus* assumed that all honest nodes will always stay honest after being elected as committee member and that they will never get into state of *Sleepy* (see [BPS16]).

In practice, motivation in *Hybrid Consensus* and traditional *Bitcoin* merely guarantees honesty of miners during mining step. However, once elected as committee member and succeed in happily receiving block reward, nodes may become lazy or dishonest (vulnerable under target corruption) as committee members, at cost of nothing. Our construction can also grant motivation for members to stay honest after being elected as committee member.

In this construction, despite traditional block reward, we add up additional reward for committee members that stay honest and diligent as a committee member.

In order to facilitate description, we assume committee size csize is an even number. Committee members are of identity $(\mathsf{ID}_1, \mathsf{ID}_2, \ldots, \mathsf{ID}_{\mathsf{csize}})$.

At the end of each round, we stipulate that each committee members should finish the following process to get additional reward:

Step 1. Each member judges on performance of each committee members during this round. Then it constructs a bit-string s, in which i-th bit denotes whether i-th node is honest and diligent (in its perspective). After that, it broadcasts this bit-string appending its signature (and identity).

Step 2. Each member collects voting results broadcast by other members. And it determines who could be entitled with additional rewards (i.e. considered honest and diligent by most members).

Step 3. Committee members negotiate a random number r, with method same to that of committee election. (All broadcast content within this procedure should be encrypted with public key of all other members).

Step 4. All honest committee members make a special transaction $\mathsf{tx}_i = (0, ID_i \cdot g^r, m)$ for each ID_i in committee (who are considered honest and diligent by most members) and broadcast them along with its signature.

Since all committee members of current round know r, they will be able to refer to this transaction when needed. This special construction will be validated as long as signed by over $2/3$ committee members, likewise validation for ordinary transactions broadcast by committee with over $2/3$ signatures.

For users

For sender and receipt of transaction, sender has to behave honestly in order to make transaction on-chain (so that receipt can get her/him go). Also, receipt had better honestly behave when cooperating with sender to form signature, so that (s)he can spend income from this transaction later.

6 Conclusion and Discussion

We proposed SHChain: a privacy-preserving bitcoin protocol from rotating committee elected from fPoW, where fPoW is *fair-proof-of-work* firstly proposed in this paper. We've provided detailed description of this protocol and security analysis. In our future works, more formal proofs will be dedicated and adapted versions of fPoW that fit into various schemes will be sketched.

References

[BCG+14] Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014, pp. 459–474. IEEE Computer Society (2014)

[BGM16] Bentov, I., Gabizon, A., Mizrahi, A.: Cryptocurrencies without proof of work. In: Clark, J., Meiklejohn, S., Ryan, P.Y.A., Wallach, D.S., Brenner, M., Rohloff, K. (eds.) Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers, Lecture Notes in Computer Science, vol. 9604, pp. 142–157. Springer (2016)

[BMC+15] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W.: Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015, pp. 104–121. IEEE Computer Society (2015)

[BPS16] Bentov, I., Pass, R., Shi, E.: The sleepy model of consensus. IACR Cryptology ePrint Archive vol. 2016, p. 918 (2016)

[CL99] Castro, M., Liskov, B.: Practical byzantine fault tolerance. In: Seltzer, M.I., Leach, P.J. (eds.) Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 22-25, 1999, pp. 173–186. USENIX Association (1999)

[ES14] Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers, pp. 436–454 (2014)

[MGGR13] Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013, pp. 397–411. IEEE Computer Society (2013)

[Nak08] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. "www.bitcoin.org" (2008)

[POA16] Enhanced-proof-of-activity: Arbitrary combination between proof-of-work and proof-of-stake (2016)

[PS16a] Pass, R., Shi, E.: Fruitchains: A fair blockchain. IACR Cryptology ePrint Archive vol. 2016, p. 916 (2016)

[PS16b] Pass, R., Shi, E.: Hybrid consensus: Efficient consensus in the permissionless model. IACR Cryptology ePrint Archive vol. 2016, p. 917 (2016)

7 Appendix A

Key Negotiation for committee members

In our construction, committee members need to negotiate for two key pairs: $(\mathsf{PK}_{com}, \mathsf{SK}_{com})$ for public key encryption scheme, and (pk_{com}, sk_{com}) for hybrid encryption (KEM/DEM) scheme. The main difficulty of key generation is to generate random numbers, and such randomness should not be controlled by any dishonest member in committee. Similarly to the case in committee election, we give an outline of how to generate a random number. In fact, the only difference of this procedure (from that in committee election) is that all broadcasting content should be encrypted with public key of all committee members, to prevent eavesdropping.

We suppose R-round committee consists of $\mathsf{CM}_R = [\mathsf{ID}_1, \mathsf{ID}_2, \ldots, \mathsf{ID}_{\mathsf{csize}}]$, public key and secret key (in public key encryption scheme) for ID_i is PK_i and corresponding secret key is SK_i.

```
Generating random number $0 \le r < 2^\kappa$ on round $R$
(for member of identity $\mathsf{ID}_i$ , $1 \le i \le$ csize)
All broadcast content here should be encrypted with key of all other members

$\boxed{\mathsf{CM}_R = [\mathsf{ID}_1, \mathsf{ID}_2, \dots, , \mathsf{ID}_{\mathsf{csize}}]}$
Prime number $p > 2^\kappa$, and $g$ be one generator of $\mathbb{Z}_p^*$;
Choose randomly $r_i \leftarrow \{0,1\}^\kappa$;
Broadcast $('Prepare', \mathsf{ID}_i, g^{r_i})$;

Broadcast $('Commit', \mathsf{ID}_i, \mathsf{ID}_i)$;
Upon receiving $('Prepare', \mathsf{ID}_j, g^{r_j})$:
    Store $g^{r_j}$;
    Complain if receiving two different tuples from $\mathsf{ID}_j$;
    Broadcast $('Commit', \mathsf{ID}_i, \mathsf{ID}_j)$;
Keep pending till receiving $('Commit', \mathsf{ID}_k, \mathsf{ID}_j)$ for all $1 \le k, j \le$ csize;

For $j$ from $1$ to csize, if $j \neq i$:
    Broadcast $('Broadcast', \mathsf{ID}_i, \mathsf{ID}_j, \mathsf{PKE.Enc}(\mathsf{PK}_j, r_i))$;
Receive and decrypt all $r'_j$ $(j \neq i)$ from other members;
For $j \neq i$: if $g^{r'_j} \neq g^{r_j}$, then $\{$set $r_j \leftarrow 0$; Accuse $j$;$\}$ or else $r_j \leftarrow r'_j$;
$r \leftarrow \mathsf{PRF}(\bigoplus_{j=1}^{\mathsf{csize}} r_j, R)$
```

Figure 4: Generating random number for key generation

Similar to that of random number generation, term "`Accuse`" means to show denial during final voting process (mentioned in **Section** 5.2).

8 Appendix C

Why fPoW excels ordinary PoW given existence of delaying?

Lemma 8.1. *For any* $0 < c < 1$*, any natural number* N*:* $c \cdot \sum_{i=0}^{\infty}(1-c)^{(iN)} - \frac{1}{N} = o(\frac{1}{N})$.

Proof.

$$\begin{aligned} c \cdot \sum_{i=0}^{\infty}(1-c)^{(iN)} &= c \cdot \lim_{k\to\infty} \frac{1-(1-c)^{Nk}}{1-(1-c)^N} = \frac{c}{1-(1-c)^N} \\ &= \frac{c}{1-\left(1^N - \binom{N}{1}1^{N-1}c + o(c)\right)} \\ &= \frac{c}{Nc - o(c)} \end{aligned}$$

And then we get

$$\begin{aligned} c \cdot \sum_{i=0}^{\infty}(1-c)^{(iN)} - \frac{1}{N} &= \frac{c}{Nc-o(c)} - \frac{c}{Nc} \\ &= \frac{c \cdot o(c)}{(Nc - o(c)) \cdot Nc} = o(\frac{1}{N}) \end{aligned}$$

□

Lemma 8.2. *For any integers $\Delta > \delta > \delta' > 0$, any $0 < c < 1$, there exists sufficiently large N, s.t.*

$$\sum_{i=\delta}^{\infty}(1-c)^{i-\delta}\cdot c\cdot(1-c)^{(N-1)(i-\delta')} < \frac{\Delta-\delta}{\Delta-\delta'}\cdot\frac{1}{N}$$

Proof. We denote $d = \delta - \delta'$, hence $\delta' = \delta - d$;

$$\begin{aligned}\sum_{i=\delta}^{\infty}(1-c)^{i-\delta}\cdot c\cdot(1-c)^{(N-1)(i-\delta')} &= \sum_{i=\delta}^{\infty}(1-c)^{i-\delta}\cdot c\cdot(1-c)^{(N-1)(i-\delta+d)}\\ &= (1-c)^{(N-1)d}\cdot c\cdot\sum_{i=\delta}^{\infty}(1-c)^{N(i-\delta)}\\ &= (1-c)^{(N-1)d}\cdot c\cdot\sum_{i=0}^{\infty}(1-c)^{(iN)}\end{aligned}$$

we use previous lemma and get:

$$\begin{aligned}(1-c)^{(N-1)d}\cdot c\cdot\sum_{i=0}^{\infty}(1-c)^{(iN)} &= (1-c)^{(N-1)d}\cdot(\frac{1}{N}+o(\frac{1}{N}))\\ &\approx \frac{1}{N}(1-c)^{(N-1)d}\end{aligned}$$

Meanwhile,

$$\frac{\Delta-\delta}{\Delta-\delta'}\cdot\frac{1}{N} = \frac{\Delta-\delta}{\Delta-\delta+d}\cdot\frac{1}{N}$$

Since for sufficiently large N:

$$(1-c)^{(N-1)d} \ll \frac{\Delta-\delta}{\Delta-\delta+d}$$

And this lemma has been proved.

$$\sum_{i=\delta}^{\infty}(1-c)^{i-\delta}\cdot c\cdot(1-c)^{(N-1)(i-\delta')} < \frac{\Delta-\delta}{\Delta-\delta'}\cdot\frac{1}{N}$$

□

Given the lemma above, we now illustrate how an inequality proves fPoW excels PoW in sense of stability when network delaying exists.

In following discussion, for simplicity, we consider such case: we have N candidates sharing the same computing power, i.e., their expectation of timing of find one nonce solution in fPoW is T_s. We assume one of them suffers from certain network delaying and has to begin puzzle-solving at time δ, and all other nodes start puzzle-solving at time $\delta' < \delta$, and we denote Δ as ending of current round.

Then, in fPoW, the probability that the node suffering network delaying (say, Tom) would become new committee of next round is:

$$\gamma_1 = \frac{\mathbb{E}[\mathsf{sol}_\delta]}{(N-1)\cdot\mathbb{E}[\mathsf{sol}_{\delta'}]+\mathbb{E}[\mathsf{sol}_\delta]} = \frac{\frac{\Delta-\delta}{T_s}}{(N-1)\cdot\frac{\Delta-\delta'}{T_s}+\frac{\Delta-\delta}{T_s}} = \frac{\Delta-\delta}{N(\Delta-\delta')}+o(\frac{1}{N})$$

where sol_δ denotes number of nonce solutions to be found if starting puzzle-solving at time δ.

In the case that we want our election forking-free, with traditional PoW, we have to stipulate that first block mined should be the on-chain block. In this case, we take a glance at the probability of Tom entering committee next round in ordinary PoW scheme:

$$\gamma_2 = \sum_{i=\delta}^{\infty} (1-c)^{i-\delta} \cdot c \cdot (1-c)^{(N-1)(i-\delta')}$$

where c is the probability that one (since we assume they share same computing power) find a nonce within one unit of time.

When $\delta = \delta'$, from **Lemma 8.1**, we get $\gamma_1 - \gamma_2 = o(\frac{1}{N})$, which fits our scenario since all them share the same probability of entering committee next round is no delaying exists (or suffering exactly same delaying).

When $\delta' < \delta < \Delta$, then from **Lemma 8.2**, fPoW excels ordinary PoW in the sense of stability since it makes the damage of delaying less $\gamma_2 < \gamma_1$.