Forking-Free Hybrid Consensus with Generalized Proof-of-Activity

Shuyang Tang[1], Zhiqiang Liu*[1],
Sherman S. M. Chow[2],
Zhen Liu*[1], Yu Long*[1], and Shengli Liu*[1]

[1] Shanghai Jiao Tong University, China
[2] The Chinese University of Hong Kong, Hong Kong

Abstract. Bitcoin and its underlying blockchain mechanism have been attracting much attention. One of their core innovations, *Proof-of-Work* (PoW), is notoriously inefficient and potentially motivates a centralization of computing power, which defeats the original aim of decentralization. *Proof-of-Stake* (PoS) is later proposed to replace PoW. However, both PoW and PoS have different inherent advantages and disadvantages, so does *Proof-of-Activity* (PoA) of Bentov et al. (SIGMETRICS 2014) which only offers limited combinations of two mechanisms. On the other hand, the hybrid consensus protocol of Pass and Shi (ePrint 2016/917) aims to improve the efficiency by dynamically maintaining a rotating committee. Yet, there are unsatisfactory issues including selfish mining and fair committee election.

In this paper, we firstly devise a generalized variant of PoW. After that, we leverage our newly proposed generalized PoW to construct forking-free hybrid consensus, which addresses issues faced by a regular hybrid consensus mechanism. We further combine our forking-free hybrid consensus mechanism with PoS for a generalization of PoA. Compared with PoA, our generalized PoA improves the efficiency and provides more flexible combinations of PoW and PoS, resulting in a more powerful and applicable consensus scheme.

Keywords: Blockchain, Consensus, Cryptocurrency, Hybrid consensus, Practical Byzantine Fault Tolerance, Proof-of-Stake, Proof-of-Work

1 Introduction

Blockchain technique has been attracting much interest since Bitcoin [Nak08] was proposed in 2008, due to its valuable potential for building an anonymous, decentralized, and trustful network. It is considered to be commencing a revolution in information technology and economics. Bitcoin utilized blockchain (we refer to the original Bitcoin blockchain as "Nakamoto chain") as the building block of consensus. *Proof-of-Work* [TJ11], firstly proposed to prevent e-mail spamming [JJ99], was introduced to Bitcoin system to make sure any newly generated block is mined by an honest node with high probability. More precisely, blockchain requires the creator of a new block to solve a hash-collision problem (i.e. a hash puzzle) with regard to the hash of the previous block. When multiple new blocks are generated, disagreement emerges, and forking of chain happens. We need a way to make sure that all nodes concede to "the miner of the next block". Honest nodes only try to solve hash puzzles and build onto the latest blocks in the longest valid chain. The first forked branch that is followed by a certain number of blocks is considered to be confirmed. When a chain is confirmed, transactions within on-chain blocks are validated.

* Corresponding authors: {liu-zq, liuzhen, longyu, liu-sl}@cs.sjtu.edu.cn

Soundness of the blockchain system is guaranteed as long as the majority of computing power is at hands of honest nodes. Forking also tackles the misbehaviour of miners. Yet, to resolve forking, PoW-based protocols often confirm transactions at an unsatisfactory speed.

Serving as a core part of the original blockchain consensus protocol, PoW has several merits such as good robustness, elaborated incentive scheme which stimulates online presence of candidates, and openness to any participant. Yet, it also suffers from the poor efficiency which potentially motivates a centralization of computing power, as well as other risks related to "tragedy of the commons" (see more in [BLMR14]). To address these issues, *Proof-of-Stake* (PoS) [Qua11,BGM16] is proposed to replace PoW of the miners with computing power to stakeholders. Yet, PoS has defects for supporting online presence of stakeholders. *Proof-of-Activity* (PoA) [BLMR14] was proposed to achieve a certain combination of PoW and PoS so as to inherit their advantages. With subtle combinations of PoW and PoS, PoA determines the miner of a new block by the factors of computing power as well as stake value.

Practical Byzantine Fault Tolerance (PBFT) algorithm [CL99] proposed by Castro and Liskov has been widely adopted in blockchains. It serves as (a part of) the consensus scheme which substitutes PoW for better efficiency under certain circumstances. *Hybrid Consensus* [PS16b] combines PoW and PBFT. It utilizes Nakamoto chain or Fruitchain [PS16a] as the underlying blockchain (called "snailchain") to dynamically maintain a rotating committee, and all transactions are verified by the committee members via PBFT. A consensus is legitimate if over 1/3 of the committee members concur and broadcast corresponding signatures. With hybrid consensus, the efficiency of transaction confirmation can be improved significantly.

1.1 Related Works

Cryptocurrency has been attracting massive attentions since Nakamoto's proposal of Bitcoin blockchain [BMC^{+}15,TS16,Swa15]. Due to this, multiple decentralized currencies have been devised following this new trend [EGSR15,Kin13,SBRS16,WV16]. Efforts have been devoted to devising scheme for next generation cryptocurrency systems [EGSR15]. Solidus was proposed to attain an efficient cryptocurrency system [AMN^{+}16]. At the same time, blockchains adopting consensus schemes other than PoW have been designed [BGM16,BLMR14].

1.2 Issues and Motivations

Firstly, there are two fairness issues regrading to the original hybrid consensus:

- **Existence of forking.** Forking of the underlying snailchain exists, which wastes a great amount of time and energy, leading to security hazards such as the possibility of selfish mining (see [ES14]). Selfish mining exists only when forking is possible. That is the reason why hybrid consensus with Nakamoto chain as the underlying snailchain required 3/4 overall honesty rate instead of 2/3. In fact, forking is necessary in traditional blockchains for security guarantee of transaction confirmation. However, it is not the case with a rotating committee, since mining has nothing to do with the validation of transactions.
- **Accuracy of evaluations.** We will show in later part that, traditional PoW is not an accurate estimation of committee candidates' computing power, due to its great variance.
- **Lacking motivation for honesty.** Honesty of committee members is guaranteed from block reward and transaction fee. However, it merely guarantees honesty and diligence of nodes as committee candidates (i.e. miners), not that as committee members.

Moreover, PoA is also rather lacking due to the following reasons:

- **Existence of Forking and Efficiency issue.** Inherited from the classical blockchain construction, forking is still required to tackle with ambiguity. Also, PoA runs a classical blockchain instance as its underlying protocol. Due to this, its efficiency is not satisfactory.
- **Flexible and Formalizable Combination of PoW and PoS.** PoA offers one specific combination of PoW and PoS. Since cryptocurrency is a rapidly developing field, requirements of different scenarios will be various. For this reason, flexible combination of PoW and PoS should be attained. However, PoA can hardly be adapted to fit into multiple scenarios. Finally, the combination of PoW and PoS in PoA is hard to be formally described.

Being key components of blockchain, we aim to further improve hybrid consensus and PoA to derive more efficient, flexible, and applicable consensus constructions.

1.3 Our Contribution

We propose generalized PoW and a forking-free hybrid consensus scheme based on generalized PoW to present a novel way of ridding hybrid consensus of forking, and equip hybrid consensus with a voting-liked reward negotiation system to provide incentives for honesty. Our fork-free hybrid consensus is endowed with the following properties.

- **Forking-free PoW.** With our generalized PoWconstruction from Sec. 3, forking can be eliminated. Hence, selfish mining can be prevented and the security of the system can be enhanced. Also, massive resources can be prevented from being wasted in following “wrong” branches.
- **Accurate evaluations of computing power.** Generalized PoW is inherited with merits of less various in the evaluation of committee candidates’ computing power. In Sec. 3, we will prove that generalized PoW has smaller variance in the evaluation of committee candidates’ PoW capability.
- **Motivation for honesty.** In our work, we provide a voting-liked reward negotiation protocol that provides incentive of honesty for committee members. Committee members will loss transaction fees and block rewards if they fail to behave honestly and diligently during working time.

We then further improve PoA to make it more efficient and applicable. To do this, we propose generalized PoA a generalized construction of PoA by leveraging our fork-free hybrid consensus. Compared with the original PoA, our generalized PoA provides the following merits.

- **Forking-free Proof-of-Activity with efficiency.** In original work of PoA, forking is still necessary to tackle with the ambiguity during block mining. In our newly proposed generalized PoA, instead of running Nakamoto chain as the underlying protocol, a committee is dynamically maintained by a generalized PoW-based forking-free protocol (see Sec. 4.1) to validate transactions through a PBFT network, so as to achieve the forking-free property and efficiency.
- **Flexible and Formalizable Combination of PoW and PoS.** In contrast to the original work of PoA, we construct generalized PoA based on generalized PoW, which allows protocol users to flexibly choose how to achieve the combination of PoW and PoS. Also, our generalized PoA is easy to be formally analyzed. We will also discuss that “concave” choices of combinations are preferred.

Table 1. Comparisons between Consensus Schemes

Consensus Scheme	Efficiency	Forking-free	PoW	PoS	Incentive of Presence	Flexible Combination
Classical PoW [TJ11]			✓		✓	
Ideal PoS [Qua11]	✓	✓		✓		
Hybrid Consensus [PS16b]	✓		✓		✓	
Proof-of-Activity [BLMR14]			✓	✓	✓	
Forking-free Hybrid Consensus	✓	✓	✓		✓	
Generalized PoA	✓	✓	✓	✓	✓	✓

1.4 Paper Organization

The remainder of this paper is organized as follows. Sec. 2 introduces the notations and preliminaries. Sec. 3 proposes the concept of generalized PoW and argues about its merits, then proposes our fork-free hybrid consensus. In Sec. 4, we further combine fork-free hybrid consensus with PoS to form generalized PoA, and analyze different strategies and the security under different cases.

2 Notations and Preliminaries

Notations in this paper are illustrated in Table 2.

2.1 Bitcoin and Blockchain

Bitcoin was a novel online payment system invented by Satoshi Nakamoto [Nak08], it utilized the blockchain (in short, blockchain) (we may refer to this in term Nakamoto Chain) as the building block of consensus. In the bitcoin system, the network links transactions by time-stamping technique, and hashing them into a chain of hash-based proof-of-work. In such a way, the history that cannot be tampered without redoing the proof-of-work is formed. Blockchain requires the generator of new block, say, $\boldsymbol{B}_i$, to solve a hash problem, that is to find a nonce (nc), so that $H(\boldsymbol{B}_{i-1}, \mathsf{nc}) \in \mathsf{target}$, where target is a predetermined short range, and $\boldsymbol{B}_{i-1}$ denotes the previous block. Disagreement happens when multiple blocks are mined simultaneously. In this case, forking of the chain happens. Honest nodes always build blocks following the longest valid chain. The first forked branch followed by certain number of blocks is considered to be confirmed.

2.2 Practical Byzantine Fault Tolerance (PBFT)

In distributed computing systems, Byzantine fault tolerance (BFT) is the system that tolerates certain failures in Byzantine generals problem. BFT protocols have been extensively studied in the area of distributed system [PSL80,LSP82,TPS87,CL99]. Among these protocols, *Practical Byzantine Fault Tolerance* (PBFT) algorithm [CL99] proposed by Castro and Liskov provides high-performance Byzantine state machine replication, and has been widely implemented in both public-chain and private-chain cryptocurrency systems.

2.3 Hybrid Consensus

Pass and Shi's hybrid consensus [PS16b] is a new consensus scheme which dynamically maintains a rotating committee. Committee members of each day corresponds to miners of a fixed interval of confirmed on-chain blocks. In hybrid consensus, the blockchain provides Proof-of-Work, but not the validation of transactions, and transactions are validated through a PBFT network among committee members. In such a way, confirmation time of transactions is bounded by actual network delays instead of the theoretical delay, and efficiency can be significantly improved.

2.4 Proof-of-Activity

Bentov et al.'s *Proof-of-Activity* (PoA) [BLMR14] allows miners to solve a hash puzzle according to the previous block and form the head of the next block. Then according to the block head broadcast by this lucky miner, N lucky stakeholders are determined to share transaction fee together with this lucky miner. N stakeholders broadcast signatures of this block head, and N-th lucky stakeholder should wrap all newly received and verified transactions into this block and broadcast it. After that, this block can be considered as a new block on-chain. In this scheme, stake holders have to stay active to perform the protocol and earn rewards.

Table 2. Table of Notations

κ, λ	security parameters
Δ	the upper bound of network delaying
R	a round number (similar to the notion of "*date*" in hybrid consensus [PS16b])
T	the maximum number of trial attempts in puzzle-solving for one user (per round)
M	the cardinality of the total range of the hash function
M_0	the cardinality of the acceptable range of nonce's hash value
csize	the size of the rotating committee, $\mathsf{csize} := \Theta(\lambda)$
N	the total number of candidates running for next day's committee member
$\boldsymbol{B}_R$	the block content for round R
target	the target set of the hash puzzle
ID_i	the public identity for node i
commit_i	a commitment for node i
rec_R	the transaction record and the nonce record of round R
nc	a nonce value
α	the upper bound of the total fraction of computing power held by the adversary
β	the upper bound of the total fraction of stakes held by the adversary
(w_i, s_i)	PoW capability and stake value for node i
(x_i, y_i)	PoW capability and stake value for node i normalized from (w_i, s_i) (so that x_i and y_i share the same expectation μ)
$L = G(x, y)$	a weight assigned to a candidate of normalized PoW capability x and normalized stake value y, which corresponds to the possibility of entering committee
com_i	the identity (i.e. public key) of i-th committee member
$cand_i$	the identity (i.e. public key) of i-th committee candidate
CM_R	$\mathsf{CM}_R = \{com_1, com_2, \ldots, com_{\mathsf{csize}}\}$ is the identity list of round-R's committee members (ordered by the time of entering the committee)
CD_R	$\mathsf{CD}_R = \{cand_1, cand_2, \ldots, cand_N\}$ is the identity list of round-R's candidates
$\mathsf{PRF}(k, R)$	a pseudorandom function that takes a key k and a round number R as input and returns a pseudorandom bit-string in $\{0,1\}^\kappa$, interpreted as a natural number in $\mathbb{Z}_{2^\kappa}$

3 Generalized Proof-of-Work and Forking-free Hybrid Consensus

3.1 Generalized Proof-of-Work

In hybrid consensus, for each round, transactions are validated through a PBFT network among committee members. However, the committee election in hybrid consensus is based on the traditional PoW, hence is not forking-free (which we will unfold in the following contents). To improve this, we propose our generalized proof-of-work, to derive a forking-free mechanism.

In the traditional construction, for each miner, *the probability of mining a nonce* for each block is roughly proportional to its computing power. In our generalized proof-of-work, we lower the difficulty of the mining puzzle to make *the expected number of nonces found* proportional to its computing power. For generalized PoW, the difficulty of the nonce-puzzle is smaller than that of PoW. In each round, each candidate u finds some nonce solutions, say, $\mathsf{nc}_{u,1}, \mathsf{nc}_{u,2}, \ldots, \mathsf{nc}_{u,P_u}$, where P_u is the number of nonces found by candidate u. Before the end of this round, each candidate submits all solutions it found to the committee, then committee members arrange all received solutions in an array W, and decide a random number $1 \leq r \leq |W| = \sum_u P_u$. Finally, committee announces the identity of a new committee member of next round, who is the miner corresponding to the r-th item of W.

In general, we hope that the committee can accurately have access to PoW capability w_i of each candidate, and hope that w_i will be nearly proportional to their real computing power and wealth, with less variance. We now give three protocols for the proof of candidates' computing power. In fact, the expected w_i's under three protocols can be regarded as proportional to candidates' computing power, so we make comparisons on their coefficients of variance and finally determine that our third construction will be more satisfiable.

To simplify the formalization, here we suppose one candidate tries the hash puzzle for T times in total, the total range of the hash function is of cardinality M, and the difficulty is properly adjusted so that the acceptable range is of cardinality M_0.

Traditional PoW. In the traditional construction, we set the puzzle difficulty very high and ask each candidate i to find a puzzle solution. If one candidate successfully finds a solution, then its w_i is 1, or else w_i is 0.

In traditional PoW, we assume $T \cdot M_0 \ll M$. The expectation of w_i is thus proportional to the computing power T:

$$\mathbb{E}[w_i] = \Pr[w_i = 1] = 1 - (1 - \frac{M_0}{M})^T \approx T \cdot \frac{M_0}{M},$$

hence

$$\mathrm{Var}[w_i] = \mathbb{E}[w_i](1 - \mathbb{E}[w_i]) \approx \frac{T \cdot M_0}{M}(1 - \frac{T \cdot M_0}{M}).$$

And the coefficient of variance is

$$C_v[w_i] = \frac{\sqrt{\mathrm{Var}[w_i]}}{\mathbb{E}[w_i]} \approx \sqrt{\frac{M - TM_0}{TM_0}} \approx \sqrt{\frac{M}{TM_0}} \gg 1.$$

We can see that the coefficient of variance is very great in the traditional PoW.

Generalized PoW. For generalized PoW, we lower the difficulty so that a candidate with considerable computing power may find more than one solutions to one hash puzzle. Final w_i will be the number of solutions it found. The expected number of solutions one candidate i with T computing power may find is

$$\gamma := \mathbb{E}[w_i] = T \cdot \frac{M_0}{M}.$$

We use X_j to denote a random variable that is 1 if j-th puzzle-solving attempt works, and 0 otherwise. We have

$$\mathrm{Var}[w_i] = \sum_{j=1}^{T} \mathrm{Var}[X_j] = T \cdot \frac{M_0}{M}(1 - \frac{M_0}{M}) = \gamma(1 - \frac{M_0}{M}),$$

and so

$$C_v[w_i] = \frac{\sqrt{\mathrm{Var}[w_i]}}{\mathbb{E}[w_i]} = \frac{\sqrt{\gamma(1 - \frac{M_0}{M})}}{\gamma} \approx \sqrt{\frac{1}{\gamma}}.$$

In conclusion, generalized PoW is endowed with a smaller coefficient of variance.

3.2 Merits of Generalized PoW

Merits of generalized PoW consists of three parts:

1. **Forking-freeness.** When forking occurred, many miners might waste massive computing resources by following wrong branches of the blockchain. Also, a longer confirmation time is needed due to the possibility of forking. Moreover, forking leads to security risks like selfish mining. Without forking, our construction is more resource-saving, efficient, and secure.
2. **Friendliness in face of delays.** We find that generalized PoW is more friendly to nodes facing constant network delays, which will be partially proved in the appendix.
3. **Fair evaluation of computing power.** In fact, generalized PoW provides satisfactory accuracy in the evaluation of computing power. To show this, we offer three PoW games that assign a PoW "score" w_i to each candidate miner i who is trying to win the game (i.e. get the highest "score"). Next, we compare between three constructions and find that generalized PoW is endowed with less coefficient of variance.

3.3 Forking-free Hybrid Consensus

Based on hybrid consensus, in which a rotating committee is elected from the underlying blockchain and transactions are validated via PBFT among this committee, we construct fork-free hybrid consensus where generalized PoW is implemented to replace the underlying PoW. In our construction, we view the list of committee members as a queue, one node enters the committee and one leaves for each "round" (similar to what is called "*Day*" in hybrid consensus). Now we present the generalized PoW-based committee election protocol of fork-free hybrid consensus.

Candidates In round R, one candidate, say, Tom, collects transaction and nonce records of round $R - 1$ (signed by over $1/3$ committee members) rec_{R-1}, then finds as much as possible nonce(s) $\mathsf{nc}_{tom,1}, \mathsf{nc}_{tom,2}, \ldots, \mathsf{nc}_{tom,P_{tom}}$ such that

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

where $\boldsymbol{B}_{R-1} := \{\mathsf{rec}_{R-1}, H(\boldsymbol{B}_{R-2}), \mathsf{CM}_{R-1}\}$ is the block of the previous round. Note that differently from traditional Bitcoin blockchain, rec_{R-1} here includes users' transactions handled by round $R-1$'s committee, reward transactions for round $R-1$'s committee (which will be specified later), and all accepted nonces during round $R-1$. CM_{R-1} is the identity list of previous round's committee members. The hash value of the previous block is included in the block content so as to form the hard-to-change time-stamping.

Tom arranges all nonces found into W_{tom}:

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

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

In practice, it is not a good idea to submit nonce(s) to all committee members and assume they will all receive the same number of nonce(s) during the whole interval of round R. The consensus on the nonce-acceptance should be reached through another PBFT network. However, to simplify the representation, we merely assume that a safe submission protocol exists.

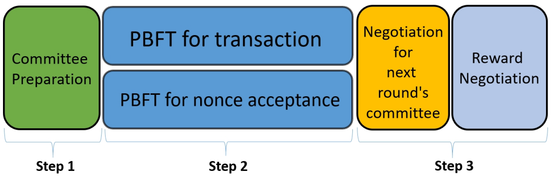

Fig. 1. Basic construction for each round

Current committee member For simplicity, we order all committee members in $1, 2, \ldots, \mathsf{csize}$. Each honest committee member receives W_u's from all candidates, puts all W_u's into W, and sorts all items in the same order, to get

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

At the termination of this round, committee members in $\mathsf{CM}_R = [\mathsf{ID}_1, \mathsf{ID}_2, \ldots, \mathsf{ID}_{\mathsf{csize}}]$ produce a random number $1 \le r \le |W| = \sum_u P_u$. After that, all members broadcast r_j. In such a way, the adversary can control nothing about the generated random number, as long as any one committee

```
Generating random number $1 \le r \le |W|$ on round $R$
(for member of identity $\mathsf{ID}_i$ , $1 \le i \le \mathsf{csize}$)
•Broadcast commitment $\mathsf{commit}_i := H(r_i)$ (and its signature);
•Wait for $\Delta$ and receive $\mathsf{commit}_j := H(r_j)$ broadcast by other members $j \neq i$;
•Broadcast $r_i$ (and its signature) towards all other members;
•Wait for $\Delta$ and receive all $r_j$ $(j \neq i)$ from other members;
•For $j \neq i$: if either $r_j$ or $\mathsf{commit}_j$ not received, then $\{$set $r_j \leftarrow 0$; Accuse $j$;$\}$
•For $j \neq i$: if $H(r_j) \neq \mathsf{commit}_j$, then $\{$set $r_j \leftarrow 0$; Accuse $j$;$\}$
•Get $r \leftarrow 1 + \lceil \Big(\mathsf{PRF}(\bigoplus_{j=1}^{\mathsf{csize}} r_j, R)\Big) \cdot \frac{|W|}{2^{\kappa}} \rceil$
```

Fig. 2. Generating a random number for the committee election

member is honest. Details are shown in Fig. 2. By the term "*Accuse*", we mean the action to vote for denial during the final reward negotiation.

Finally, committee members broadcast rec_R and declare ("Enter", ID') along with their signatures, where ID' is the identity of the miner of the r-th nonce. This lucky candidate is enrolled into the committee of next round if this declaration is signed by over 1/3 current committee members. After that, the reward negotiation begins.

3.4 Bootstrapping Techniques

To bootstrap the system, we need csize genesis blocks maintained by the system creator. Differently from Bitcoin, the system creator (i.e. the maintainer of the csize genesis blocks) should have certain computing power to perform the consensus for the first csize rounds.

3.5 Determination on Commencement and Termination Time

All users' transaction commands and candidates' nonces are submitted to the committee, and committee members reach the consensus through PBFT. PBFT is an ordered procedure during which transaction commands and nonces are proposed by PBFT participants (i.e. committee members) sequentially in turn. With this property, we can stipulate that each round is terminated at the time of M^{th} proposal within the PBFT process, where M is a predetermined parameter.

3.6 Incentive of Honesty

In the original construction of hybrid consensus, the incentive of participants' honesty is inherited from that of Bitcoin. This mechanism offers the motivation for honesty during candidates' mining process. However, it ignored the incentive for honesty and presence of members after being elected into the committee. To guarantee honesty and presence of committee members, we devise a voting-liked mechanism. In detail, we design reward transactions with a specially designed structure to reward honest and diligent members, thereby providing incentives of honesty. At the termination of each round, each committee member sends out reward transactions for each other members, and appends proper signatures to reward transactions that belong to those who acted honestly and diligently (not in the blacklist) in this round. Each reward transaction becomes legitimate as long as over 1/3 members broadcast signatures on this transaction. Reward transactions should have specially designed structures so that they can be validated without specifying payers. More specifically, this *reward negotiation* procedure proceeds as follows:

1. At the termination of each round, each committee member sets reward for each honest committee member as $S_{reward} = \frac{S_{\mathsf{tx}}+S_{block}}{\mathsf{csize}}$, where S_{tx} denotes the total amount of transaction fee included in this round (all honest nodes should have reached the consensus on this amount after PBFT) and S_{block} stands for the predetermined amount of block reward.
2. For each committee member (say, member i), it generates and signs on the reward transaction tx_j (whose receipt is member j, containing reward amount S_{reward}) for each honestly behaved member j. Then, all reward transactions are broadcast along with corresponding signatures.
3. Similarly to the case of ordinary transactions, for each committee member (say, member i), reward transaction tx_i is validated as long as over $1/3$ committee members broadcast tx_i along with proper signatures.

After reward negotiation, committee members broadcast rec_R and declare the terminate this round, along with proper signatures.

3.7 Merits of Forking-free Hybrid Consensus

Compared with the original hybrid consensus using Nakamoto chain as the underlying blockchain, our proposed fork-free hybrid consensus has the following advantages:

- **Forking-freeness.** One issue of hybrid consensus is that, forking still happens in the competition for the block mining game, unless we can revise the rule and stipulate that blocks should be appended with their broadcast time and the firstly broadcast one should be the next block. This is impractical, since it is hard for all nodes to share the same clock, and nodes have no reason to behave honestly when appending the broadcast time. In our construction, we need neither guarantees on time synchronization nor honesty of nodes, to achieve a fair competition for "the *first* miner of next block" without the existence of forking.
- **Accuracy in computing power evaluation.** Apart from the forking-free property, our construction is endowed with better fairness with respect to the evaluation of miners' computing power.
- **Friendliness in face of delays** Our construction guarantees on better fairness on candidates suffering from network delays. In the appendix, we will give a basic proof to show that our newly proposed construction excels ordinary PoW considering the existence of network delays.
- **Looser assumption against mildly agile corruption.** In hybrid consensus, the adversary is allowed to perform the mildly agile corruption, i.e., they can choose nodes to corrupt according to the configuration of the environment. τ-agility, which means an adversary has to wait for time τ to corrupt an honest node, is defined to describe the assumption on the adversary's capability. In our work, the assumption on τ can be much looser than that required for hybrid consensus.
- **Selfish mining prevention.** In our forking-free construction, selfish mining has no reason to work. Without potential selfish mining, a looser security assumption is required to attain the same security property [ES14].
- **Tolerated corruption.** In hybrid consensus, 3/4 overall honesty has to be guaranteed (if the underlying snailchain is Nakamoto chain) to achieve the 2/3-chain quality, due to the existence of selfish mining. In this work, we merely require a 2/3 overall honesty to achieve a 2/3-chain quality, so as to assure the security of PBFT.

4 Generalized Proof-of-Activity

We generalize proof-of-activity to support flexible combinations of generalized PoW and PoS. More specifically, we build our consensus protocol based on the framework of hybrid consensus with fair proof-of-work, by a novel way of rotating committee election, i.e., for a candidate with PoW capability w and stake value s, a function $G(w, s)$ can be established to assign a weight L to each candidate that reflects PoW capability w and stake value s.

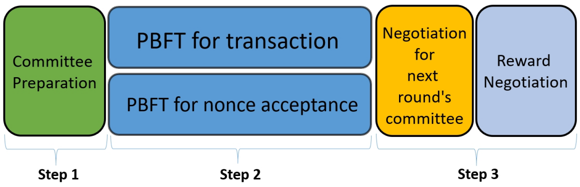

Fig. 3. Basic construction for each round

4.1 Detailed Protocols

We formally present the protocol of miners from the beginning of each round to the commencement of next round, in its entirety. We will discuss protocols for candidates and committee members separately, detailed illustrations of protocols are shown in Table 3 and Table 4. We suppose the set of committee members of round R is $\mathsf{CM}_R = \{com_1, com_2, \ldots, com_{\mathsf{csize}}\}$, and the set of candidates is $\mathsf{CD}_R = \{cand_1, cand_2, \ldots, cand_N\}$. To facilitate representation, we will use the term "committee member i" or "candidate i" together with "com_i" or "$cand_i$" interchangeably, in fact, they stand for the same meaning.

In the following, we refer to a protocol instance of either a committee member or a candidate as a *node*. In generalized PoW, the value of PoW capability w is often a small number, while the stake value s is often greater. For example, one node with stake value of 10^5 (in US dollar) may find at most one or two nonce(s). It is not reasonable to set $w = 2$ and $s = 10^5$. For this reason, we need to normalize w, s before calculating $G(w, s)$. In this section, we assume that all w's and s's are normalized to x's and y's so that

$$x = \frac{\mu}{\mathbb{E}[w]} \cdot w \propto w,$$
$$y = \frac{\mu}{\mathbb{E}[s]} \cdot s \propto s,$$

and then $\mathbb{E}[x] = \mathbb{E}[y] = \mu$ holds (the expectation is taken among all candidates). We consider x, y as continuous variables over $\mathbb{R}^+$. Furthermore, y should be greater than 1 when the logarithm exists, this makes sense since stakeholders should have a certain stake. In such a way, descriptions and derivations can be simplified.

Candidate In round R, for a candidate i who tries to enter the committee of the next round. It performs operations as follows:

1. It packs rec_{R-1}, together with the hash value of block $\boldsymbol{B}_{R-2}$ (to form time-stamp) and the list of committee members of previous round CM_{R-1}, into the block of round $R-1$ — $\boldsymbol{B}_{R-1}$.
2. It tries to find as much as possible nonce(s) $\mathsf{nc}_1, \mathsf{nc}_2, \ldots, \mathsf{nc}_\ell$ so that $H(\boldsymbol{B}_{R-1}, \mathsf{ID}_i, \mathsf{nc}_j) \in \mathsf{target}$ for all $1 \le j \le \ell$; And then, it submits $\{\mathsf{nc}_1, \mathsf{nc}_2, \ldots, \mathsf{nc}_\ell\}$ to committee members, appended with proper signatures.
3. It receives rec_R (with signatures from over 1/3 committee members) at the end of the round.

Table 3. Switchover techniques in the candidate side

Candidate Side (in round R, for candidate i)
•Pack $\boldsymbol{B}_{R-1} := \{\mathsf{rec}_{R-1}, H(\boldsymbol{B}_{R-2}), \mathsf{CM}_{R-1}\}$;
•Try to find as much as possible nonce(s) $\mathsf{nc}_1, \mathsf{nc}_2, \ldots, \mathsf{nc}_\ell$, so that $H(\boldsymbol{B}_{R-1}, \mathsf{ID}_i, \mathsf{nc}_j) \in \mathsf{target}$ for all $1 \leq j \leq \ell$;
•Submit $\{\mathsf{nc}_1, \mathsf{nc}_2, \ldots, \mathsf{nc}_\ell\}$ to committee members (appended with proper signatures);
•Collect validated transactions into rec_R, including reward transactions (signed by over 1/3 committee members);

Committee member Now we present the protocol for committee members in round R.

1. Each miner checks the committee list of the current round, and performs the following procedures if its identity is included in the list. Then, it packs $\boldsymbol{B}_{R-1}$.
2. Committee members run two PBFT instances, one for the consensus on transaction validation, one for the consensus on nonce-acceptance. At the same time, they calculate normalized PoW capabilities and stake values of each candidate (i.e. x_j and y_j for each candidate j).
3. Before the termination of round R, each committee member calculates $L_j := G(x_j, y_j)$ for each candidate j. And then they negotiate a uniform random number k_R, and decide one lucky candidate according to k_R. Finally, they produce reward transactions for each committee members, and sign on each reward transaction if the corresponding member is honest and diligent. Same as ordinary transactions, each reward transaction will be validated if over 1/3 fraction of committee members have signed on it.
4. It broadcast rec_R declare termination of this round, along with signatures.

To agree upon a uniform random number k_R, each committee member uniformly picks a random number and firstly send out a commitment (e.g., Pedersen commitment) of it. After receiving all committed random numbers, they reveal their random numbers. k_R will be set to an xor-summation of all these random numbers. Even when all but one members are dishonest, k_R will remain random. Table 4 shows the detailed procedures.

4.2 Strategy and Security Analysis

Definition 1. *We say function $G : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ is **concave** if and only if this holds: For any $\boldsymbol{v}, \boldsymbol{v'} \in (\mathbb{R}^+)^2$, it always holds that $G(\boldsymbol{v}) + G(\boldsymbol{v'}) \leq G(\boldsymbol{v} + \boldsymbol{v'})$.*

We will discuss the following two cases separately: one for a concave establishment of G and one for otherwise. The strategy of the adversary will be different in two cases. In concave case, nodes will prefer to aggregate their computing power and stake values to form stronger PoW power and maximize the possibility of being elected. On the other hand, in the non-concave case, dishonest nodes tend to divide its computing power and stake and multiple identities it spawned, to maximize the total probability of being elected. In this case, a heavy network burden would be caused. For this reason, we suggest that function $G(x, y)$ should be concave.

A non-concave case

We begin with an example of non-concave $G(x, y)$ to show that miners tend to split their PoW capabilities and stake values into different forked identities, to maximize the total probability of entering the committee of the next round. For this reason, a heavy network burden is caused.

Table 4. Switchover techniques in the committee side

Committee Side (in round R, for committee member i)
Step 1 •Check its identity in round-R committee list CM_R; •Pack $\boldsymbol{B}_{R-1} := \{\mathsf{rec}_{R-1}, H(\boldsymbol{B}_{R-2}), \mathsf{CM}_{R-1}\}$;
Step 2 •Run a PBFT instance for transaction validation; •Run a PBFT instance to reach consensus on candidates' nonce submission; •Collect w_j as the number of satisfiable nonce(s) submitted by candidate j; •Collect s_j which is the total stake held by candidate j; •Normalize (w_j, s_j) into (x_j, y_j) for each candidate j;
Step 3 •Calculate $L_j := G(x_j, y_j)$ for each candidate j; •Calculate $\mathsf{sum}_L := \sum_{j \in \mathsf{CD}_R} L_j$; •Pick a random number $r_i \xleftarrow{\$} \{0,1\}^\kappa$; •Broadcast $\mathsf{commit}_i := H(r_i)$ (along with a proper signature); •Wait for time Δ, and receive commit_j from each committee member j; •Broadcast r_i (along with a proper signature); •Wait for time Δ, and receive r_j from each fellow committee member j; •For each $j \in \mathsf{CM}_R$: if either r_j or commit_j not received: set $r_j \leftarrow 0$ and put j into the blacklist; •For each $j \in \mathsf{CM}_R$: check whether $\mathsf{commit}_j = H(r_j)$, put j into the blacklist if not; •Calculate $k_R \leftarrow \bigoplus_{j \in \mathsf{CM}_R} r_j$; •Calculate $\mathsf{rand} \leftarrow \mathsf{PRF}(k_R, R) \cdot (\mathsf{sum}_L / 2^\kappa)$; •Find first k that $\sum_{j=1}^{k-1} L_j \leq \mathsf{rand} < \sum_{j=1}^{k} L_j$; •Claim member list of the next round is $\mathsf{CM}_{R+1} = \{com_2, com_3, \ldots, com_{\mathsf{csize}}, cand_k\}$; •Generate reward transactions tx_j for each member $j \in \mathsf{CM}_R$; •Sign on tx_j and broadcast it if j worked honestly, diligently and is not in the blacklist; •Broadcast rec_R along with a proper signature.

We consider $G(x, y) = \ln(xy)$. We assume that $x, y \geq 1$ always holds. In this case, the adversary would split its x, y's into several spawned nodes to maximize the total probability of being elected. Suppose one candidate holds computing capability x', total stake y', and splits x', y' evenly into ℓ forked nodes. We show that the probability of entering the committee in next round is maximized when ℓ reaches some value greater than 1 (i.e. division of x' and y' exists).

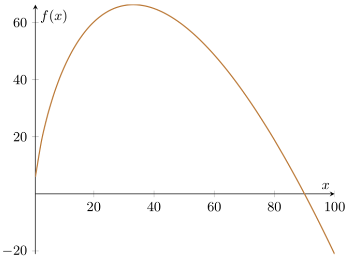

Fig. 4. Function $f(x) = x(a - 2 \ln x)$ with $a = 9.0$

In fact, under our assumption, this candidate tends to maximize

$$\ell \cdot \ln(\frac{x'}{\ell} \cdot \frac{y'}{\ell}) = \ell \cdot (\ln(x'y') - 2 \ln \ell).$$

After simple derivations, we can see that this reaches the maximum when ℓ approaches $e^{\frac{\ln(x'y')-2}{2}}$, which is often much greater than 1. Hence, we can see that in non-concave case, miners tend to split their total resource into multiple spawned nodes. Since a heavy network burden might be caused in this way, it is suggested to avoid non-concave choice of $G(\cdot,\cdot)$. We can imagine non-concave functions that do not suffer (or not suffer much) from such attacks, but they should be carefully analyzed before implementing.

We define the adversary advantage $\mathbf{Adv}_{\alpha,\beta}$ to be the approximate possibility of a maliciously spawned node entering the committee of next round:

$$\mathbf{Adv}_{\alpha,\beta} = \frac{G(\alpha \cdot \mathbb{E}[\sum_{i=1}^{N} x_i], \beta \cdot \mathbb{E}[\sum_{i=1}^{N} y_i])}{\mathbb{E}[\sum_{i=1}^{N} G(x_i, y_i)]},$$

where N is the total number of nodes, α is the fraction of total computing power held by the adversary, and β is the fraction of total stakes held by the adversary.

Foundation of generalized PoA security is based on the 2/3 overall honesty among all committee members. That is to assure that $\mathbf{Adv}_{\alpha,\beta}$ should be small enough. Since the further calculation of $\mathbf{Adv}_{\alpha,\beta}$ highly depends on the choice of $G(\cdot,\cdot)$, in the following context of this section, we will propose three recommended establishments of $G(x,y)$, and present security analyses (i.e., calculation of $\mathbf{Adv}_{\alpha,\beta}$) separately.

Case A. Considering PoW capability and stake value evenly

When we consider PoW and PoS evenly (i.e. of same significance), we may set $G(x,y)$ as $\frac{x+y}{2}$, or $\sqrt{\frac{x^2+y^2}{2}}$. However, we hope to make the adversary harder to reach a high $G(x,y)$ value. It would be easier to have a high x value or high y value, but harder to make both x and y great enough. For this reason, we want $G(x,y)$ to be a function that can hardly reach a great value when either x or y is not great enough, and this function must be symmetric. Hence, we set $G(x,y) = \sqrt{xy}$.

We first prove that this evaluation function $G(x,y) = \sqrt{xy}$ is concave.

For any $(x_1,y_1),(x_2,y_2) \in \mathbb{R}^+ \times \mathbb{R}^+$:

$$G(x_1+x_2, y_1+y_2) \geq G(x_1,y_1) + G(x_2,y_2)$$

For any $(x_1,y_1),(x_2,y_2) \in \mathbb{R}^+ \times \mathbb{R}^+$:

$$x_1y_2 + x_2y_1 \geq 2\sqrt{x_1x_2y_1y_2}$$

this can be derived from the basic mean value inequality. From here,

$$\begin{aligned} x_1y_1 + x_1y_2 + x_2y_1 + x_2y_2 &\geq (\sqrt{x_1y_1})^2 + (\sqrt{x_2y_2})^2 + 2\sqrt{x_1x_2y_1y_2}, \\ \sqrt{(x_1+x_2)(y_1+y_2)} &\geq \sqrt{x_1y_1} + \sqrt{x_2y_2}, \end{aligned}$$

hence $G(x_1+x_2, y_1+y_2) \geq G(x_1,y_1) + G(x_2,y_2)$ always holds.

After that, we estimate the probability of the adversary being elected.

$$\begin{aligned} \mathbf{Adv}_{\alpha,\beta} &= \frac{G(\alpha \cdot \mathbb{E}[\sum_{i=1}^{N} x_i], \beta \cdot \mathbb{E}[\sum_{i=1}^{N} y_i])}{\mathbb{E}[\sum_{i=1}^{N} G(x_i, y_i)]} \\ &= \frac{\sqrt{\alpha\mathbb{E}[N]\mathbb{E}[x] \cdot \beta\mathbb{E}[N]\mathbb{E}[y]}}{\mathbb{E}[N] \cdot \mathbb{E}[\sqrt{xy}]} = \frac{\sqrt{\alpha\mathbb{E}[x] \cdot \beta\mathbb{E}[y]}}{\mathbb{E}[\sqrt{xy}]} = \frac{\sqrt{\alpha\beta} \cdot \mu}{\mathbb{E}[\sqrt{xy}]}. \end{aligned}$$

We can see that the advantage of the adversary will be limited to $\sqrt{\alpha\beta}$ within a multiplicative constant factor. Our detailed analyses in the appendix will show that this construction is feasible.

Case B. More considerations on PoW capability

Under certain environments, PoW capability should be more significant than stake during the committee election. Under such consideration, we can set $G(x, y) = x \ln y$, where is x is the normalized PoW capability and y is the normalized stake value. In case that $G(x, y) = x \ln y$, miners do not have to be rich enough (i.e. have a great stake value) to reach a high value of $G(w, s)$, but they should have some.

It is easy to see that this evaluation function $G(x, y) = x \ln y$ is also concave, since for any $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^+ \times \mathbb{R}^+$:

$$G(x_1 + x_2, y_1 + y_2) = (x_1 + x_2) \ln(y_1 + y_2) > x_1 \ln y_1 + x_2 \ln y_2 = G(x_1, y_1) + G(x_2, y_2).$$

We assume $y \geq 1$ always holds since candidates should hold some stake. For the probability of the adversary entering the committee, we have

$$\mathbf{Adv}_{\alpha,\beta} = \frac{\alpha \cdot \mathbb{E}[\sum_{i=1}^{N} x_i] \cdot \ln(\beta \cdot \mathbb{E}[\sum_{i=1}^{N} y_i])}{\mathbb{E}[N] \cdot \mathbb{E}[x \ln y]} = \frac{\alpha \cdot \mathbb{E}[N] \cdot \mathbb{E}[x] \cdot \ln(\beta \cdot \mathbb{E}[N]\mathbb{E}[y])}{\mathbb{E}[N] \cdot \mathbb{E}[x \ln y]} = \frac{\mu\alpha \ln(\mu\beta \cdot \mathbb{E}[N])}{\mathbb{E}[x \ln y]}$$

which is proportional to $\alpha \cdot \ln(c\beta)$ (where $c = \mu\mathbb{E}[N]$ is a constant), and hence meets our demand.

Case C. More considerations on stake value

One may consider that stake should play a more important role during the committee election. In this case we can choose $G(x, y) = y \ln x$, and its analysis is similar to that of Case B.

5 Conclusion

We generalized the classical PoW to make it forking-free which leads to a better evaluation of computing power. We then constructed fork-free hybrid consensus based on generalized PoW to address the issues of selfish mining and fair committee election in the original hybrid consensus.

With these, we presented a novel generalization of PoA. We firstly built our consensus protocol based on the framework of hybrid consensus with generalized PoW. Then we presented a flexible way of rotating committee election, i.e., for a candidate with PoW capability w and stake value s, a function $G(w, s)$ can be established to determine the probability that the candidate is elected into the committee. We showed that we should avoid non-"concave" choice of $G(w, s)$ which would lead to heavy network burden. Meanwhile, we gave security analyses of generalized PoA under different strategies of combining PoW and PoS. Taking the advantage of PoS generalized PoA is an improvement of hybrid consensus. Moreover, compared with Bentov et al.'s PoA, generalized PoA further improves the efficiency and provides a more flexible combination of PoW and PoS.

Forking-free hybrid consensus or generalized PoA could be adopted in blockchains requiring an efficient and flexible consensus mechanism. For future work, it will be interesting to consider appropriate privacy solutions for our new proposals.

References

AMN+16. Abraham, I., Malkhi, D., Nayak, K., Ren, L., Spiegelman, A.: Solidus: An incentive-compatible cryptocurrency based on permissionless Byzantine consensus. CoRR vol. abs/1612.02916 (2016)

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)

BLMR14. Bentov, I., Lee, C., Mizrahi, A., Rosenfeld, M.: Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]. SIGMETRICS Performance Evaluation Review vol. 42(3), pp. 34–37 (2014)

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)

CG05. Clementi, F., Gallegati, M.: Pareto's law of income distribution: Evidence for grermany, the united kingdom, and the united states. In: Econophysics of wealth distributions. Milan: Springer-Verlag (2005)

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)

EGSR15. Eyal, I., Gencer, A.E., Sirer, E.G., van Renesse, R.: Bitcoin-NG: A scalable blockchain protocol. CoRR vol. abs/1510.02037 (2015)

ES14. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: Christin, N., Safavi-Naini, R. (eds.) Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers, Lecture Notes in Computer Science, vol. 8437, pp. 436–454. Springer (2014)

JJ99. Jakobsson, M., Juels, A.: Proofs of work and bread pudding protocols. In: Preneel, B. (ed.) Secure Information Networks: Communications and Multimedia Security, IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS '99), September 20-21, 1999, Leuven, Belgium, IFIP Conference Proceedings, vol. 152, pp. 258–272. Kluwer (1999)

Kin13. King, S.: Primecoin: Cryptocurrency with prime number proof-of-work (2013). http://primecoin.io/bin/primecoin-paper.pdf

LSP82. Lamport, L., Shostak, R.E., Pease, M.C.: The Byzantine generals problem. ACM Trans. Program. Lang. Syst. vol. 4(3), pp. 382–401 (1982)

Nak08. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)

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)

PSL80. Pease, M.C., Shostak, R.E., Lamport, L.: Reaching agreement in the presence of faults. J. ACM vol. 27(2), pp. 228–234 (1980)

Qua11. QuantumMechanic: Proof of stake instead of proof of work (2011). https://bitcointalk.org/index.php?topic=27787.0

SBRS16. Sengupta, B., Bag, S., Ruj, S., Sakurai, K.: Retricoin: Bitcoin based on compact proofs of retrievability. In: Proceedings of the 17th International Conference on Distributed Computing and Networking, Singapore, January 4-7, 2016, pp. 14:1–14:10. ACM (2016)

Swa15. Swan, M.: Blockchain thinking : The brain as a decentralized autonomous corporation [commentary]. IEEE Technol. Soc. Mag. vol. 34(4), pp. 41–52 (2015)

TJ11. van Tilborg, H.C.A., Jajodia, S.: Proof of work. In: Encyclopedia of Cryptography and Security, 2nd Ed., p. 984. Springer (2011)

TPS87. Toueg, S., Perry, K.J., Srikanth, T.K.: Fast distributed agreement. SIAM J. Comput. vol. 16(3), pp. 445–457 (1987)

TS16. Tschorsch, F., Scheuermann, B.: Bitcoin and beyond: A technical survey on decentralized digital currencies. IEEE Communications Surveys and Tutorials vol. 18(3), pp. 2084–2123 (2016)

WV16. Wustrow, E., VanderSloot, B.: DDoSCoin: Cryptocurrency with a malicious proof-of-work. In: 10th USENIX Workshop on Offensive Technologies, WOOT 16, Austin, TX, August 8-9, 2016. USENIX Association (2016)

A Generalized PoW under Network Delay

In the following, for simplicity, we consider N candidates sharing the same computing power, i.e., their expectation of timing of finding one nonce solution in generalized PoW is T_s. We assume one of them (say, Tom) suffers from network delays and has to begin the puzzle-solving at time δ, and all other nodes start the puzzle-solving at time $\delta' < \delta$. We use Δ' to denote the ending time of the current round. Firstly, we begin with the following lemmas.

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

Proof.

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

Then we get

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

Lemma 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. Let $d = \delta - \delta'$, 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 the previous lemma and get:

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

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:

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

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

In practice, the number of miners N can be regarded as a great number. For this reason, we can merely consider the case under "sufficiently large N".

Given the lemmas above, we now illustrate how an inequality proves generalized PoW is better than PoW in terms of fairness under network delay. In generalized PoW, the probability that Tom becomes the new committee of the 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 is the number of nonce solutions to be found if starting the puzzle-solving at time δ.

When we stipulate that the first block mined should be the on-chain block (even if possible to realize), the probability of Tom's entering committee next round in an ordinary PoW is:

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

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

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

When $\delta' < \delta < \Delta'$, from Lemma 2, the damage of delay is less in generalized PoW, i.e., $\gamma_2 < \gamma_1$.

B Security Analysis for Case A

Firstly, we introduce the logarithmic normal distribution (for short, log-normal distribution).

Definition 2 (Logarithmic Normal Distribution). *When distribution X follows logarithmic normal distribution $LN(\mu, \sigma^2)$, its density function is:*

$$p(x) = \frac{1}{\sqrt{2\pi} x \sigma} \exp\{-\frac{(\ln x - \mu)^2}{2\sigma^2}\}, x \geq 0$$

with the expectation $\mathbb{E}[X] = \exp\{\mu + \sigma^2/2\}$.

$$\mathbb{E}[y] = \exp\{\mu_2 + \tfrac{\sigma_2^2}{2}\} = \mu$$
$$\mathbb{E}[x|y] = \exp\{\mu_1(y) + \tfrac{\sigma_1^2}{2}\} = y$$
$$\mathbb{E}[x] = \mathbb{E}[\mathbb{E}[x|y]] = \mu$$
$$\mu_1(y) = \ln y - \tfrac{\sigma_1^2}{2}$$
$$\mu_2 = \ln \mu - \tfrac{\sigma_2^2}{2}$$

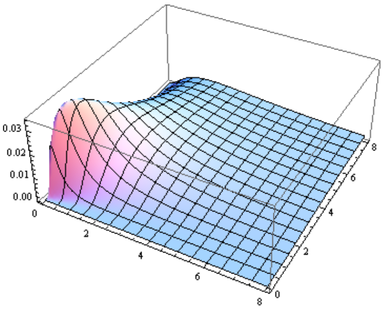

Fig. 5. Log-Normal Distribution

In economics, evidence has shown that the income of over 97% of the population is distributed log-normally [CG05]. In our scenario, we use it to describe the distribution of normalized proof-of-work (x) and proof-of-stake (y).

In reality, holders of more stake are more likely to have greater computing power. In the following discussion, we will consider that the distribution of y follows $y \sim LN(\mu_2, \sigma_2^2)$, and that the distribution of x conditioned on y follows $x \sim LN(\mu_1(y), \sigma_1^2)$, where $\mu_1(y) = \ln y - \frac{\sigma_1^2}{2}$, x is normalized PoW capability, and y is the normalized PoS value (now we have made $\mathbb{E}[x] = \mathbb{E}[y] = \mu$). Here we give a detailed analysis on Case A under assumptions above.

In Sec. 4, we have illustrated that under Case A:

$$\mathbf{Adv}_{\alpha,\beta} = \frac{G(\alpha \cdot \mathbb{E}[\sum_{i=1}^{N} x_i], \beta \cdot \mathbb{E}[\sum_{i=1}^{N} y_i])}{\mathbb{E}[\sum_{i=1}^{N} G(x_i, y_i)]} = \frac{\sqrt{\alpha\mathbb{E}[x] \cdot \beta\mathbb{E}[y]}}{\mathbb{E}[\sqrt{xy}]} = \frac{\sqrt{\alpha\beta} \cdot \mu}{\mathbb{E}[\sqrt{xy}]},$$

We first evaluate $\mathbb{E}[\sqrt{xy}]$.

$$\begin{aligned}
\mathbb{E}[\sqrt{xy}] &= \iint_{D=\mathbb{R}^+ \times \mathbb{R}^+} \sqrt{xy} \cdot p_x(x|y) \cdot p_y(y) \cdot \mathrm{d}\sigma \\
&= \iint_D \sqrt{xy} \cdot \frac{1}{\sqrt{2\pi x}\sigma_1} \exp\{-\frac{(\ln x - \mu_1(y))^2}{2\sigma_1^2}\} \cdot \frac{1}{\sqrt{2\pi y}\sigma_2} \exp\{-\frac{(\ln y - \mu_2)^2}{2\sigma_2^2}\} \cdot \mathrm{d}\sigma \\
&= \frac{1}{2\pi} \int_0^{+\infty} \int_0^{+\infty} \frac{1}{\sqrt{x}\sigma_1} \exp\{-\frac{(\ln x - \ln y + \frac{\sigma_1^2}{2})^2}{2\sigma_1^2}\} \cdot \frac{1}{\sqrt{y}\sigma_2} \exp\{-\frac{(\ln y - \ln \mu + \frac{\sigma_2^2}{2})^2}{2\sigma_2^2}\} \cdot \mathrm{d}x\mathrm{d}y \\
&= \frac{1}{2\pi} \int_0^{+\infty} \frac{1}{\sqrt{y}\sigma_2} \exp\{-\frac{(\ln y - \ln \mu + \frac{\sigma_2^2}{2})^2}{2\sigma_2^2}\} \left[\int_0^{+\infty} \frac{1}{\sqrt{x}\sigma_1} \exp\{-\frac{(\ln x - \ln y + \frac{\sigma_1^2}{2})^2}{2\sigma_1^2}\} \cdot \mathrm{d}x \right] \mathrm{d}y.
\end{aligned}$$

For the last term,

$$\begin{aligned}
&\int_0^{+\infty} \frac{1}{\sqrt{x}\sigma_1} \exp\{-\frac{(\ln x - \ln y + \frac{\sigma_1^2}{2})^2}{2\sigma_1^2}\} \cdot \mathrm{d}x = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{e^t}\sigma_1} \exp\{-\frac{(t - \ln y + \frac{\sigma_1^2}{2})^2}{2\sigma_1^2}\} \cdot e^t \mathrm{d}t \\
=&\int_{-\infty}^{+\infty} \frac{1}{\sigma_1} \exp\{-\frac{(t - \ln y + \frac{\sigma_1^2}{2})^2 - t\sigma_1^2}{2\sigma_1^2}\} \cdot \mathrm{d}t = \int_{-\infty}^{+\infty} \frac{1}{\sigma_1} \exp\{-\frac{t^2 - 2t\ln y + (\ln y - \frac{1}{2}\sigma_1^2)^2}{2\sigma_1^2}\} \cdot \mathrm{d}t \\
=&\int_{-\infty}^{+\infty} \frac{1}{\sigma_1} \exp\{-\frac{(t - \ln y)^2 + (-\sigma_1^2 \ln y + \frac{1}{4}\sigma_1^4)}{2\sigma_1^2}\} \cdot \mathrm{d}t \\
=&\int_{-\infty}^{+\infty} \frac{\sqrt{2\pi}}{\sqrt{2\pi}\sigma_1} \exp\{-\frac{(t - \ln y)^2}{2\sigma_1^2}\} \cdot \exp\{-\frac{(-\sigma_1^2 \ln y + \frac{1}{4}\sigma_1^4)}{2\sigma_1^2}\} \cdot \mathrm{d}t \\
=&\sqrt{2\pi} \exp\{-\frac{(-\sigma_1^2 \ln y + \frac{1}{4}\sigma_1^4)}{2\sigma_1^2}\} \cdot \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\{-\frac{(t - \ln y)^2}{2\sigma_1^2}\} \cdot \mathrm{d}t \\
=&\sqrt{2\pi} \exp\{-\frac{-\ln y + \frac{1}{4}\sigma_1^2}{2}\}.
\end{aligned}$$

Putting it back to our derivation of $\mathbb{E}[\sqrt{xy}]$,

$$\begin{aligned}
\mathbb{E}[\sqrt{xy}] &= \frac{1}{2\pi} \int_0^{+\infty} \frac{1}{\sqrt{y}\sigma_2} \exp\{-\frac{(\ln y - \ln \mu + \frac{\sigma_2^2}{2})^2}{2\sigma_2^2}\} \left[\int_0^{+\infty} \frac{1}{\sqrt{x}\sigma_1} \exp\{-\frac{(\ln x - \ln y + \frac{\sigma_1^2}{2})^2}{2\sigma_1^2}\} \cdot \mathrm{d}x\right] \mathrm{d}y \\
&= \frac{1}{2\pi} \int_0^{+\infty} \frac{1}{\sqrt{y}\sigma_2} \exp\{-\frac{(\ln y - \ln \mu + \frac{\sigma_2^2}{2})^2}{2\sigma_2^2}\} \sqrt{2\pi} \exp\{-\frac{-\ln y + \frac{1}{4}\sigma_1^2}{2}\} \mathrm{d}y \\
&= \int_0^{+\infty} \frac{1}{\sqrt{2\pi}\sqrt{y}\sigma_2} \exp\{-\frac{(\ln y - \ln \mu + \frac{\sigma_2^2}{2})^2}{2\sigma_2^2}\} \exp\{-\frac{-\sigma_2^2 \ln y + \frac{1}{4}\sigma_1^2\sigma_2^2}{2\sigma_2^2}\} \mathrm{d}y \\
&= \int_{-\infty}^{+\infty} \frac{e^{t/2}}{\sqrt{2\pi}\sigma_2} \exp\{-\frac{(t - \ln \mu + \frac{\sigma_2^2}{2})^2}{2\sigma_2^2}\} \exp\{-\frac{-\sigma_2^2 t + \frac{1}{4}\sigma_1^2\sigma_2^2}{2\sigma_2^2}\} \mathrm{d}t \\
&= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\{-\frac{-t\sigma_2^2}{2\sigma_2^2}\} \exp\{-\frac{(t - \ln \mu + \frac{\sigma_2^2}{2})^2}{2\sigma_2^2}\} \exp\{-\frac{-\sigma_2^2 t + \frac{1}{4}\sigma_1^2\sigma_2^2}{2\sigma_2^2}\} \mathrm{d}t \\
&= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\{-\frac{(t - \ln \mu + \frac{1}{2}\sigma_2^2)^2 - 2t\sigma_2^2 + \frac{1}{4}\sigma_1^2\sigma_2^2}{2\sigma_2^2}\} \cdot \mathrm{d}t \\
&= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\{-\frac{[t - (\ln \mu + \frac{1}{2}\sigma_2^2)]^2 - 2\sigma_2^2 \ln \mu + \frac{1}{4}\sigma_1^2\sigma_2^2}{2\sigma_2^2}\} \cdot \mathrm{d}t \\
&= \exp\{\ln \mu - \frac{1}{8}\sigma_1^2\} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\{-\frac{[t - (\ln \mu + \frac{1}{2}\sigma_2^2)]^2}{2\sigma_2^2}\} \cdot \mathrm{d}t \\
&= \mu \cdot e^{-\sigma_1^2/8}.
\end{aligned}$$

So

$$\mathbf{Adv}_{\alpha,\beta} = \frac{\sqrt{\alpha\beta} \cdot \mu}{\mathbb{E}[\sqrt{xy}]} = \sqrt{\alpha\beta} \cdot e^{\sigma_1^2/8}.$$

Suppose that σ_1 is small enough, plotting $(\alpha, \beta, \mathbf{Adv}_{\alpha,\beta})$'s into a graphic, we have Fig. 6. When $\sigma_1 = 1, \alpha = \beta = 29\%$, $\mathbf{Adv}_{\alpha,\beta} < \frac{1}{3}$ holds and the security of PBFT can be guaranteed.

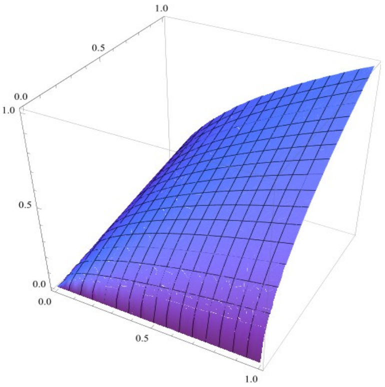

Fig. 6. Probability of the adversary victory in Case A: the perpendicular dimension is probability, two horizontal dimensions are α and β

C Supplementary Material: A Formal Analysis for "Bouncing PoW"

As our initial attempt in improving the traditional PoW, we formulate the following construction of bouncing PoW. In this construction, we are free from setting a predetermined difficulty for the hash puzzle. During each round, each candidate tries to find a nonce nc_i so that the hash result h_i can be as small as possible. And the final w_i is simply set to $1/h_i$. (We may adjust this hash function as the original cryptographic hash function incremented by 1, so that $h_i = 0$ never happens.) We defer to the appendix for a formal justification of our choice of $1/h_i$ (instead of, say $1/h_i^2$ or $1/(h_i \ln h_i)$). For the coefficient of variance,

$$\begin{aligned}
\mathbb{E}[w_i] =& \sum_{j=1}^{M} T \cdot (1 - \frac{j}{M})^{T-1} \cdot \frac{1}{M} \cdot \frac{1}{j} \approx \frac{T}{M} \sum_{j=1}^{M} \frac{1}{j} e^{-\frac{T}{M} j}, \\
\mathbb{E}[w_i^2] =& \sum_{j=1}^{M} T \cdot (1 - \frac{j}{M})^{T-1} \cdot \frac{1}{M} \cdot \frac{1}{j^2} \approx \frac{T}{M} \sum_{j=1}^{M} \frac{1}{j^2} e^{-\frac{T}{M} j}, \\
C_v[w_i] =& \frac{\sqrt{\mathrm{Var}[w_i]}}{\mathbb{E}[w_i]} = \frac{\sqrt{\mathbb{E}[w_i^2] - \mathbb{E}^2[w_i]}}{\mathbb{E}[w_i]}.
\end{aligned}$$

For instance, when $M/T > 10^3$, $M = 2^{60}$, with some scientific computing techniques, we can know $C_v[w_i] > 6$. This result is better than that of the traditional PoW, while still not very satisfactory.

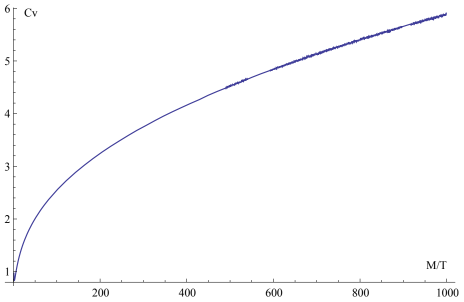

Fig. 7. Coefficient of variance for bouncing PoW as M/T grows ($M = 2^{60}$)

We now formally prove that a "score" assigned based on $1/h_i$ (where h_i is the least hash value candidate i found by solving the hash puzzle) can be a dependable inquiry into candidate's computing power. To this end, we will show that the expectation $\mathbb{E}[h_i]$ is proportional to the inverse of computing power ($1/T$), suppose that candidate i tries to solve the hash puzzle for T times each round. Let the length of the output range of this hash be M, we have

$$\mathbb{E}[h_i] = \sum_{j=1}^{M} T \cdot (1 - \frac{j}{M})^{T-1} \cdot \frac{1}{M} \cdot j \approx \frac{T}{M} \sum_{j=1}^{M} j \cdot e^{-\frac{T}{M} j}.$$

After some derivations, we obtain

$$\mathbb{E}[h_i] \approx -\frac{\exp\{\frac{T}{M} - \frac{(1+M)T}{M}\}(e^{\frac{T}{M}} - e^{\frac{(1+M)T}{M}} - M + e^{\frac{T}{M}} M)T}{(-1 + e^{\frac{T}{M}})^2 M}$$

By omitting some negligible components, we can see the expectation of h_i is proportional to $1/T$:

$$\mathbb{E}[h_i] \approx \frac{T/M}{(e^{T/M} - 1)^2} = \frac{T/M}{(T/M + o(T/M))^2} \approx \frac{M}{T} \propto \frac{1}{T}.$$

This illustrates the inverse relation between h_i and T. Thus the "score" should be set to $w_i = \frac{1}{h_i}$.