Accepted Manuscript

Fork-free hybrid consensus with flexible Proof-of-Activity

Zhiqiang Liu, Shuyang Tang, Sherman S.M. Chow, Zhen Liu, Yu Long

PII: S0167-739X(18)32625-6
DOI: https://doi.org/10.1016/j.future.2019.02.059
Reference: FUTURE 4812

To appear in: *Future Generation Computer Systems*

Received date : 26 October 2018
Revised date : 13 January 2019
Accepted date : 22 February 2019

Please cite this article as: Z. Liu, S. Tang, S.S.M. Chow et al., Fork-free hybrid consensus with flexible Proof-of-Activity, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.02.059

Fork-Free Hybrid Consensus with Flexible Proof-of-Activity

Zhiqiang Liu[1], Shuyang Tang[1(✉)],
Sherman S.M. Chow[2], Zhen Liu[1], and Yu Long[1]

[1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*
{ilu_zq, htftsy, liuzhen, longyu}@sjtu.edu.cn

[2]*Department of Information Engineering, Chinese University of Hong Kong, Shatin, Hong Kong*
sherman@ie.cuhk.edu.hk

Abstract

Bitcoin and its underlying blockchain mechanism have been attracting much attention. One of their core innovations, Proof-of-Work (PoW), is notoriously inefficient which potentially motivates a centralization of hash power, defeating the original goal 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 hybrids of two mechanisms. On the other hand, the hybrid consensus protocol of Pass and Shi (DISC 2017) aims to improve the efficiency by dynamically maintaining a rotating committee. Yet, there are unsatisfactory issues including chain forks and fair committee election.

In this paper, we firstly devise a generalized variant of PoW. After that, we leverage our generalized PoW to construct a fork-free hybrid consensus protocol. We further combine our fork-free hybrid consensus mechanism with

PoS for a flexible version of PoA with tunable parameters between PoW and PoS. Compared with Bentov et al.'s PoA, our "flexible PoA" improves the efficiency, leading to a more applicable consensus protocol.

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

---

1. Introduction

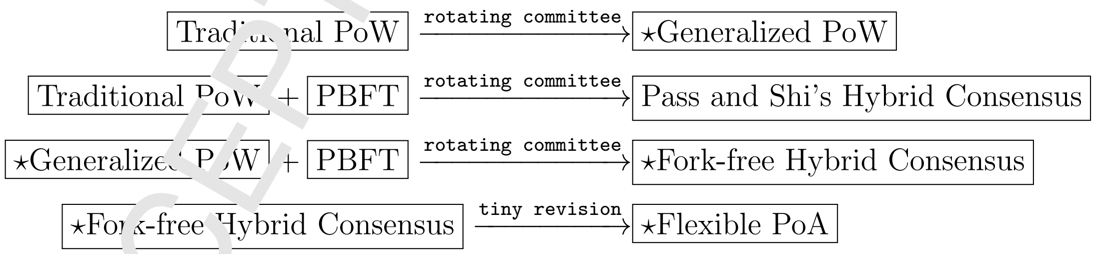
Blockchain, or "Nakamoto chain" (for differentiating it from later proposals), has been attracting much interest (e.g., see Bonneau et al. [2015], Swan [2015], Tschorsch and Scheuermann [2016]) since it first appears as an implicit consensus mechanism used by Bitcoin (Nakamoto [2008]) and subsequent decentralized cryptocurrencies (e.g., Abraham et al. [2016], Sengupta et al. [2016], Wustrow and VanderSloot [2016]). Blockchain keeps a growing distributed ledger of blocks, each of which includes an ordered list of transactions. Blockchain is built upon the methodology of *Proof-of-Work* (PoW) (e.g., see van Tilborg and Jajodia [2011]), which requires the creator of a new block to solve a hash puzzle regarding the hash of the previous block, an ordered list of transactions, as well as other necessary information. Solving a hash puzzle regarding some content w is to find a solution x so that $H(x||w)$ falls into a target range. Thereby, any newly generated block is created by an honest node with high probability, as most computing power (called "hash rate", or "hash power") solving this puzzle is at hands of honest nodes. After a solution is obtained, the lucky solver (also called miner, for the possibility of gaining some bitcoins after completing this process) can then propose a block containing a list of transactions to the peer-to-peer bitcoin network, and the distributed ledger of blocks grows. PoW ensures that tampering the records on the blockchains requires investing a lot of computing power. We refer this as "traditional PoW", or just "PoW" when no ambiguity exists.

When multiple new blocks are generated "simultaneously", the disagreement manifests in the form of a chain fork (or simply a *fork*) having more than one branch. The fork may be a result of coincidence or tampering attempt from malicious nodes. To confirm which branch is valid, the rule used by the bitcoin system is to pick the first forked branch that is followed by a certain number of blocks. and discard any other branches. As such, honest nodes should only work on the longest valid chain. Resolving the fork tackles the misbehavior of (malicious) miners, i.e., clearing any disagreement and making all nodes concede to "the miner of the next block". Yet, users have to wait long to make sure one block will not be nullified by other forks. Also, fork leads to issues like selfish mining (Eyal and Sirer [2014]), which undermines both fairness and security. A fork-free blockchain consensus is thus desired.

Serving as a core part of the consensus protocol underlying bitcoin, PoW shows several potential merits such as openness to any participant and good robustness. The puzzle should be hard enough so that expectedly only one block can be solved in a certain period of time, which is ten minutes in bitcoin. PoW-based protocols thus often confirm the validity of a newly added block at an unsatisfactory speed. Since an individual may take years to find a puzzle solution, mining pools emerges which bring us back to a more centralized setting.

Two major approaches are considered for addressing the above issues. The first approach is to replace PoW with *Proof-of-Stake* (PoS) (Quantum-Mechanic et al. [2011], Bentov et al. [2016], Gilad et al. [2017]), which moves the decision basis from computing power to possession of stake in the system (e.g., in the form of cryptocurrency). With PoS, specific risk of having a few mining farms dominating PoW is mitigated, and the fork-free property can be achieved. Yet, PoS still faces another kind of centralization risk (from large stakeholders). Another approach is to adjust the protocol of PoW, such as *Fruitchain* (Pass and Shi [2017a]) which aims to reduce the variance in mining revenue without a centralized mining pool. Other works are done to provide an instant transaction confirmation (Pass and Shi [2017b], Abraham et al. [2016]). However, to our knowledge, no PoW-based protocol simultaneously achieves the fork-free property, significant improvements to the variance, and instant transaction confirmations. This motivates our work.

Figure 1: Conceptual Design of Primitives in This Paper (our innovations are marked with $\star$)



We aim to achieve a fork-free property and a smaller variance of miners'

Table 1: Comparisons between Consensus Schemes

Consensus Scheme	Efficiency	Fork-free Property	PoW	PoS	Incentive of Presence	Flexible Hybrid
Classical PoW (van Tilborg and Jajodia [2011])			✓		✓	
Ideal PoS (QuantumMechanic et al. [2011])	✓	✓		✓		
Hybrid Consensus (existing) (Pass and Shi [2017b])	✓		✓		✓	
Proof-of-Activity (Bentov et al. [2014])			✓	✓	✓	
Fork-free Hybrid Consensus	✓	✓	✓		✓	
Flexible PoA	✓	✓	✓	✓	✓	✓

revenues, thereby we change the principle of blockchain mining so that multiple puzzle solutions can be found each round. For the first time, blockchain-based consensus protocol accepts multiple solutions, and we name it "the generalized PoW". All of these solutions are submitted to a committee directly without causing any fork by the means of a practical Byzantine fault tolerance (PBFT) from the distributed system literature. Moreover, all of them are recorded, so that the history of records is still hard to forge. Based on the idea of hybrid consensus proposed by Pass and Shi (Pass and Shi [2017b]), and the generalized PoW, we construct a scheme which we call the fork-free hybrid consensus. Note that the protocol of Pass and Shi elects a committee by the blockchain to verify transactions, who are miners of certain blocks. In contrast, our fork-free hybrid consensus protocol lets the committee (instead of block proposers) decide the record for the current round (including transactions, accepted puzzle solutions) and future committee members once for all without any ambiguity.

We can further allow different rules of committee election. Specifically, we establish a function to assign a weight to each candidate according to its PoW power and its PoS capability, and the election is based on such a weight. We thus propose a *flexible PoA* protocol. This takes a step further from the notion of *Proof-of-Activity* (PoA) proposed by Bentov et al. (Bentov et al. [2014]) which aims to inherit the advantages of both PoW and PoS by determining the miner of a new block by taking into account both its hash power as well as its stake. Basing on the fork-free hybrid consensus, our flexible PoA is also fork-free. Tab. 1 compares between our constructions and other few consensus schemes. We show the roadmap of our constructions in Fig. 1.

Technical Novelty of Our Work

1. **The first fork-free PoW-based blockchain in the permissionless environment.** In bitcoin, the integrity of transactions in a block is guaranteed by fork resolutions (e.g., blocks including double-spending transactions are resolved), since any malicious branch can be out-raced by an honest one. We employed the paradigm of hybrid consensus which leverages the security of practical Byzantine fault tolerance (PBFT) to get rid of fork resolution while ensuring transaction integrity. To the best of our knowledge, achieving fork-free property in this way is not yet identified by the literature including the work of Pass and Shi.
2. **Reducing variance without centralized mining pools.** Traditional PoW crucially relies on accepting a single hash puzzle to ensure that existing records cannot be tampered with. Our proposed functionality of generalized PoW accepts multiple solutions for the same puzzle in each round, this reduces the mining-revenues variance. This func-

tionality is hard to realize in bitcoin since its setting provides nothing to “operate” on different solutions. But our fork-free hybrid consensus achieves this functionality by leveraging a rotating committee.

3. **Flexible hybrid of PoW and PoS.** We construct a flexible PoA by having a committee perform the election based on a hybrid weight regarding the participants’ PoW power w and the PoS capability s. The relationship between the hybrid weight (w and s) can be flexibly determined according to different scenarios. To our knowledge, such a flexibility is never considered in previous works.

2. Notations and System Model

2.1. Notations

The set $\{1, 2, \ldots, N\}$ is denoted by $[N]$. $x||y$ denotes the concatenation of x and y. $A := B$ assigns B to the variable A, $A \xleftarrow{\$} B$ selects an element of B uniformly at random (if B is a set) or according to B (if B is a distribution). Table 2 lists more notations. A *node* is either a candidate of the committee in the next round or a current committee member.

2.2. Security Model

1. **Network.** We follow the security and network assumptions of Pass and Shi’s hybrid consensus (Pass and Shi [2017b]). We consider the network as partially synchronous, where an adversary may deliver messages out of order, but all messages can be delivered in time Δ. We also assume that all participants have access to the public blockchain, connected by insecure channels (where man-in-the-middle attacks are possible).

2. **Honesty Rate.** We assume a peer-to-peer network without trust on any specific peer, while over α fraction of the hash power and over β of stake are at hands of honest participants.

3. **Other Assumptions.** We assume the collision-resistance of cryptographic hash functions. We also assume PBFT is executed ideally as long as over 2/3 participants are honest with a sufficiently high probability.

2.3. Security Features and Performance Requirements

1. **Fork-Free.** To throughly eliminate the selfish mining and speed up transactions confirmation, we require a novel consensus scheme without chain forks.

2. **Hard-to-Forge (Hard-to-Tamper).** Any adversary with less than half total hash power should have no capability of maintaining a forged chain of valid blocks.

3. **High Chain Quality.** In a fork-free consensus scheme, faulty blocks will stay on the chain instead of being eliminated by other forks, so we require that the fraction of honest blocks (that is, blocks generated by committees of an honest rate over 2/3) should be sufficiently high (in the cryptocurrency literature, such a fraction is referred to "chain quality"). Specifically, we require a $(1-\mathsf{negl}(\lambda))$-chain quality for some negligible function $\mathsf{negl}(\cdot)$ where λ is the security parameter.

4. **Security Against Mildly Agile Corruption.** In hybrid consensus, the adversary is allowed to perform mildly agile corruptions, i.e., they can choose nodes to corrupt according to the configuration of the envi-

ronment. τ-agility means an adversary has to wait for time τ to corrupt an honest node.

5. **Low Communication Complexity.** Communication complexity refers to the number of all interactions required, which includes delivery of blocks from proposers to all network nodes and all interactions among consensus participants (either for the consensus or leader elections). The lower the complexity the better. Yet, a certain degree of complexity such as number of rounds can be inherently required for a secure protocol.

3. Technical Preliminaries

3.1. Practical Byzantine Fault Tolerance

Practical Byzantine fault tolerance (PBFT) algorithm (Castro and Liskov [1999]) (among many other BFT protocols, see Pease et al. [1980], Lamport et al. [1982], Toueg et al. [1987]) provides a high performance Byzantine state machine replication for tolerating certain failures in Byzantine general problem. It has been widely adopted for maintaining distributed ledgers. In this work, we treat PBFT as a blackbox among n participants, by which a consensus on a linearly ordered log can be attained at the communication cost of $O(n^2)$ provided a 2/3 honest rate of the committee. This is a permissioned protocol, while applicable to a permissionless environment with a delicate hybrid protocol with a blockchain (like Pass and Shi [2017b] and ours).

3.2. Hybrid Consensus

A hybrid of blockchain and a permissioned protocol can improve the performance of blockchain (see Kokoris-Kogias et al. [2016], Decker et al.

[2016], Pass and Shi [2017b]). The newest result is the *Hybrid Consensus* of Pass and Shi, which combines a Byzantine fault-tolerance protocol in the permissioned setting (where participants cannot leave or join during protocol executions) with a blockchain in the permissionless setting (where participants can dynamically leave or join).

The blockchain no longer directly validates transactions, but is the basis of the election of a dynamically-determined rotating committee (in short, committee). Specifically, committee members of each round correspond to miners of a fixed sequence of confirmed on-chain blocks. This committee serves as the leader of transaction confirmations and all transactions are verified by the committee via a PBFT among committee members. This inherits the efficiency advantage of PBFT and speeds up transaction confirmations significantly in a permissionless environment.

4. Generalized Proof-of-Work and Fork-free Hybrid Consensus

We propose the functionality of our generalized PoW, show how traditional PoW fits with that, and argue the merits of our generalized notion. Afterwards, the fork-free hybrid consensus is demonstrated to realize the generalized PoW.

4.1. Generalized Proof-of-Work

We propose the ideal functionality of our generalized proof-of-work, an alternative leader election that simultaneously elects csize leaders among candidates. To do this, we lower the difficulty of the mining puzzle so that multiple solutions each round can be found. These nonce solutions are collected by the functionality and csize of them are randomly selected in which the

Functionality $\bar{\mathcal{G}}_{\mathrm{GPoW}}$

Shared Functionality $\bar{\mathcal{G}}_{\mathrm{GPoW}}$ interacts with all parties (candidates) $P_1, P_2, \ldots, P_N$ (the first $k \xleftarrow{\$} [\lfloor N/3 \rfloor]$ of them are controlled by the adversary), the environment $\mathcal{Z}$, the adversary $\mathcal{A}$, as well as a publicly shared global clock functionality $\bar{\mathcal{G}}_{\mathrm{CLOCK}}$.

This functionality is parameterized by the number of candidates N (this is a variant in the permissionless setting, but we take this notation for the simplicity of description), the expected time length of each round t', the number of adversary controlled parties k, the cryptographic hash function $H(\cdot)$, and a target range target within the range of $H(\cdot)$.

– **Puzzle Issuance**

• Obtain puzzle m from the environment $\mathcal{Z}$, issue m to the adversary $\mathcal{A}$ and honest candidates $P_{k+1}, P_{k+2}, \ldots, P_N$.

• Query the global time clock $\bar{\mathcal{G}}_{\mathrm{CLOCK}}$ to attain the time t_0.

– **Nonce Collection**

• Keep an array of $\{(\mathsf{ID}_u, \mathsf{nc}_{u,j})\}$ ($u \in \{P_1, P_2, \ldots, P_N\}, j \in \mathbb{N}^+$), where j denotes the order of nonce solution found by one participant (starting from 1) since one may find more than one solutions. Let $\boldsymbol{W}$ be this array, initially set as $\boldsymbol{W} \leftarrow \emptyset$.

• Set variables $\ell_1, \ell_2, \ldots, \ell_N$ as zero.

• Interact with participants (the adversary $\mathcal{A}$ and $P_{k+1}, P_{k+2}, \ldots, P_N$) to fetch possible nonce solutions. For each received nonce solution nc from P_j, if $H(m, \mathsf{nc}) \in \mathsf{target}$, set $\ell_j \leftarrow \ell_j + 1$, append item $(\mathsf{ID}_{P_j}, \mathsf{nc})$ to $\boldsymbol{W}$.

• Query the global time clock $\bar{\mathcal{G}}_{\mathrm{CLOCK}}$ for time t, go back to the previous step if $t < t_0 + t'$.

– **Member Release**

• Generate csize random numbers $\mathsf{rand}_1, \mathsf{rand}_2, \ldots, \mathsf{rand}_{\mathsf{csize}} \in \left[\sum_{i=1}^{N} \ell_i\right]$.

• Find the $\mathsf{rand}_i^{\mathrm{th}}$ items in $\boldsymbol{W}$ for each $i \in [\mathsf{csize}]$, which are denoted by $(\mathsf{ID}_{\mathsf{CM}_1}, \mathsf{nc}_{\mathsf{CM}_1}), (\mathsf{ID}_{\mathsf{CM}_2}, \mathsf{nc}_{\mathsf{CM}_2}), \ldots, (\mathsf{ID}_{\mathsf{CM}_{\mathsf{csize}}}, \mathsf{nc}_{\mathsf{CM}_{\mathsf{csize}}})$.

• Release the list $(\mathsf{CM}_1, \mathsf{CM}_2, \ldots, \mathsf{CM}_{\mathsf{csize}})$ to all parties. The new committee is formed to substitute the existing one.

Figure 2: The Generalized PoW Functionality

solution providers are determined as the leaders. The protocol is fair as the chance of being elected is proportional to its hash power for each participant.

Specifically, in each round, each candidate finds some nonce solutions and submit them to the functionality $\bar{\mathcal{G}}_{\text{GPoW}}$. These nonce solutions are received and arranged by $\bar{\mathcal{G}}_{\text{GPoW}}$ into an array W. Afterwards, csize random numbers $(\mathsf{rand}_1, \mathsf{rand}_2, \dots, \mathsf{rand}_{\mathsf{csize}})$ are generated within $\bar{\mathcal{G}}_{\text{GPoW}}$. Finally, the identities of next round's committee members are given by the rand_i-th's items of W (for $i \in [\mathsf{csize}]$). Fig. 2 shows the formal description of this functionality.

In this way, the more hash puzzle solutions are found, a greater chance (proportional to the number of solutions found) of being elected. Obviously, the expected number of nonces found is proportional to the hash power of each participant. Hence the chance of being elected is still proportional to candidates' PoW ability like traditional PoW.

Roughly, traditional PoW is a special case of the generalized PoW where the second solution is forbidden and $\mathsf{csize} = 1$.

4.2. Computing Power Evaluation of (Generalized) PoW

While generalized PoW facilitates the simultaneous election of multiple leaders, it also guarantees a better "evaluation" of candidates' hash power. In our latter constructions of the fork-free hybrid consensus and the flexible PoA (in Sec. 5), we hope to assign a "score" w_i to each candidate, to evaluate the hash power (hash rate) of candidates. To form an accurate evaluation, w_i's should be proportional to candidates' real hash power expectedly, with less variance.

We now formally compare the generalized PoW with the traditional one concerning the accuracy of the hash power evaluation. In fact, the expected

w_i's under two protocols can be regarded as proportional to candidates' hash power, we thus make comparisons on their coefficients of variance and finally determine that our new construction is more satisfiable.

To simplify the formalization, 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. For the generalized PoW, let $\gamma_T := \frac{M_0}{M}T$ be the expected number of valid hash puzzle solutions found by this candidate in one round. Moreover, for the traditional PoW, we denote the probability of having one valid hash puzzle solution found by p_T.

4.2.1. Traditional PoW

Traditional PoW can be viewed as the following game: we set the puzzle difficulty very high and ask each candidate i to try 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 $T \cdot M_0 \ll M$ holds for each individual. The expectation of w_i is thus proportional to the hash power T, by definition: $\mathbb{E}[w_i] = p_T$.

In bitcoin, the chance for a participant to find more than one solution is negligible, we regard that w_i satisfies a binomial distribution, so $\mathrm{Var}[w_i] \approx \mathbb{E}[w_i](1 - \mathbb{E}[w_i]) = p_T(1 - p_T)$.

And the coefficient of variance is

$$C_v[w_i] = \frac{\sqrt{\mathrm{Var}[w_i]}}{\mathbb{E}[w_i]} \approx \sqrt{\frac{1 - p_T}{p_T}} \approx \sqrt{\frac{1}{p_T}} > 1.$$

This holds since each candidate's possibility of find one hash puzzle solution is small (i.e., $p_T \ll 1$). We can see that the coefficient of variance is significant in the traditional PoW.

4.2.2. Generalized PoW

Generalized PoW lowers the difficulty so that a candidate with considerable hash power may find more than one solutions to a hash puzzle. The final value of w_i will be the number of solutions it found. For example, suppose that the difficulty is lowered down to 1% of traditional blockchain's, then 100 solutions can be found each round in expectation. A powerful participant holding 10% overall hash power may find many solutions to the puzzle, say, 10 solutions, then its w_i is 10. The expected number of solutions one candidate i with T hash power may find is

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

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

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

and so

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

For example, if $\gamma_T = 10$, i.e., 10 valid puzzle solutions are expected to be found by this candidate in one round, $C_v[w_i] \approx \sqrt{1/10}$ is much smaller than the bitcoin case (traditional PoW). In conclusion, the generalized PoW is endowed with a smaller coefficient of variance. Next, we introduce our fork-free hybrid consensus protocol that securely realizes the generalized PoW $\bar{\mathcal{G}}_{\mathrm{GPoW}}$.

4.3. Fork-free Hybrid Consensus

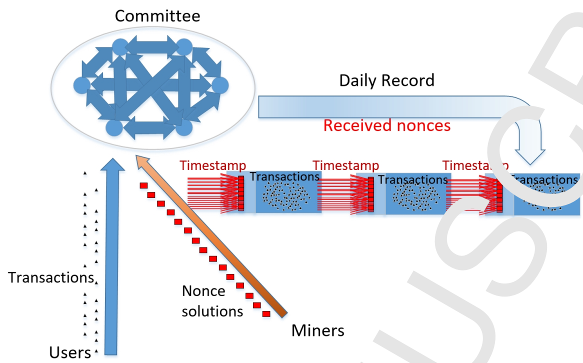


Figure 3: Fork-free hybrid consensus

Similar to the existing hybrid consensus, our fork-free hybrid consensus protocol adopts a committee of size csizewhich is rotated every round. Transactions are verified by this committee via PBFT. Each committee is elected from the previous committee except for the generation of the first csizeblocks (one generator is needed to start the protocol and maintain the first csizeblocks and the first committee). The outline of the routine of each round is shown in Fig. 4. Below we present our fork-free hybrid consensus protocol.

For simplicity, we order all committee members in $1, 2, \ldots,$ csize. Different from the traditional bitcoin blockchain, round record rec_R here includes users' transactions handled by round R's committee, reward transactions for round R's committee (which will be specified later), and all accepted nonces during round R. CM_R is the identity list (i.e., public keys) of committee members of the R-th round.

1. In round R, each candidate, say, u, collects transactions and nonce

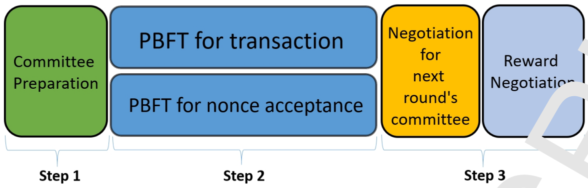


Figure 4: The Round Routine

records of round $R-1$ (signed by over $1/3$ committee members) as rec_{R-1}, member of this committee determined by the previous committee as CM_R. Then it receives committee members' signatures on the previous block header[1]. Next, it recovers previous block $\boldsymbol{B}_{R-1} = \{\mathsf{rec}_{R-1}, H(\mathsf{header}(\boldsymbol{B}_{R-2})), \mathsf{CM}_R\}$, and aborts this procedure if $\mathsf{header}(\boldsymbol{B}_{R-1})$ does not match over $1/3$ of committee members' block header signatures.

2. The committee of round R is assembled according to CM_R. Committee members start an instance of PBFT that reaches consensus on candidates' puzzle solutions and an instance for the consensus on newly received transactions (see Fig. 4).

3. Each candidate u finds as much as possible nonce(s) $\mathsf{nc}_{u,1}, \mathsf{nc}_{u,2}, \ldots, \mathsf{nc}_{u,P_u}$ such that

$$H\left(\mathsf{header}(\boldsymbol{B}_{R-1})\,||\mathsf{ID}_u||\mathsf{nc}_{u,i}\right) \in \mathsf{target}(1 \leq i \leq P_u).$$

[1] The header of a block should at least contain the block height, the hash to the previous block, the hash of the block body and the member list of the next committee.

4. u arranges all nonces found into W_u:

$$W_u = \begin{bmatrix} \mathsf{nc}_{u,1} & \mathsf{ID}_u \\ \mathsf{nc}_{u,2} & \mathsf{ID}_u \\ \vdots & \vdots \\ \mathsf{nc}_{u,P_u} & \mathsf{ID}_u \end{bmatrix}$$

   and submits all items in W_u to the rotating committee before the end of round R.

5. Each honest committee member receives nonces from all candidates, puts all received nonces into an array, and sorts all items in the same order, to get W that is the merged array of all W_u's. At the termination of this round, committee members in $\mathsf{CM}_R = [\mathsf{ID}_1, \mathsf{ID}_2, \ldots, \mathsf{ID}_{\mathsf{csize}}]$ calculate the xor-summation of all received nonces that have passed though the PBFT consensus (denoted by k_R). After that, csize nonces are determined according to k_R among the received nonces. The committee of the next round CM_{R+1} is set to the miners of csizedetermined nonces.

6. After the reward negotiation (in Sec. 4.4), committee members broadcast rec_R with signatures on $\mathsf{header}(\boldsymbol{B}_R)$, where $\boldsymbol{B}_R = \{\mathsf{rec}_R, H(\mathsf{header}(\boldsymbol{B}_{R-1})), \mathsf{CM}_{R+1}\}$. The csize lucky candidates in CM_{R+1} are enrolled into the committee of next round.

Fig. 3 shows an outline of the execution of our protocol.

4.4. Reward Negotiation

To further guarantee honesty and the presence of committee members, we devise a voting-liked mechanism executed by each committee member at the termination of each round.

Specifically, each committee member sets reward for each honest committee member as $S_{reward} = \frac{S_{\mathsf{tx}}+S_{block}}{\mathsf{csize}}$, where S_{tx} is the total transaction fee included in the round record (all honest nodes should have reached the consensus on this amount after PBFT) and S_{block} stands for the predetermined amount of block reward. Afterwards, each committee member (say, member i) generates and signs on the reward transaction tx_j (whose receipt is member j, containing reward amount S_{reward}) for each honest member j. All reward transactions are broadcast to the network along with corresponding signatures.

Similar 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. Thereby each member is rewarded only if over 1/3 committee members approve. For fear of losing rewards from the voting, dishonest behaviors are discouraged.

5. The Flexible Proof-of-Activity

We propose an alternative proof-of-activity to support flexible hybrids of generalized PoW and PoS. Specifically, 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 its PoW capability w and its stake value s. The probability of entering the next committee is determined by such a weight.

We discuss protocols for candidates and committee members separately, detailed illustrations of protocols are shown in Tables B.3 and B.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 the representation, we will use the term "committee member i" or "candidate i" together with "com_i" or "$cand_i$" interchangeably.

In generalized PoW, the PoW capability w and the stake value s are not in the same metric space. For this reason, we normalize w, s before calculating $G(w, s)$, assuming that w's and s's are normalized to x's and y's so that for a node with $w = w'$, $s = s'$, its normalized PoW capability x' and stake value y' are

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

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 $\mathbb{R}^+$.

- *Candidate*

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

1. It packs rec_{R-1}, together with the hash value of block header $\mathsf{header}(\boldsymbol{B}_{R-2})$ (to make records hard-to-tamper) and the list of committee members released by previous committee CM_R, into $\boldsymbol{B}_{R-1}$, the block of round $(R-1)$.
2. It tries to find as much nonces as possible (say, ℓ nonces), which satisfiy $H(\mathsf{header}(\boldsymbol{B}_{R-1}), \mathsf{ID}_i, \mathsf{nc}_j) \in \mathsf{target}$ $\forall 1 \leq j \leq \ell$. Then, it submits the set of nonces $\{\mathsf{nc}_1, \mathsf{nc}_2, \ldots, \mathsf{nc}_\ell\}$ to committee members.
3. It receives rec_R (with corresponding signatures) at the end of the round.

- *Committee member*

  For the committee in round R:

  1. Each node checks the committee list of the current round CM_R, and performs the following procedures if its identity is included in the list. Then, it packs $\boldsymbol{B}_{R-1} = \{\mathsf{rec}_{R-1}, H(\mathsf{header}(\boldsymbol{B}_{R-2})), \mathsf{CM}_R\}$.
  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 $x_j := \frac{\mu}{\mathbb{E}[w]} \cdot w_j$, $y_j := \frac{\mu}{\mathbb{E}[s]} \cdot s_j$ and $L_j := G(x_j, y_j)$ for each candidate j. They then calculate k_R as the xor-summation of all accepted nonces, and decide csize lucky candidates (the committee CM_{R+1} of next round) 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 to ordinary transactions, each reward transaction will be validated if over 1/3 of committee members have signed on it.
  4. It broadcasts rec_R and the signature on $\mathsf{header}(\boldsymbol{B}_R)$, declaring the termination of a round, where $\boldsymbol{B}_R = \{\mathsf{rec}_R, H(\mathsf{header}(\boldsymbol{B}_{R-1})), \mathsf{CM}_{R+1}\}$.

Table B.4 shows the detailed procedures. Strategy analyses of this scheme (and a recommendation on a "concave" $G(\cdot,\cdot)$) are shown in the appendix. The security analysis is shown together with the fork-free hybrid consensus in Sec. 6

6. Security and Performance Analysis

Here, we provide a security analysis for our fork-free hybrid consensus protocol and flexible PoA protocol. The discussion applies on both unless specified otherwise.

6.1. Fork-Freeness

In our hybrid consensus, fork is eliminated since record for each round is generated by the committee once for all without causing any ambiguity. Hence no fork exists in our constructions (both the fork-free hybrid consensus and flexible PoA).

6.2. Hard-to-Forge

One party may try to forge the whole history since it may include only one nonce solution in each block to assembly a new "history" (one party with sufficient hash power may have such capability). However, such an issue can be solved by stipulating that, when two branches of "histories" are found, one with more total nonce solutions inclusions overruns the other one, and the other one is surely forged.

Specifically, since all nonce solutions received by committee members are comprehended into the block via a PBFT among the committee, adjacent blocks are linked by multiple nonce solutions of our generalized PoW, instead of one single solution that is relatively easy to solve. Due to this, any adversary with less than half total hash power is unable to forge a long sequence of forged blocks with competitive total number of comprehended nonce solutions.

6.3. Chain Quality

Theorem 1 (Chain Quality of $(1-e^{-\Omega(\lambda)})$). *Our fork-free hybrid consensus, and the flexible PoA, achieve a $(1-e^{-\Omega(\lambda)})$ chain quality, as long as the fraction of hash power controlled by the adversary (to the fork-free hybrid consensus) or the fraction of total combined weight (to the flexible PoA) is less than 1/3.*

Proof 1. *We let $\alpha = \frac{1}{3} - \epsilon$ be the fraction of hash power (to the fork-free hybrid consensus) or the fraction of total combined weight (to the flexible PoA) controlled by the adversary, Win be the event that the adversary successfully controlled over 1/3 members of next round's committee by one attempt (adversary's controlling over 1/3 committee members is equivalent to generating an adversary block), and indicator X with $\mathbb{E}[X] = \alpha \cdot \mathsf{csize}$ be the number of controlled members in one attempt. By Chernoff bound,*

$$\Pr[X \geq (1+\delta)\alpha \cdot \mathsf{csize}] \leq e^{-[(1+\delta)\ln(1+\delta)-\delta]\alpha\cdot \mathsf{csize}}.$$

Choosing $\delta = \frac{1}{3\alpha} - 1$, we have

$$\Pr[\mathsf{Win}] = \Pr[X \geq \frac{1}{3}\mathsf{csize}] \leq e^{-(\frac{1}{3\alpha}\ln\frac{1}{3\alpha} - \frac{1}{3\alpha}+1)\alpha\cdot\mathsf{csize}}$$
$$= e^{-\Theta(\mathsf{csize})},$$

where $\frac{1}{3\alpha}\ln\frac{1}{3\alpha} - \frac{1}{3\alpha} + 1 > 0$ holds for all $0 < \alpha < 1/3$, hence $\Pr[\mathsf{Win}]$ is negligible in csize. Since $\mathsf{csize} = \Theta(\lambda)$,

$$\Pr[\mathsf{Win}] = e^{-\Omega(\lambda)}.$$

An adversary may choose to disclose its random number or not during the random number negotiation in an attempt of attaining its "favorite" random *number (so that more committee members might be its spawn nodes). In such a case, we assume that the adversary may try to control the committee of the next round by ignoring or adding nonces in the nonce acceptance step for a polynomial number of attempts (denoted by* $\mathsf{attempt}(\lambda)$*). However, the probability for its controlling over* $1/3$ *is still negligible. Specifically, following the formulation above, adversary's probability of succeeding in any attempt is*

$$1-(1-\Pr[\mathsf{Win}])^{\mathit{attempt}(\lambda)} \approx \mathit{attempt}(\lambda)\Pr[\mathsf{Win}] = e^{-\Omega(\lambda)},$$

in that $\Pr[\mathsf{Win}] = e^{-\Omega(\lambda)} \ll \frac{1}{\mathit{attempt}(\lambda)}$.

In the complementary sense, the probability of each block's being honestly generated, and hence the chain quality is $(1-e^{-\Omega(\lambda)})$*, i.e.,* $(1-\mathsf{negl}(\lambda))$ *with a negligible function* $\mathsf{negl}(\cdot)$.

6.4. Looser Assumption Against Mildly Agile Corruptions

In our work, the assumption of τ can be much looser than that required for hybrid consensus, since that once a node is elected into the committee, it will start to work before a long exposure to adversary's target corruption.

6.5. Communication Complexity

All nonce solutions are submitted to the committee like transactions. It is the committee that runs a PBFT (with communication cost $O(\mathsf{csize}^2)$) to reach agreement on nonce acceptance instead of the miners. That is to say, the actual communication cost is $O(\mathsf{csize}^2+n)$ where csize is the size of the rotating committee, and n is total number of nodes within the network. The communication complexity is thus roughly the same as that of Nakamoto consensus, in which the communication cost is $O(n)$.

7. Conclusion

We generalized the classical PoW to make it fork-free, which also leads to a better evaluation of hash 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. The election mechanism for rotating committee in our protocol is flexible in the sense that it takes into the account of both the PoW capability w and stake value s of a candidate. In other words, a function $G(w, s)$ can be established to determine the probability that the candidate is elected into the committee. This flexible PoA is an improvement of hybrid consensus which also inherits the advantage of PoS. Fork-free hybrid consensus or the flexible PoA could be adopted in blockchains requiring an efficient and flexible consensus mechanism.

Appendix A.

In this part, we discuss the strategy of miners under different establishments of $G(\cdot, \cdot)$. Also, during the discussion, we demonstrate the flexibility of our combination by an example that evaluates a miner by the geometric mean of its stake and its hash power which is not yet achieved in the existing PoA. To begin with, we introduce the following definition.

Definition 1. *A 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 $G(\boldsymbol{v}) + G(\boldsymbol{v'}) \leq G(\boldsymbol{v} + \boldsymbol{v'})$.

The strategy of the adversary will be different in two cases to maximize the probability of being elected. In the non-concave case, dishonest nodes tend to divide its hash power and stake to multiple identities it spawned, causing a heavy network burden. While in the concave case, nodes prefer to aggregate their hash power and stake values to form stronger PoW and PoS power so as to maximize the possibility of being elected, which forbids node spawning.

Due to this, we suggest that function $G(x, y)$ should be concave. Since the detailed analyses of the strategy under two cases highly depend on the establishment of $G(\cdot, \cdot)$, two specific establishments are shown for a clear illustration.

A Non-Concave Case

As the case of a non-concave $G(x, y)$, we consider $G(x, y) = \ln(xy)$, and assume that $x, y \geq 1$ holds. 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 next committee is maximized when ℓ reaches some value greater than 1 (i.e., the division of x' and y' exists in the optimal strategy). The total probability of (at least one spawned node's) being elected is

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

After simple derivations, this probability reaches its maximum when ℓ approaches $e^{\frac{\ln(x'y')-2}{2}}$, which is often much greater than 1. Hence, we can see that miners tend to split their total resource into multiple spawned nodes.

A Concave Case

We define the adversary advantage $\mathbf{Adv}_{\alpha,\beta}$ as the upper bound of the possibility of entering the next committee of an adversary:

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

where N is the total number of nodes, α (β) is the fraction of total hash power (stake) held by the adversary, $x_1, x_2, \ldots, x_N$ ($y_1, y_2, \ldots, y_N$) are normalized PoW capabilities (PoS values) of each node. Since it is a upper bound , we consider that all malicious parties are cooperating.

When we consider PoW and PoS evenly (i.e., of same significance), we may set $G(x, y)$ as $\frac{x+y}{2}$, or $\sqrt{\frac{x^2+y^2}{2}}$ (a symmetric binary function). However, we can make the adversary harder to reach a high $G(x, y)$ value with $G(x, y) = \sqrt{xy}$, since it is easier to have a high x value or high y value, but harder to make both x and y great enough (and reach a high $\sqrt{xy}$).

We first prove that this evaluation function $G(x, y) = \sqrt{xy}$ is concave. For any $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^+ \times \mathbb{R}^+$:

$$x_1 y_2 + x_2 y_1 \geq 2\sqrt{x_1 x_2 y_1 y_2},$$

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

$$x_1 y_1 + x_1 y_2 + x_2 y_1 + x_2 y_2 \geq (\sqrt{x_1 y_1})^2 + (\sqrt{x_2 y_2})^2 + 2\sqrt{x_1 x_2 y_1 y_2},$$

$$\sqrt{(x_1 + x_2)(y_1 + y_2)} \geq \sqrt{x_1 y_1} + \sqrt{x_2 y_2},$$

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,

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

Hence the advantage of the adversary will be limited to $\sqrt{\alpha\beta}$ within a multiplicative constant factor. We introduce the logarithmic normal (log-normal) distribution for further calculations.

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

In economics, evidence has shown that the income of over 97% of the population is distributed log-normally (Clementi and Gallegati [2005]). In our scenario, we use it to describe the distribution of normalized proof-of-work (x) and proof-of-stake (y) (see Fig. B.5).

In reality, holders of more stake are more likely to have greater hash power. Hence we 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 this case under assumptions above. Previously we have illustrated that

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

where $\mathbb{E}[\sqrt{xy}] = \iint_{D=\mathbb{R}^+ \times \mathbb{R}^+} \sqrt{xy} \cdot p_x(x|y) \cdot p_y(y) \cdot dx = \mu \cdot e^{-\sigma_1^2/8}$, and so forth

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

When $\sigma_1 = 1, \alpha = \beta = 29\%$, $\mathbf{Adv}_{\alpha,\beta} < \frac{1}{3}$ holds and the security of PBFT is guaranteed.

Appendix B. Discussions

Appendix B.1. Comparison with Hybrid Consensus of Pass and Shi [2017b]

Hybrid consensus merits from a more general framework on top of any admissble underlying blockchain (a classical Nakamoto chain or a fruitchain) and a thorough cryptographic analysis. In comparison, our work merits from several perspectives.

Compared with hybrid consensus on top of the Nakamoto chain, our work is more secure against adaptive target corruptions. To the existing hybrid consensus with Nakamoto chain, there has to be a significant interval (to resolve forks) between "being very likely to enter the next committee" and "entering the next committee" for each miner that proposes a valid block. During this interval, these miners are exposed to adaptive target corruptions which is a considerable treat to the committee honesty (and so forth the safety). In contrast, our protocol requires no such time interval due to the fork-free property.

Compared with hybrid consensus on top of the fruitchain that is built on an utterly novel chain framework, our scheme is easy to implement with an existing dynamic committee by introducing another instance of PBFT. Such a simplicity makes our scheme more practical.

Appendix B.2. Bootstrapping Techniques

To bootstrap the system, we need csize genesis blocks maintained by the first participating party (we assume this party is honest). Differently from the bitcoin, this party have certain hash power to perform the consensus for the first csize rounds.

Appendix B.3. Determination of Commencement and Termination Time

In each round of both fork-free hybrid consensus and flexible PoA, we need to have committee members agree on the same commencement and termination time for each round. PBFT is an ordered procedure during which transactions are processed in the sequential order same to all committee members. With this property, we can stipulate that each round is terminated after the M^{th} transaction is processed, where M is a predetermined parameter.

References

Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. Solidus: An incentive-compatible cryptocurrency based on

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

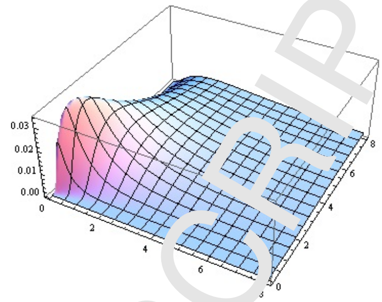


Figure B.5: Log-Normal Distribution

permissionless Byzantine consensus. *CoRR*, abs/1612.02916, 2016. URL http://arxiv.org/abs/1612.02916.

Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]. *SIGMETRICS Performance Evaluation Review*, 42(3):34–37, 2014. doi: 10.1145/2695533.2695545. URL http://doi.acm.org/10.1145/2695533.2695545.

Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies without proof of work. In Jeremy Clark, Sarah Meiklejohn, Peter Y. A. Ryan, Dan S. Wallach, Michael Brenner, and Kurt Rohloff, editors, *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, volume 9604 of *Lecture Notes in Computer Science*, pages 142–157. Springer, 2016. ISBN 978-3-662-53356-7. doi: 10.1007/978-3-662-53357-4_10. URL http://dx.doi.org/10.1007/978-3-662-53357-4_10.

Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. 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*, pages 104–121. IEEE Computer Society, 2015. ISBN 978-1-4673-6949-7. doi: 10.1109/SP.2015.14. URL http://dx.doi.org/10.1109/SP.2015.14.

Miguel Castro and Barbara Liskov. Practical Byzantine fault tolerance. In Margo I. Seltzer and Paul J. Leach, editors, *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 22-25, 1999*, pages 173–186. USENIX Association, 1999. ISBN 1-880446-39-1. doi: 10.1145/296806.296824. URL http://doi.acm.org/10.1145/296806.296824.

F. Clementi and Mauro Gallegati. 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.

Christian Decker, Jochen Seidel, and Roger Wattenhofer. Bitcoin meets strong consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking, Singapore, January 4-7, 2016*, pages 13:1–13:10. ACM, 2016. ISBN 978-1-4503-4032-8. doi: 10.1145/2833312.2833321. URL http://doi.acm.org/10.1145/2833312.2833321.

Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security - 18th International Conference,* *FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, volume 8437 of *Lecture Notes in Computer Science*, pages 436–454. Springer, 2014. ISBN 978-3-662-45471-8. doi: 10.1007/978-3-662-45472-5_28. URL http://dx.doi.org/10.1007/978-3-662-45472-5_28.

Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*, pages 51–68. ACM, 2017. ISBN 978-1-4503-5085-3. doi: 10.1145/3132747.3132757. URL http://doi.acm.org/10.1145/3132747.3132757.

Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing bitcoin security and performance with strong consistency via collective signing. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 279–296. USENIX Association, 2016. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kogias.

Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982. doi: 10.1145/357172.357176. URL http://doi.acm.org/10.1145/357172.357176.

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. URL http://www.bitcoin.org.

Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 315–324. ACM, 2017a. ISBN 978-1-4503-4992-5. doi: 10.1145/3087801.3087809. URL http://doi.acm.org/10.1145/3087801.3087809.

Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. In *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, pages 39:1–39:16, 2017b. doi: 10.4230/LIPIcs.DISC.2017.39. URL https://doi.org/10.4230/LIPIcs.DISC.2017.39.

Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980. doi: 10.1145/322186.322188. URL http://doi.acm.org/10.1145/322186.322188.

QuantumMechanic et al. Proof of stake instead of proof of work. Bitcoin forum, 2011. https://bitcointalk.org/index.php?topic=27787.0.

Binanda Sengupta, Samiran Bag, Sushmita Ruj, and Kouichi Sakurai. 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*, pages 14:1–14:10. ACM, 2016. ISBN 978-1-4503-4032-8. doi: 10.1145/2833312.2833317. URL http://doi.acm.org/10.1145/2833312.2833317.

Melanie Swan. Blockchain thinking : The brain as a decentralized autonomous corporation [commentary]. *IEEE Technol. Soc. Mag.*, 34(4): 41–52, 2015. doi: 10.1109/MTS.2015.2494358. URL http://dx.doi.org/10.1109/MTS.2015.2494358.

Sam Toueg, Kenneth J. Perry, and T. K. Srikanth. Fast distributed agreement. *SIAM J. Comput.*, 16(3):445–457, 1987. doi: 10.1137/0216031. URL http://dx.doi.org/10.1137/0216031.

Florian Tschorsch and Björn Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys and Tutorials*, 18(3):2084–2123, 2016. doi: 10.1109/COMST.2016.2535718. URL http://dx.doi.org/10.1109/COMST.2016.2535718.

Henk C. A. van Tilborg and Sushil Jajodia. Proof of work. In *Encyclopedia of Cryptography and Security, 2nd Ed.*, page 984. Springer, 2011. doi: 10.1007/978-1-4419-5906-5_1060. URL http://dx.doi.org/10.1007/978-1-4419-5906-5_1060.

Eric Wustrow and Benjamin VanderSloot. 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. URL https://www.usenix.org/conference/woot16/workshop-program/presentation/wustrow.

Table 2: Table of Notations

κ	a security parameter of the signature scheme
λ	the number of new blocks required to confirm a block, serves as another security parameter of the block chain
Δ	the upper bound of network delaying
R	a round number (similar to the notion of "day" in Pass and Shi's hybrid consensus Pass and Shi [2017b])
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, $\mathsf{csize} = O(\lambda)$
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
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 hash 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 the i-th committee member
$cand_i$	the identity (i.e., public key) of the 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
CD_R	$\mathsf{CD}_R = \{cand_1, cand_2, \ldots, cand_N\}$ is the identity list of round-R's candidates
t'	the expected time length of each round
$\mathsf{PRF}(k, R)$	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 $\mathbb{Z}_{2^{\kappa}}$
$\mathsf{header}(\boldsymbol{B})$	the header of block $\boldsymbol{B}$, including the public key of the proposer, the hash of included transactions, and other auxiliary information

Table B.3: Switchover techniques in the candidate side

CANDIDATE SIDE (in round R, for candidate i)
•Pack $\boldsymbol{B}_{R-1} := \{\mathsf{rec}_{R-1}, H(\mathsf{header}(\boldsymbol{B}_{R-2})), \mathsf{CM}_R\}$; •Try to find as much as possible nonce(s) $\mathsf{nc}_1, \mathsf{nc}_2, \ldots, \mathsf{nc}_\ell$, so that $H(\mathsf{header}(\boldsymbol{B}_{R-1}), \mathsf{ID}_i, \mathsf{nc}_j) \in \mathsf{target}$ 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);

Table B.4: Switchover techniques in the committee side

COMMITTEE SIDE (in round R, for committee member i)
Step 1 •Pack $\boldsymbol{B}_{R-1} = \{\mathsf{rec}_{R-1}, H(\mathsf{header}(\boldsymbol{B}_{R-2})), \mathsf{CM}_R\}$; •Check its identity in round-R committee list CM_R;
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;
Step 3 •Calculate $L_j := G(x_j, y_j) = G(\frac{\mu}{\mathbb{E}[w]} \cdot w_j, \frac{\mu}{\mathbb{E}[s]} \cdot s_j)$ for each candidate j; •Calculate $\mathsf{sum}_L := \sum_{i \in \mathsf{CD}_R} L_j$; •Calculate k_R as xor summation of all received nonces passed though the consensus; •Calculate $\mathsf{rand}_i \leftarrow \mathsf{PRF}(k_R, i) \cdot (\mathsf{sum}_L/2^\kappa)$ for each $1 \leq i \leq \mathsf{csize}$; •Find first t_i that $\sum_{j=1}^{t_i-1} L_j \leq \mathsf{rand}_i < \sum_{j=1}^{t_i} L_j$ for each $1 \leq i \leq \mathsf{csize}$; •Claim member list of the next round is $\mathsf{CM}_{R+1} = \{cand_{t_1}, cand_{t_2}, cand_{t_3}, \ldots, cand_{t_{\mathsf{csize}}}\}$; •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 on the header of $\boldsymbol{B}_R$.

Shuyang Tang received the BS degree in Computer Science and Technology in Shanghai Jiao Tong University in 2018. He is currently a PhD candidate in Shanghai Jiao Tong University. His research interests include Program Semantics, Interactive Program Verification and Cryptocurrency.

Zhiqiang Liu received the BS and MS degrees in Mathematics, and the PhD degree in Cryptography from Shanghai Jiao Tong University in 1998, 2001, and 2012, respectively. He is currently an associate professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include Symmetric-Key Cryptography, Post-Quantum Cryptography, Cryptocurrency, Block Chain and DAG-based Distributed System.

Sherman S.M. Chow joined the Department of Information Engineering at the Chinese University of Hong Kong in November 2012, and received the Early Career Award 2013/14 by the Hong Kong Research Grant Council. Before CUHK, he was a research fellow at Department of Combinatorics and Optimization, University of Waterloo, a position he commenced after receiving his Ph.D. degree from the Courant Institute of Mathematical Sciences, New York University. He interned at NTT Research and Development

(Tokyo),Microsoft Research (Redmond) and Fuji Xerox Palo Alto Laboratory, and has made research visits to Friedrich-Alexander-U., U. Maryland, U. Calgary, U. Texas, HKU, MIT, Academia Sinica, and Queensland University of Technology. He has published in major conferences such as CCS, EUROCRYPT, ITCS, NDSS, and Usenix Security.

Zhen Liu received the BS and MS degrees in Mathematics, and the PhD degree in Cryptography from Shanghai Jiao Tong University in 1999, 2002, and 2013, respectively. He is currently an special researcher in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include Applied Cryptography, Information Security, Attribution-based Encryption, Cloud Computing Security, Big Data Security, Blockchain and Cryptocurrency.

Yu Long received the BS from Southwest Jiao Tong University, MS and the PhD degree in Computer Science from Shanghai Jiao Tong University in 2002, 2005, and 2008, respectively. She is currently an associate professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include Public Key Encryption and Blockchain.

1. **The first fork-free PoW-based blockchain in the permissionless environment.** In bitcoin, the integrity of transactions in a block is guaranteed by fork resolutions (e.g., blocks including double-spending transactions are resolved), since any malicious branch can be outraced by an honest one. We employed the paradigm of hybrid consensus which leverages the security of PBFT to get rid of fork resolution while ensuring transaction integrity. To the best of our knowledge, achieving fork-free property in this way is not yet identified by the literature including the work of Pass and Shi.

2. **Reducing variance without centralized mining pools.** Traditional PoW crucially relies on accepting a single hash puzzle to ensure that existing records cannot be tampered with. Our proposed functionality of generalized PoW accepts multiple solutions for the same puzzle in each round, this reduces the mining-revenues variance. This functionality is hard to realize in bitcoin since its setting provides nothing to "operate" on different solutions. But our fork-free hybrid consensus achieves this functionality by leveraging a rotating committee.

3. **Flexible hybrid of PoW and PoS.** We construct a flexible PoA by having a committee perform the election based on a hybrid weight regarding the participants' PoW power w and the PoS capability s. The relationship between the hybrid weight w and s can be flexibly determined according to different scenarios. To our knowledge, such a flexibility is never considered in previous works.