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

No Author Given

No Institute Given

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 computing 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 combinations of two mechanisms. On the other hand, the hybrid consensus protocol of Pass and Shi (ePrint 16/917) 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 newly proposed generalized PoW to construct a fork-free hybrid consensus protocol, which addresses issues faced by the existing hybrid consensus mechanism. We further combine our fork-free hybrid consensus mechanism with PoS for a flexible version of PoA, which offers a flexible combination of PoW and PoS. Compared with Bentov et al.'s PoA, our "*flexible* PoA" improves the efficiency and provides more flexible combinations of PoW and PoS, resulting in a more powerful and applicable consensus protocol.

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 a decentralized ledger among other applications. It is considered to be commencing a revolution in information technology and economics [BMC+15,Swa15,TS16]. Multiple decentralized cryptocurrencies are also devised [AMN+16,SBRS16,WV16]. Bitcoin utilized blockchain, which is referred to as "Nakamoto chain" (for differentiating it from later proposals) for an implicit consensus mechanism keeping a distributed ledger of blocks, which grows by time. Each block includes an ordered list of transactions. Blockchain is built upon the methodology of *Proof-of-Work* (PoW) [TJ11], which requires the creator of a new block to solve a hash puzzle (a hash puzzle regarding content w is to find a solution x so that $H(x||w)$ falls into a target range) regarding the hash of the previous block, an ordered list of transactions, as well as other necessary information. 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. Note that since later on we will define an alternative proof of hash power named as "generalized PoW", the PoW method mentioned above is referred to "traditional PoW", or just "PoW" when no ambiguity exists.

When multiple new blocks are generated "simultaneously" following the same previous block, disagreement emerges and 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. Any other branches will be discarded. 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".

Serving as a core part of the consensus protocol underlying the bitcoin, PoW shows several potential merits such as openness to any participant and good robustness . Yet, since the hardness of the puzzle should be great enough so that only one block in expectation can be solved in a certain period of time (like bitcoin's ten minutes), PoW-based protocols often confirm the validity of a newly added block at an unsatisfactory speed. Also, the hardness of the puzzles causes a centralization of computing power which is already happening, since one individual may take years to find a puzzle solution. To address these issues, *Proof-of-Stake* (PoS) [Qua11,BGM16] is proposed to replace PoW which moves the decision basis from computing power to possession of stake in the system (e.g., in the form of cryptocurrency). Yet, while the specific risk of having a few mining farms dominating PoW is mitigated, it still faces another kind of centralization risk (from large stakeholders), and other risks due to an economic phenomenon known as the "tragedy of the commons" [BLMR14].

A step further, *Proof-of-Activity* (PoA) proposed by Bentov et al. [BLMR14] aims to inherit the advantages of both PoW and PoS. PoA determines the miner of a new block by taking into account both its computing power as well as its stake. We just borrow the terminology of PoA to signify a combination of PoW and PoS, Bentov et al.'s construction of PoA is not crucial to this article.

Practical Byzantine Fault Tolerance (PBFT) algorithm [CL99] provides a high performance Byzantine state machine replication for tolerating certain failures in Byzantine general problem among many other BFT protocols [PSL80,LSP82,TPS87], which has been widely adopted in the field of 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 $O(n^2)$. This is a permissioned protocol, while applicable to a permissionless environment with the delicate construction of hybrid consensuses (both Pass and Shi's and our newly proposed).

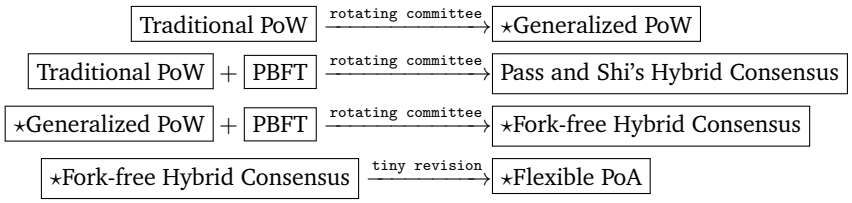

Fig. 1. Conceptual Structures Among Terminologies (our innovations are marked with stars)

Hybrid Consensus proposed by Pass and Shi [PS17b] adopts both blockchain's PoW (a protocol under a permissionless environment) and PBFT (a protocol under a permissioned environment). It utilizes Nakamoto chain or Fruitchain [PS17a] (both chains adopt traditional PoW) as the underlying blockchain (called "snailchain" for its slow growth rate) to dynamically maintain a "rotating committee" (in short, the committee) that serves as the leader of transaction confirmations. All transactions are verified by the committee via a PBFT among committee members. The consensus protocol goes by rounds. Committee members of each round correspond to miners of a fixed sequence of confirmed on-chain blocks. For example, the committee of round R are miners of the $(R-1)\mathsf{csize}^{\text{th}}$ to $(R \cdot \mathsf{csize} - 1)^{\text{th}}$ blocks (where csize is the committee size). In such a way, the blockchain provides PoW based committee election, while the validation of transactions is separated from the blockchain, since transactions are validated through a PBFT network among committee members. This inherits the efficiency advantage of PBFT and speeds up transaction confirmations significantly in a permissionless environment (where anonymous participants can join or leave dynamically, like in bitcoin). We note that chain fork happens in existing hybrid consensus's underlying snailchain, during the committee election.

Table 1. Comparisons between Consensus Schemes

Consensus Scheme	Efficiency	Fork-free	PoW	PoS	Incentive of Presence	Flexible Combination
Classical PoW [TJ11]			✓		✓	
Ideal PoS [Qua11]	✓	✓		✓		
Hybrid Consensus (existing) [PS17b]	✓		✓		✓	
Proof-of-Activity [BLMR14]			✓	✓	✓	
Fork-free Hybrid Consensus	✓	✓	✓		✓	
Flexible PoA	✓	✓	✓	✓	✓	✓

We aim to achieve a fork-free consensus and an accurate PoW power evaluation (which helps our later constructions), thereby we for the first time change the principle of blockchain mining so that multiple puzzle solutions can be found each round (we name such a principle as "the generalized PoW"), all of these solutions are submitted to a committee directly without causing any fork and all of them are recorded (so that the history of records is still hard to forge).

In Pass and Shi's hybrid consensus, a committee is elected by the blockchain to verify transactions, who are miners of certain blocks. Based on Pass and Shi's scheme, we construct a scheme and call it the fork-free hybrid consensus, which 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.

Upon the fork-free construction, we further revise the rule of committee election so that more election principles can be constructed. Specifically, a function can be established to assign a weight to each candidate according to its PoW power and its PoS capability, and the election can be based on such a weight. In this way, an efficient and flexible PoA protocol is formed for the first time to our knowledge. Since it is based on the fork-free hybrid consensus, the flexible PoA is also endowed with a fork-free property.

Technical Novelty of Our Work

1. In ordinary PoW, the hardness of a single hash puzzle is crucial to the hard-to-tamper property of records. We discover that, the hard-to-tamper property can also be determined by multiple puzzle solutions instead of one. With this methodology, we propose the functionality of "generalized PoW", which, for the first time, ensures security when multiple solutions for the same puzzle are accepted in each round (if securely realized). This functionality is hard to be realized in bitcoin since there is no trustful third party (TTP) to conduct certain protocols on all nonce solutions. This leads to the first fork-free consensus protocol, which securely realizes this functionality.
2. In bitcoin, the integrity of transactions in block is guaranteed by fork resolutions (e.g., blocks including double-spending transactions are resolved), since any malicious branch will be competed by honest ones. However, with Pass and Shi's hybrid consensus, such an integrity is guaranteed by security properties of PBFT. That is to say, fork resolution is no longer a necessity to transaction integrity for a hybrid consensus based cryptocurrency system. However, to our knowledge, such an unnecessity is not mentioned in Pass and Shi's work and moreover not yet found by anyone to day. Therefore, we rid our newly proposed consensus schemes of forks without security concerns for the first time.
3. We construct a flexible hybrid of PoW and PoS by having a committee perform the election based on a combined weight regarding the participants' PoW power w and the PoS capability s simultaneously. The relationship between such a weight, w and s can be determined according to different scenarios, and hence a flexibility is achieved. To our knowledge, such a flexibility is never considered in previous works.

Paper Organization

The remainder of this paper is organized as follows. Sec. 2 introduces notations and the security model (as well as security goals). 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 the flexible PoA. Strategies and the security analysis under different cases are provided for the flexible PoA in the appendix. Finally, in Sec. 5, we provide an analysis to shown that our newly proposed fork-free hybrid consensus and the flexible PoA achieve our security goals.

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 "date" in Pass and Shi's hybrid consensus [PS17b])
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, 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
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 proposer, the hash of included transactions, and other auxiliary information

2 The Model

2.1 Notations

The set of natural numbers $\{1, 2, \ldots, N\}$ is denoted by $[N]$. "$x||y$" denotes the concatenation of x and y. "$A := B$" assigns B to the variable A. Table 2 lists more notations. A "*node*" is either a candidate of leader election (i.e. election of next round's committee member) or a current member of the committee.

2.2 Security Model

1. **Network.** We follow the security and network assumptions of Pass and Shi's hybrid consensus [PS17b], where 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. And we 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.** Moreover, we assume a peer-to-peer network without trust on any specific peer, while it can be made sure that over α fraction of the computing power and over β of stake are at hands of honest participants.
3. **Other Assumptions.** Also, we assume the collision-resistant property of cryptographic hash functions. PBFT is executed ideally as long as over $2/3$ participants are honest.
4. **Security Goals.**
   – **Fork-Freeness.** To throughly eliminate the selfish mining, and speed up transactions confirmation (in bitcoin, users have to wait long to make sure one block will not be erased by chain forks), we require a novel consensus scheme without chain forks.
   – **Hard-to-Forge (Hard-to-Tamper).** Any adversary with less than half total hash power should have no capability of maintaining another forged chain of valid blocks.
   – **Chain Quality.** Chain quality means the fraction of blocks generated by honest participants, in a fork-free consensus scheme, the quality must be $1 - negl(\lambda)$ for some negligible function $negl(\cdot)$, and the security parameter λ. Since faulty blocks will stay on chain instead of being eliminated by chain forks. The chain quality goal of our newly proposed two schemes depends on the $2/3$ honesty of PBFT participants. That is, over $2/3$ PBFT participants must be honest ones. This must be met, or else the malicious parties can manipulate transaction confirmations and so forth honest transactions may not be confirmed while the malicious ones can. Due to this, the probability of malicious party's becoming a leader, i.e. entering the rotating committee (see Sec. 3) should be less than $1/3 - \epsilon$ for some marginal ϵ.
   – **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.
   – **Robustness.** Since our scheme is fork-free, any block generated by the committee shouldn't be faulty except for a negligible probability. In the view of theoretical cryptography, a negligible function is satisfiable. However, due to the limitation of the committee size, the outcome of a negligible function may not be totally negligible enough in practice. Therefore, the demand for the robustness requires that the protocol can resume its functionality even in the worst case (even a case happens with a negligible probability). Specifically, even if the all participants of one committee are corrupted, the next committee should have over $2/3$ honest members with an overwhelming probability, no matter how the adversary deviates from the protocol.
   – **Liveness.** The liveness of the system requires certain participation. Moreover, our protocol should not encourage participants to split their power into spawned nodes. Or else the network, and hence the liveness, might be undermined by an overflow throughput.
   – **Communication Complexity.** The communication complexity should be considered. Specifically, the communication complexity denotes all the number of all interactions should be made (include delivery of blocks from proposers to all network nodes, and all interactions among consensus participants (either for the consensus or leader elections).

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

In this section, we firstly introduce Pass and Shi's hybrid consensus. Then we propose the functionality of our generalized PoW, and argue its merits. Afterwards, the fork-free hybrid consensus is demonstrated to realized the generalized PoW.

Pass and Shi's Hybrid Consensus

In Pass and Shi's hybrid consensus [PS17b], a novel primitive is proposed to combine 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), for the first time to our knowledge. In such a way, the efficiency quality of permissioned protocols is leveraged in a permissionless environment.

More specifically, the blockchain no longer serves the direct validation of transactions, but is the basis of the election of a rotating committee, and all transactions are validated via this committee. That is, each round (same to the term "day" in [PS17b]) R's committee $\mathsf{CM}_R = \{com_1, com_2, \ldots, com_{\mathsf{csize}}\}$ of size csize are miners of certain blocks of the underlying blockchain. Each round, transactions are validated via PBFT of the committee. In detail, the protocol proceeds as follows.

1. In round R, miners of the $(R-1)\mathsf{csize}^{\text{th}}$ to the $(R \cdot \mathsf{csize} - 1)^{\text{th}}$ blocks on chain are chosen to be committee members. It is tolerated that some members may share the same identity (i.e. one participant mines more than one block).
2. Each committee member begins a PBFT instance, during which transactions are proposed in turn from leader's memory pools. Each proposal is validated as long as over $\mathsf{csize}/3$ members show approvals to it.
3. When csize new blocks, i.e. the $R \cdot \mathsf{csize}^{\text{th}}$ to the $((R+1)\mathsf{csize} - 1)^{\text{th}}$ blocks, are confirmed on the underlying blockchain, the committee performs a switchover, thereby the next round begins.

Generalized Proof-of-Work

We newly 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 attained by participants. These nonce solutions are collected by the functionality and csize of them are randomly selected. The csize leaders are determined as the finder of these randomly selected nonce solutions. In such a way, the election of csize leaders is attained, without compromising the fairness (that is, 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 $\bar{\mathcal{G}}_{\mathsf{GPoW}}$. These nonce solutions are received and arranged by $\bar{\mathcal{G}}_{\mathsf{GPoW}}$ into an array W. Afterwards, csize random numbers $(\mathsf{rand}_1, \mathsf{rand}_2, \ldots, \mathsf{rand}_{\mathsf{csize}})$ are generated within $\bar{\mathcal{G}}_{\mathsf{GPoW}}$. Finally, the identity of next round's committee members are determined as rand_i-th's items of W (for $i \in [\mathsf{csize}]$). The formal description of the functionality is shown in Fig. 2.

In this way, the more hash puzzle solutions are found, a greater chance (proportional to the number of solutions found) of being elected is attained. Obviously, the expected number of nonces found is proportional to each participants' computing power. By combining the two facts, surely the chance of being elected is still proportional to candidates' PoW ability likewise traditional PoW. Our newly proposed protocol is referred to as "generalized PoW", since traditional PoW can be viewed as a special case of our newly proposed primitive where the second solution is forbidden and with $\mathsf{csize} = 1$.

Next, we discuss the significance of our newly proposed generalized PoW. Despite the fact that our newly proposed generalized PoW facilitates the simultaneous election of multiple leaders, our generalized PoW 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. 4), we hope to assign a "score" w_i for each candidate, to evaluate the computing power (hash rate) of the candidate. To form an accurate evaluation, w_i's should be proportional to candidates' real computing power, with less variance. We now formally compare the generalized PoW with the traditional one concerning the accuracy of the computing power evaluation. In fact, the expected w_i's under two protocols can be regarded as proportional to candidates' computing power, therefore we make comparisons on their coefficients of variance and finally determine that our new construction is more satisfiable.

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

Shared Functionality $\bar{\mathcal{G}}_{\mathsf{GPoW}}$ interacts with all parties (candidates) $P_1, P_2, \ldots, P_N$ (the first $k < N/3$ of them are controlled by the adversary), and the environment $\mathcal{Z}$, the adversary $\mathcal{A}$, as well as a publicly shared global clock functionality $\bar{\mathcal{G}}_{\mathsf{CLOCK}}$.
This functionality is parameterized by the number of candidates N (this is a variant in the permissionless setting, but we take this notation for simplicity of descriptions), and 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 Issue.

• Attain puzzle m from the environment $\mathcal{Z}$, issue m to the adversary $\mathcal{A}$ all honest candidates $P_{k+1}, P_{k+2}, \ldots, P_N$.
• Query the global time clock $\bar{\mathcal{G}}_{\mathsf{CLOCK}}$ to attain 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. Denote this array as $\boldsymbol{W}$, initially set as $\boldsymbol{W} \leftarrow \emptyset$.
• Set variables $\ell_1, \ell_2, \ldots, \ell_N$ as zeros.
• 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 $\bar{\mathcal{G}}_{\mathsf{CLOCK}}$ for time t, go back to the previous step if $t < t_0 + t'$.

–Member Release.

• Generate csizerandom numbers $\mathsf{rand}_1, \mathsf{rand}_2, \ldots, \mathsf{rand}_{\mathsf{csize}} \in \left[\sum_{i=1}^{N} \ell_i\right]$.
• Find the $\mathsf{rand}_i^{\mathsf{th}}$ items in $\boldsymbol{W}$ for each $i \in [\mathsf{csize}]$, denote them as $(\mathsf{ID}_{\mathsf{CM}_1}, \mathsf{nc}_{\mathsf{CM}_1}), (\mathsf{ID}_{\mathsf{CM}_2}, \mathsf{nc}_{\mathsf{CM}_2}), \ldots, (\mathsf{ID}_{\mathsf{CM}_{\mathsf{csize}}}, \mathsf{nc}_{\mathsf{CM}_{\mathsf{csize}}})$.
• 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.

Fig. 2. The Generalized PoW Functionality

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. Moreover, we denote the expected number of valid hash puzzle solutions found by this candidate in one round as

$$\gamma_T := \frac{M_0}{M}T.$$

Traditional PoW. In this article, we refer to the existing method of proof-of-work as the traditional PoW. Traditional PoW can be viewed as such a 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 computing power T, by definition:

$$\mathbb{E}[w_i] = \gamma_T.$$

Since in bitcoin, the chance of one participants' finding more than one puzzle solution is negligible, we regard that w_i satisfies a binomial distribution, hence

$$\mathrm{Var}[w_i] \approx \mathbb{E}[w_i](1 - \mathbb{E}[w_i]) = \gamma_T(1 - \gamma_T).$$

And the coefficient of variance is

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

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

Generalized PoW. For our generalized PoW, we lower the difficulty so that a candidate with considerable computing power may find more than one solutions to a hash puzzle. Final 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 (e.g. mining pool) holding 10% overall computing 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 computing 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 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, in case $\gamma_T = 10$, 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 newly proposed fork-free hybrid consensus that securely realizes the generalized PoW $\bar{\mathcal{G}}_{\mathsf{GPoW}}$.

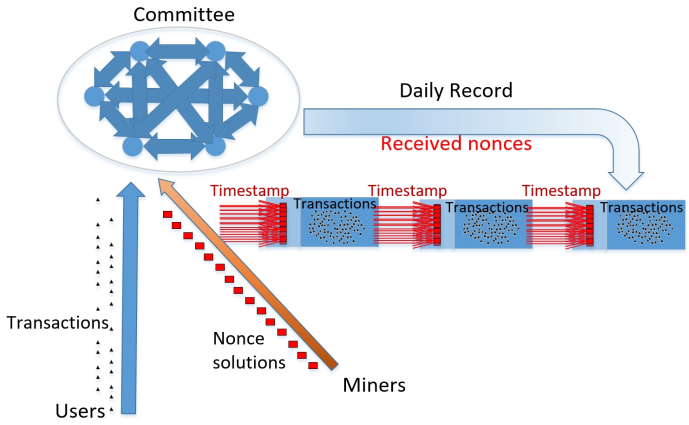

Fig. 3. Fork-free hybrid consensus

Fork-free Hybrid Consensus

In this part, we introduce our newly proposed fork-free hybrid consensus, which is a realization of our proposed generalized PoW. Similarly to that of the existing hybrid consensus, we adopt a committee of size csize which is rotated every "round" (similar to what is called "Day" in Pass and Shi's hybrid consensus). Transactions are verifies by this committee via PBFT. Except for the generators of the first csize blocks (see more bootstrapping details in appendix sec. B.1), the only way to non-committee member miners' (also referred to as "candidates") entering the committee is the committee election via the generalized PoW. In another word, each committee is elected from the previous committee. Now we present the protocol of our fork-free hybrid consensus protocol. For simplicity, we order all committee members in $1, 2, \ldots,$ csize.

Note that differently 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 previous round's committee members.

1. In round R, each candidate, say, u, collects transactions and nonce records of round $R-1$ (signed by over $1/3$ committee members) as rec_{R-1}, and receives committee members' signatures on the previous block header. Next, it recovers previous block $\boldsymbol{B}_{R-1} = \left\{\mathsf{rec}_{R-1}, H\left(\mathsf{header}\left(\boldsymbol{B}_{R-2}\right)\right), \mathsf{CM}_{R-1}\right\}$, aborts this procedure if $\mathsf{header}\left(\boldsymbol{B}_{R-1}\right)$ does not match over $1/3$ of committee members' block header signatures.
2. The committee of round R is assembled according to CM_{R-1}. Committee members start an instance of PBFT that reaches consensus on candidates' puzzle solutions and newly received transactions.
3. After that, 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

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

4. After the procedures above, 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 W, and sorts all items in the same order, to get

$$W = \begin{bmatrix} \mathsf{nc}_{A,1} & \mathsf{ID}_A \\ \mathsf{nc}_{B,1} & \mathsf{ID}_B \\ \vdots & \vdots \\ \mathsf{nc}_{u,1} & \mathsf{ID}_u \\ \mathsf{nc}_{u,2} & \mathsf{ID}_u \\ \vdots & \vdots \\ \mathsf{nc}_{u,P_u} & \mathsf{ID}_u \\ \vdots & \vdots \end{bmatrix}_{|W|\times 2}$$

   At the termination of this round, committee members in $\mathsf{CM}_R = [\mathsf{ID}_1, \mathsf{ID}_2, \ldots, \mathsf{ID}_{\mathsf{csize}}]$ calculate the xor-summation of all received nonces that have passed though the PBFT consensus (denoted as k_R). After that, csize nonces are determined according to k_R among the received nonces. The committee of the next round is set to the miners of csize determined nonces.
6. Finally, after the reward negotiation (to be described in the introduction to honesty incentives), committee members broadcast rec_R and their signatures on $\mathsf{header}(\boldsymbol{B}_R)$, where $\boldsymbol{B}_R = \{\mathsf{rec}_R, H(\mathsf{header}(\boldsymbol{B}_{R-1})), \mathsf{CM}_R\}$. The csize lucky candidates in CM_R are enrolled into the committee of next round.

Reward Negotiation

In this construction, incentive of participation is guaranteed for both miners and committee members. Miners should work honestly with great efforts to enter the committee. Also, committee members will participant in PBFT and block generation to have more valid transactions pass through PBFT and reach a higher transaction fee. Moreover, any adversary behavior leads to no marginal reward so unfriendly behaviors can be discouraged.

To further guarantee honesty and the 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 termination of this round, along with proper signatures.

Merits of Fork-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:

- **Fork-freeness.** Obviously, blocks of our chain are generated once for all, without any necessity of chain competitions. Hence, fork is totally eliminated.
- **Accuracy in computing power evaluation.** Apart from the fork-free property, our evaluation of miners' computing power based on the generalized PoW has a smaller variance, as discussed above.
- **Friendliness in face of delays.** Moreover, 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 a 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. In our work, the assumption on τ can be much looser than Pass and Shi's hybrid consensus. In Pass and Shi's constructions, once a node is elected into the committee, it waits till the generation of sufficient new blocks to start to work, causing a long exposure to adversary's target corruption. On the contrary, with our construction, csizecommittee members are elected simultaneously, without such an exposure period.

4 The Flexible Proof-of-Activity

We propose an alternative proof-of-activity to support flexible combinations of generalized PoW and PoS. More specifically, we build our consensus protocol based on the framework of the fork-free hybrid consensus, by a novel principle of rotating committee election. 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.

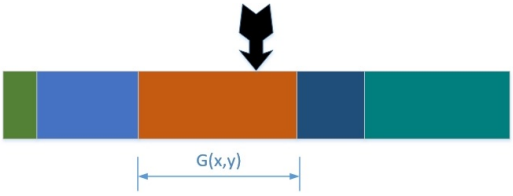

Fig. 4. Combination of PoW capability x and PoS value y

Detailed Protocols

We formally present the protocol of miners for each specific round. We will discuss protocols for candidates and committee members separately, detailed illustrations of protocols are shown in Tables 3 and 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, since they share the same meanings.

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 $G(w, s)$. Here, we assume that w's and s's are normalized to x's and y's so that

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

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 $\mathbb{R}^+$. Furthermore, y should be greater than 1 when the logarithm over it exists, this makes sense since stakeholders should have a certain stake.

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 header $\mathsf{header}(\boldsymbol{B}_{R-2})$ (to make records hard-to-tamper) 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 it satisfies $H(\mathsf{header}(\boldsymbol{B}_{R-1}), \mathsf{ID}_i, \mathsf{nc}_j) \in \mathsf{target}$ for all $1 \leq j \leq \ell$. Then, it submits $\{\mathsf{nc}_1, \mathsf{nc}_2, \ldots, \mathsf{nc}_\ell\}$ to committee members.
3. It receives rec_R (with corresponding signatures) at the end of the round.

Committee member. For committee members in round R:

1. Each miner 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 $\boldsymbol{B}_{R-1} = \{\mathsf{rec}_{R-1}, H(\mathsf{header}(\boldsymbol{B}_{R-2})), \mathsf{CM}_{R-1}\}$.
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. They then calculate k_R as the xor-summation of all accepted nonces, and decide csize lucky candidates 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 broadcasts rec_R and the signature on $\mathsf{header}(\boldsymbol{B}_R)$, declaring the termination of a round, where $\boldsymbol{B}_R = \{\mathsf{rec}_R, H(\mathsf{header}(\boldsymbol{B}_{R-1})), \mathsf{CM}_R\}$.

Table 4 shows the detailed procedures. Strategy and security analyses of this scheme are shown in the appendix.

5 Security Analysis

In this part, we provide a security analysis for our newly proposed consensus schemes. Since the fork-free hybrid consensus shares most security properties with the flexible PoA, in the following discussions, we refer to both schemes if no specification is made.

1. **Fork-Freeness.** In our newly proposed hybrid consensus, fork is eliminated since record for each round is generated by the committee once for all. Thereby, differently from the bitcoin blockchain, there is no "a graph of blocks" in our constructions.
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 competes 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 power 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. Therefore, when two history of chains are found, the one with more total nonce solutions competes the other one before the generation of λ new block, i.e., before confirmation of the block that the adversary tries to forge.

3. **Chain Quality.** As discussed in the security model, we required a $1-negl(\lambda)$ overall chain quality of the block chain, with a negligible function $negl(\cdot)$. We now prove that we have reached this aim.

   Theorem 1 (The Achievement of A $1-e^{-\Omega(\lambda)}$ Chain Quality). *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. We denote $\alpha = \frac{1}{3} - \epsilon$ 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), event Win as adversary's successfully controlling 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 $\mathbb{E}[X] = \alpha \cdot \mathsf{csize}$ as number of controlled members in one attempt. By a 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 $\mathsf{csize} = \Theta(\lambda)$,

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

   and in the complementary sense, the probability of each block's being honestly generated, and hence the chain quality is $1 - e^{-\Omega(\lambda)}$.

4. **Looser Assumption Against Mildly Agile Corruptions.** In our work, the assumption on 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.
5. **Robustness.** This construction is robust even in case of the worst incident than may exist among all traces of views, in which an adversary controls the whole committee. In such a case, the adversary may try to control the committee of the next round by ignoring or adding nonces in the nonce acceptance step for polynomial number of attempts (denoted as $\mathsf{attempt}(\kappa)$). However, the probability for its controlling over $2/3$ is negligible in contrast to $\frac{1}{\mathsf{attempt}(\kappa)}$, thereby the committee of next round is not controlled by the adversary. More formally, following the formulation above, adversary's probability of succeeding in any attempt is

   $$1 - (1 - \Pr[\mathsf{Win}])^{\mathsf{attempt}(\kappa)} \approx \mathsf{attempt}(\kappa)\Pr[\mathsf{Win}] \ll 1,$$

   in that $\Pr[\mathsf{Win}] \ll \frac{1}{\mathsf{attempt}(\kappa)}$.
6. **Liveness.** In our newly proposed fork-free hybrid consensus, Our liveness is provided as long as there is certain number of participation. In the flexible PoA, the liveness requires that the function $G(\cdot,\cdot)$ does not encourage participants to split their power (either hash power or stake) into spawned node and undermine the network. This is met as long as "concaveness" of the function is guaranteed (details are shown in the appendix).
7. **Communication Complexity.** All nonce solutions are submitted to the committee just like the transactions. It is the committee that runs a PBFT to reach agreements 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. Therefore, the communication complexity is roughly the same as that of Nakamoto consensus, in which the communication cost is $O(n)$.

6 Conclusion

We generalized the classical PoW to make it fork-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 alternative PoA. We firstly built our consensus protocol based on the framework of Pass and Shi's 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 the flexible PoA under different strategies of combining PoW and PoS. Taking the advantage of PoS, the flexible PoA is an improvement of hybrid consensus. Moreover, compared with Bentov et al.'s PoA, the flexible PoA further improves the efficiency and provides a more flexible combination of PoW and PoS.

Fork-free hybrid consensus or the flexible PoA could be adopted in blockchains requiring an efficient and flexible consensus mechanism. For future work, it will be interesting to consider appropriate privacy enhancements 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)

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)

PS17a. Pass, R., Shi, E.: Fruitchains: A fair blockchain. In: Schiller, E.M., Schwarzmann, A.A. (eds.) Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017, pp. 315–324. ACM (2017)

PS17b. Pass, R., Shi, E.: Hybrid consensus: Efficient consensus in the permissionless model. IACR Cryptology ePrint 2016/917 (2017)

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 et al.: Proof of stake instead of proof of work. Bitcoin forum (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 Strategy and Security Analysis for Sec. 4

Table 3. Switchover techniques in the candidate side

CANDIDATE SIDE (in round R, for candidate i)
•Pack $\boldsymbol{B}_{R-1} := \{\mathsf{rec}_{R-1}, H(\mathsf{header}(\boldsymbol{B}_{R-2})), \mathsf{CM}_{R-1}\}$;
•Try to find as much as possible nonce(s) $\mathsf{nc}_1, \mathsf{nc}_2, \ldots, \mathsf{nc}_\ell$, so that $H(\mathsf{header}(\boldsymbol{B}_{R-1}), \mathsf{ID}_i, \mathsf{nc}_j) \in \mathsf{target}$ for all $1 \le j \le \ell$;
•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 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 $\boldsymbol{B}_{R-1} = \{\mathsf{rec}_{R-1}, H(\mathsf{header}(\boldsymbol{B}_{R-2})), \mathsf{CM}_{R-1}\}$;
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 $\mathsf{sum}_L := \sum_{j \in \mathsf{CD}_R} L_j$;
•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 $1 \le i \le \mathsf{csize}$;
•Find first t_i that $\sum_{j=1}^{t_i - 1} L_j \le \mathsf{rand}_i < \sum_{j=1}^{t_i} L_j$ for each $1 \le i \le \mathsf{csize}$;
•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 $\boldsymbol{B}_R$.

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

We will discuss the following two cases separately: one for a concave establishment of G and one for the otherwise. The strategy of the adversary will be different in two cases. In the 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 to multiple identities it spawned, thereby maximizing 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.

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 the next round is maximized when ℓ reaches some value greater than 1 (i.e. division of x' and y' exists in the optimal strategy).

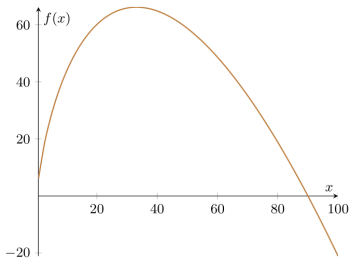

Fig. 5. Function $f(x) = x(a - 2\ln x)$ with $a = 9.0$

In fact, under our assumption, this candidate tends to maximize the total probability of (at least one spawned node's) being elected:

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

After simple derivations, we can see that this probability 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 further implementing.

We define the adversary advantage $\mathbf{Adv}_{\alpha,\beta}$ as the upper bound approximation for the possibility of a maliciously spawned node's entering next round's committee:

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

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. Since it is a upper bound corresponding to the worst situation, we consider that all malicious parties are cooperating.

The foundation of flexible 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 $G(x,y)=\sqrt{xy}$ as an example.

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,

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

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 B.3 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. For such $G(x,y)$, 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 $(x_1,y_1),(x_2,y_2)\in\mathbb{R}^+\times\mathbb{R}^+$:

$$\begin{aligned}&G(x_1+x_2,y_1+y_2)=(x_1+x_2)\ln(y_1+y_2)\\&>x_1\ln y_1+x_2\ln y_2=G(x_1,y_1)+G(x_2,y_2).\end{aligned}$$

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

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

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.

B More Details

B.1 Bootstrapping Techniques

To bootstrap the system, we need csize genesis blocks maintained by the first few participants. Differently from the bitcoin, the system launcher (i.e. the proposers of csize genesis blocks) should have certain computing power to perform the consensus for the first csize rounds.

B.2 Determination on Commencement and Termination Time

All users' transactions and candidates' nonces are submitted to the committee, and committee members reach the consensus via PBFT. PBFT is an ordered procedure during which transactions are proposed by one PBFT participant (i.e. committee members), and all members take part in the decision of their validity in the same sequential order. With this property, we can stipulate that each round is terminated at the M^{th} proposal within the PBFT process, where M is a predetermined parameter.

B.3 Generalized PoW under Network Delay

In the following contents, for simplicity, we consider N candidates share 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 $0 < c \ll 1$ and natural number N, $c\sum_{i=0}^{\infty}(1-c)^{(iN)} - \frac{1}{N} = o(\frac{1}{N})$.*

Proof.

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

Then we get

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

Lemma 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:

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

Meanwhile,

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

Since for sufficiently large N:

$$\begin{aligned}
(1-c)^{(N-1)d}&\ll\frac{\Delta'-\delta}{\Delta'-\delta+d},\\
\sum_{i=\delta}^{\infty}(1-c)^{i-\delta}\cdot c\cdot(1-c)^{(N-1)(i-\delta')}&<\frac{\Delta'-\delta}{\Delta'-\delta'}\cdot\frac{1}{N}.
\end{aligned}$$

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 proves that 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:

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

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

When we ideally stipulate that the first block mined will be the final one, 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., $\gamma_2<\gamma_1$.

Security Analysis for Case A

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

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

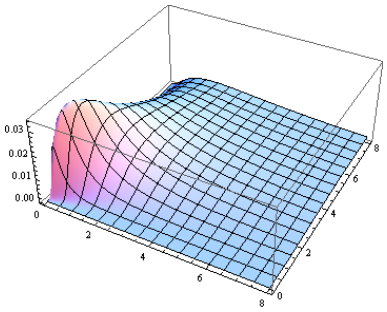

Fig. 6. Log-Normal Distribution

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

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:

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

where

$$\begin{aligned}\mathbb{E}[\sqrt{xy}] &= \iint_{D=\mathbb{R}^+\times\mathbb{R}^+} \sqrt{xy} \cdot p_x(x|y) \cdot p_y(y) \cdot \mathrm{d}\sigma \\ &= \mu \cdot e^{-\sigma_1^2/8},\end{aligned}$$

more details for this step is shown in Fig. B.3, 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 can be guaranteed.

The ultimate elimination of centralized mining pools

Briefly, we introduce one methodology that may work to eliminate centralized mining pools. Our proposed method is to have the system itself take mining pools' job. Some easier puzzle can be set similarly to our construction of the generalized PoW, and anyone who submits any solution can be rewarded. This can be done after modifications over our fork-free hybrid consensus.

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

For the last term,

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

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

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

Fig. 7. Detailed deductions on $\mathbb{E}[\sqrt{xy}]$ for case A