ASHWAChain: A Fast, Scalable and Strategy-proof Committee-based Blockchain Protocol

Sanidhay Arora[0000−0002−5821−8294], Anurag Jain[0000−0002−4637−4708], Sankarshan Damle[0000−0003−1460−6102], and Sujit Gujar[0000−0003−4634−7862]

Machine Learning Lab,
International Institute of Information Technology, Hyderabad, India
sanidhay.arora@students.iiit.ac.in, {anurag.jain, sankarshan.damle}@research.iiit.ac.in
sujit.gujar@iiit.ac.in

Abstract. Most cryptocurrencies are practically limited, mostly because of their significant time to finality and lack of scalability. Moreover, most of the existing literature for blockchain consensus protocols assumes the miners as honest. The assumption results in the protocols being susceptible to strategic attacks such as selfish mining, undercutting. Consequently, designing scalable, strategy-proof blockchain consensus protocol forms the basis of this work. Towards this, we present ASHWAChain, which deploys committees, generated through an underlying Proof-of-Work blockchain, to reach consensus using PBFT. Through a sophisticated analysis of system performance, we show that ASHWAChain's performance is significantly better than the current state-of-the-art. Additionally, we analyze miners' strategic behavior in ASHWAChain and prove that at equilibrium, the miners will honestly follow the protocol under certain assumptions.

Keywords: Distributed Ledgers · Scalable Blockchains · Peer-to-Peer Networks · Game Theory

1 Introduction

Bitcoin [17] promised to transform the financial system by proposing a decentralized peer-to-peer currency. *Miners* maintain this system by honestly following the underlying *consensus protocol* through appropriate incentives/rewards. In Bitcoin and similar *cryptocurrencies* such as *Ethereum*, every miner validates transactions and tries to append a set of valid transactions to the set of validated transactions, i.e., append a block on the blockchain. These miners compete against one another in a "mining" game. Typically, these miners have to produce Proof-of-Work (PoW) to write new transaction data to the blockchain.

As of October 2020, Bitcoin's and Ethereum's network processes an average of 3-4 and 10 transactions per second (TPS), respectively [1,3]. In contrast, Visa's global payment system handles a reported 1,700 TPS and claims to be capable of handling more than 24,000 TPS [23]. Besides, using such cryptocurrencies for micro-payments, like groceries, coffee, etc., is not feasible as the protocols only provide *eventual consistency*. It means that each transaction requires a certain number of block confirmations for it to be confirmed, i.e., accepted as irreversible, with high probability. For instance, Bitcoin takes at-least 60 minutes to confirm a transaction[1]. Thus, low transaction throughput and only eventual consistency hinder the widespread acceptance of cryptocurrencies.

Note that the miners involved in maintaining the transaction data could be *strategic players*. That is, they may deviate from the prescribed protocol to gain additional *rewards*. Selfish mining, petty mining, undercutting are some of the strategies that may lead to greater rewards for the miners [7,19]. Designing blockchain protocols robust to such strategic deviations is a challenge and a new area of research with limited prior work [21]. In view of this, it may not be a strategic miner's *best response* to follow the protocol honestly. In summary, resolving these practical and game-theoretic limitations to design a decentralized, strategy-proof, and scalable blockchain consensus protocol is a considerable challenge.

Researchers have proposed several protocols to improve blockchain technology's practical performance for better applicability [9, 15, 20, 28]. These protocols provide 3-5 times improvement over Bitcoin in terms of TPS [20]. Chen et al. [10] use game-theory to establish the trade-off between full verification, scalability, and finality-duration. However, they do not consider network delays in their model, and hence, their bounds are quite optimistic. Even with this, it is impossible to have a scalable and consistent, fully decentralized blockchain.

[1] It is also referred to as a time to finality.

With this impossibility as a backdrop, in this paper, we propose a blockchain consensus protocol, namely ASHWAChain[2], which trade-offs some of the decentralization for improved consistency and scalability; thus, increasing its practical applicability. Specifically, we propose to select a *committee* of identities controlled by the miners who mine the blocks for some assigned duration. We refer to such duration as an *epoch*. We elect the committee for every epoch.

Such a *committee-based blockchain* is similar to EOS [2] and DIFINITY [22]. Unlike these protocols, in ASHWAChain, we present a more efficient way to select the committee. Additionally, we also consider the strategic behavior of miners. Specifically, ASHWAChain uses PoW as a method to guarantee Sybil-resistance. The guarantee holds under the standard assumption that no miner will have the majority of the computational resource. In ASHWAChain, after solving a PoW puzzle, a miner gains an *identity* in the committee. We call the identities in this committee as *validators* that run a Byzantine agreement protocol, PBFT [8], to reach consensus.

First, we provide security and scalability analysis of ASHWAChain. We show that ASHWAChain can achieve a throughput of at-least 300 TPS (Section 3.2). Thus outperforming the cryptocurrencies mentioned above. Next, to game-theoretically analyze miners' behavior at equilibrium in ASHWAChain, we consider three types of miners: *honest*, *rational*, and *Byzantine*. We prove that it is Bayesian Nash Equilibrium for rational players to validate and sign the blocks (Theorem 1). That is, ASHWAChain is robust to the strategic behavior of miners. In summary, ASHWAChain provides strong consistency (fast confirmations) and high scalability while maintaining the system's security.

Related Work. Committee-based blockchains like EOS [2] claim to scale up-to 5000 TPS but still have open security concerns. While DFINITY [22] can handle up-to 40 TPS, it is significantly less for a financial transaction platform. Our work is most similar to PeerCensus [6], where identities in the network gain privileges to produce and validate blocks by solving a PoW puzzle. However, as the size of the committee is ever-growing, PeerCensus becomes infeasible to scale. Garay et al. [12] showed that deferring the resolution of blockchain forks is inefficient and strong consistency provides more applicability. Most of these protocols, however, assume participants as either honest or Byzantine, failing to explore the effect of rational participants thoroughly.

Solidus [14] is the first to consider rational participants by proposing an incentive-compatible Byzantine-Fault-Tolerant protocol for blockchain. However, it does not provide a game-theoretic analysis. Biais et al. [5] model Bitcoin as a coordination game but only considering rational participants. Yackolley et al. in [26] provide a game-theoretic analysis in consensus-based blockchains considering rational and Byzantine players and model a dynamic game. Manshaei et al. in [16] consider a non-cooperative static game approach for an intra-committee protocol where they show that rational players can free-ride if rewards are equally shared. Contrarily to previous studies, the consideration of the costs of block validation in the player's utilities makes our study more realistic. We follow the BAR model [4] and study the rational behavior in a non-cooperative setting, as shown by Halpern et al. in [13].

2 ASHWAChain: Protocol

In this section, we present our committee-based blockchain protocol, namely ASHWAChain. We first define the underlying network in it.

Network Model. We consider a peer-to-peer network consisting of miners who control identities in the network. These identities are denoted by their public-private (pk, sk) key pair. Miners are connected by a broadcast network over which they can send messages to everyone.

ASHWAChain comprises of two layers, (i) the PoW Blockchain layer (PBL) and (ii) the Agreement layer (AL). PBL is responsible for providing Sybil-resistant identities, and AL is responsible for maintaining consensus, i.e., fork-resolution and transaction confirmation.

2.1 Proof-of-Work Blockchain Layer (PBL)

The key insight behind PoW mechanisms is that the computational resources needed to solve cryptographic puzzles are not easily acquired and may not be scaled at will. In ASHWAChain, we leverage the same to provide Sybil-resistant identities. PBL consists of a blockchain running a PoW protocol, similar to Bitcoin. Each block at height i, i.e., b_i, is of the form $b_i = \langle h, d, x, p \rangle$. Here $h = H(b_{i-1})$ is the hash of previous block, d is the difficulty of mining, x is the nonce, and p is the identity controlled by a miner that will join the committee to become a *validator*.

[2] *Ashwa* refers to a horse which one can ride with high speed without much hassle [24].

Once a miner mines a block b, PBL provides a `proposeBlock`(b) operation to interact with the Agreement Layer (AL). Specifically, it proposes b to AL. Note that if more than one valid block is proposed at the same time, then only one of them is accepted. Once b is accepted in AL, a `commitBlock`(b) event is triggered in PBL for `proposeBlock`(b). Post this, the identity p in b joins the AL. All the miners in PBL acknowledge the addition of this block. The updated chain is then considered to mine further blocks, i.e., each miner stops mining the current block, chooses a different identity in the block if their own block was accepted, and then continues to mine a new block.

2.2 Agreement Layer (AL)

AL is maintained by a committee of identities. We refer to the identities present in the committee as *validators*. AL comprises a shared state which consists of PowChain, T, ComChain, TxChain, O and B, which are defined later in this subsection and summarised in the Appendix (Table A.1). This shared state is maintained by the committee of validators running SGMP [18] and PBFT [8] agreement protocols, similar to PeerCensus [6]. The shared state can only be modified by three pre-determined operations: `powBlock`(b), `txBlock`(t) and `comBlock`(c) which are defined later in Algorithm 1. For easy reference, summary is provided in Appendix (Table A.2).

AL proceeds in epochs where each epoch consists of t_{epoch} rounds. In ASHWAChain, a round is said to be complete after each commit of the `powBlock`(b) operation (refer Algorithm 1). The committee is *updated* after each epoch. The committee is of a fixed number of identities, $n_C = advRate \times t_{epoch}$, where $advRate$ is a system parameter that determines the size of the committee based on epoch size.

The state ComChain stores a blockchain with each block c containing the list of validators. This state is updated after each epoch with `comBlock`(c), as defined in Algorithm 1. The identities present in the latest block of ComChain are considered as validators for the current epoch. These identities are the same as those present in the latest n_C blocks in the PowChain state, which stores the PoW blockchain from PBL, and is updated with `powBlock`(b) operation (Algorithm 1).

A primary validator is selected to run the agreement protocol, formally presented in Appendix (Algorithm 2). In PowChain, for chain length l and blocks $(b_1, \ldots, b_{l-t_{epoch}+1}, \ldots, b_l)$, the identity in block $b_{l-t_{epoch}+1}$ is considered the *primary* and in case of Byzantine behaviour, next block in the PowChain will used to select the primary validator, using similar procedures as in PBFT. When the miners proceed to the next round after b is committed in PowChain, the next validator in the committee is considered as primary.

The agreement procedure for an identity p in the committee is explained as follows, formally presented in Appendix (Algorithm 2). The Primary validator creates the necessary operation and broadcasts it to the committee. The operation is then validated according to its pre-defined rule in Algorithm 1. Validators wait to collect all the messages for a certain time according to network latency. Once the operation is signed by a super-majority [3] of validators, they append this operation to operation log O. Post this, they commit the operation according to its commit rule as defined in Algorithm 1.

Algorithm 1 defines all the operations of the AL. Validation of any operation involves validating all the signatures, transactions, and the block structure. The `powBlock`(b) operation is broadcasted by the Primary validator after receiving the `proposeBlock`(b) request from a miner in the PBL layer. Then, b is appended to the PowChain once this operation is committed. In `txBlock`(t) operation, t is the block of transactions and each transaction is of the form $tx = \langle from, to, sign, coins, txFee \rangle$, where $from$ and to are the addresses of sender and recipient, $sign$ is the signature of the sender, $coins$ is the number of coins, and $txFee$ is the transaction fee. Once this operation is committed, all the transactions are applied accordingly by updating the state B, which stores the Account Balances. Once all the transactions are applied, the state T, which stores the list of validators (who signed the transactions for each tx) is updated. Note that v is the validator address in Algorithm 1. Post this, Block Reward BR, and the total transaction fee is distributed equally among the validators who signed the respective transactions. Finally, the state TxChain, which stores the transaction blockchain, is updated by appending t to it. In summary, Figure 1 provides the schematic representation of ASHWAChain.

3 ASHWAChain: Analysis

We now analyze the security and game-theoretic aspects of ASHWAChain. For this, we begin by defining our adversary and player models.

[3] $\frac{2}{3}^{rd}$ of total validators + 1 as we deploy PBFT.

Algorithm 1: Operations

```
Validate powBlock(b): begin
    $b^* \leftarrow$ latest block in $PowChain$
    if $b$ is child of $b^*$ and $b$ is valid then
        return Valid
    else
        return Invalid

Validate comBlock(c): begin
    if All $t_{epoch}$ identities in $c$ match with the PowChain and $c$ is valid then
        return Valid
    else
        return Invalid

Validate txBlock(t): begin
    if signatures and all txs are valid then
        return Valid
    else
        return Invalid

Commit powBlock(b): begin
    Append powBlock(b) in $O$
    Append $b$ to PowChain

Commit comBlock(c): begin
    Append comBlock(c) in $O$
    Append $c$ to ComChain

Commit txBlock(t): begin
    Append txBlock(t) to $O$
    for $tx \in t$ do
        $B(from) \leftarrow B(from) - coins$
        $B(to) \leftarrow B(to) + coins$
    Append the list of validators of $t$ in $T$
    for $v \in T(t)$; $nSig \leftarrow$ Total signatures do
        $B(v) \leftarrow B(v) + \frac{\sum txFee + \text{Block Reward}}{nSig}$
    Append $t$ to TxChain
```

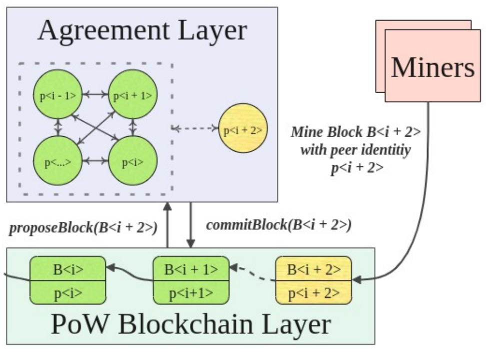

Fig. 1: ASHWAChain overview at height i

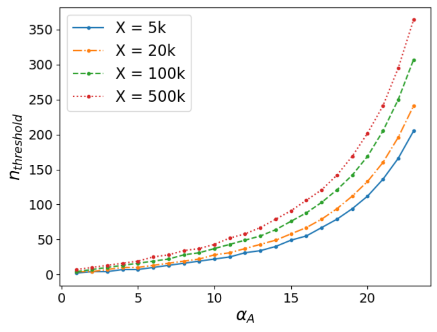

Fig. 2: For a given α_A, required $n_{threshold}$ such that the probability of the system not being in secure state is $1/X$

Adversary Model. We consider an adversary A controlling α_A fraction of computational resources in the network. The remaining resources are controlled by a meta entity H. The goal of the adversary is to control more than n_{ag} identities, i.e., the minimum number of identities needed for agreement. As we deploy PBFT to reach consensus, we have $n_{ag} \geq \frac{2}{3}n_C + 1$. Moreover, the probability of an adversary joining the committee will be directly proportional to its computation power. Similar to [6], we analyze ASHWAChain in its *steady state*, i.e., the number of validators and computational resources are governed by their respective expected value.

Player Model. In ASHWAChain, we consider the miners to be strategic. Towards analyzing their behavior at equilibrium, we consider three *types* of players:

$\mathcal{H}$ **Honest.** These players do not deviate from the defined protocol.

$\mathcal{R}$ **Rational.** These players follow the protocol if it yields the highest utility but may switch to alternative strategies, such as double spending, if they yield a higher utility.

$\mathcal{B}$ **Byzantine.** These players actively try to compromise the system, by trying to double spend etc., irrespective of the utilities they obtain.

As stated, the reward structure of any blockchain consensus protocol induces a game among the participants. To analyze this we define the following notations: Consider a set of miners $\mathcal{M} = \{1, \ldots, m\}$. Let each miner i's strategy be s_i and its type be τ_i, where $\tau_i \in \{\mathcal{H}, \mathcal{R}, \mathcal{B}\}$. Note that, the strategy will depend on the miner's type. We have $\mathcal{S} = \{s_1, \ldots, s_m\}$ as the strategy vector and $\mathcal{S}_{-i}$ as the vector without player i. Let r_i denote miner i's reward. With this, $u_i(\mathcal{S}, \tau_i, r_i)$ represents a miner i's utility from its participation. In ASHWAChain, we game-theoretically analyze the player's behavior at equilibrium using the following notion.

Definition 1 (Bayesian Nash Equilibrium (BNE)). *A strategy vector $\mathcal{S}^* = \{s_1^*, \ldots, s_m^*\}$ is said to be a Bayesian Nash Equilibrium (BNE) if for every player i, it maximizes its expected utility $u_i(\mathcal{S}^*, \tau_i, r_i)$ i.e., $\forall i \in \mathcal{M},\ \forall j$,*

$$\mathbb{E}_{\mathcal{S}_{-i}}\left[u_i(\mathcal{S}^*, \tau_i, r_i)\right] \geq \mathbb{E}_{\mathcal{S}_{-i}}\left[u_i(s_i, \mathcal{S}_{-i}^*, \tau_i, r_i)\right]\ ;\ \forall s_i. \tag{1}$$

Intuitively, BNE states that on average it is the best response for a player to follow $\mathcal{S}^*$, assuming every other player is following it.

3.1 Security Analysis

We first analyze the security aspects of ASHWAChain. Our analysis relies on the standard assumption of PBFT that the system will only work correctly if the number of adversary identities in the committee is less then one third of the committee size. To capture this formally, we define the following.

Definition 2 (Secure State). *The system is said to be in a secure state if n_A, i.e., the number of validators controlled by an adversary, is strictly less than $\frac{1}{3} \cdot n_C$.*

Proposition 1. *The system will be in a secure state with high probability depending on $n_{threshold}$ and β if (i) α_A, the fraction of computational resources in the network controlled by the adversary A is upper bounded by a fraction β and (ii) $n_C \geq n_{threshold}$, where $n_{threshold} \in \mathbb{Z}^+$.*

Proposition 1 follows with Definition 2. We provide the formal proof in Appendix (Section A.1).

Selecting n_C. Observe that the probability of a miner being added to PBL is directly proportional to α_{miner}, i.e., the computational power it holds in the system. Consequently, we can model the probability distribution of the identities that will be added to PBL as a *binomial distribution* [25]. We know that, from Proposition 1, the system will not be in a secure state if the number of validators, n_A, controlled by the adversary are $\geq \frac{1}{3} \cdot n_C$. Thus, the probability of the system not being in a secure state becomes the binomial distribution of n_A successes in n_C independent experiments, with α_A as the probability of a each success[4]. With this observation, Figure 2 shows the graph between $n_{threshold}$, i.e., the minimum number of identities needed to ensure secure state with respect to α_A for different probability guarantees. More concretely, Figure 2 presents the threshold of the committee size required, for a given α_A, such that the probability with which the system is *not* in a secure state is $1/X$, where $X \in \mathbb{Z}^+$.

3.2 Scalability Analysis

ASHWAChain runs PBFT agreement protocol which has $O(n^2)$ time complexity where n is the number of identities. Thus, the transaction throughput in our protocol will depend on n_C. Yaqin et al. [27] show that PBFT can scale up-to 200 TPS for 100 identities, where each transaction's size is 50 bytes. We remark that ASHWAChain provides strong consistency, i.e., a transaction is considered as irreversible once it is accepted. This is in contrast to other cryptocurrencies running PoW and PoS protocols like Bitcoin, Ethereum, etc., which need multiple block confirmations for block finality. As a result, the PBFT agreement protocol is the computational bottleneck in ASHWAChain. Note that, the aforementioned cryptocurrencies handle around 20 TPS. Committee-based blockchains like BitShares and DFINITY handle around 18 and 40 TPS respectively. IOTA [11] uses a DAG-based protocol and handles up-to 50 TPS.

From our security analysis, if we consider $\alpha_A = 0.18$ fraction of the computational power controlled by the adversary, then for $n_C = 90$, the probability of the system being in a secure state will be greater than 1.44×10^{-4} (Figure 2). For this n_C, ASHWAChain can support at-least 300 TPS, which is a significant improvement over existing cryptocurrency protocols.

3.3 Equilibrium Analysis

Towards this, we assume that all rational players incur a cost $\kappa > 0$ if any malicious transaction block is committed, similar to [26]. This assumption is based on the premise that the entire ecosystem of the currency inflicts harm when an invalid block is accepted and, since rational players have invested resources and possess a stake in the system, they must incur some cost depending on it. We also assume that the objective of Byzantine players is to minimize the utility of the rational players and prevent the protocol from achieving its goal, regardless of the cost they incur. With these assumptions, we

Term	Definition
N	Set of identities, $\{1, 2, \ldots, n_C\}$
τ_i	Player i's type, i.e.,$\tau_i \in \{\mathcal{H}, \mathcal{R}, \mathcal{B}\}$
u_i	Utility of player i; $u_i : \tau_i \times TR \times S \to \mathbb{R}$
s_i^1	Player i signs the transaction block without validating
s_i^2	Player i signs the transaction block only if its valid
s_i^3	Player i signs and proposes only invalid blocks
S_i	Set of pure strategies of players i, $\{s_i^1, s_i^2, s_i^3\}$

Table 1: Game constituents

Term	Definition
c_{mine}	The cost of mining a block
c_{val}	The cost to validate a tx
$txFee$	Average transaction fee
κ	Cost incurred by rational players if invalid block is accepted
ϕ	Number of transactions in each block
ψ	Number of identities signing a block
BR	Transaction Block reward
$P_{invalid}$	Belief probability of invalid block being accepted, $P_{invalid} : N^{n_C} \times S_i \to [0, 1]$
TR	Total reward gained, $\frac{\phi \cdot txFee + BR}{\psi}$

Table 2: Relevant Notations

model a game between honest, rational and Byzantine players' identities in the committee, defined as: $\Gamma = \langle N, (\tau_i), (S_i), (u_i) \rangle$. Table 1 and Table 2 comprises the notations used in our analysis.

Note that $P_{invalid}(N, s_i^*)$ is player i's belief that an invalid block is accepted after it follows s_i^*. Honest and Byzantine players will always follow s_i^2 and s_i^3, respectively. Hence we only consider rational players' behavior and therefore their equilibrium strategies. The expected utilities of a rational player i for one round are as follows:

$$u_i(\cdot, TR, s_i^1) = TR - \frac{c_{mine}}{(n_C - t_{epoch})} - P_{invalid}(N, s_i^1) \cdot \kappa \tag{2}$$

$$u_i(\cdot, TR, s_i^2) = TR - \frac{\phi \cdot c_{val}}{\psi} - \frac{c_{mine}}{(n_C - t_{epoch})} - P_{invalid}(N, s_i^2) \cdot \kappa \tag{3}$$

In a committee of n_C identities, let n_H, n_R, and n_B be number of honest, rational and Byzantine players' identities respectively. Trivially, $n_C = n_H + n_R + n_B$. We use BNE for analysis, as we have a static game with asymmetric information. As stated, the minimum number of identities needed for agreement are n_{ag}. As we use PBFT, we have, $n_{ag} = \frac{2}{3} n_C + 1$. Our analysis relies on the standard assumption that the system will only work correctly if $n_H + n_R \geq n_{ag}$, i.e., $n_B < n_C - n_{ag}$.

Claim 1. *For every rational player's identity i, its belief regarding the probability of an invalid block being accepted will be more if it signs a block without validating, as compared to when it validates and then signs, i.e.,* $\delta = P_{invalid}(N, s_i^1) > P_{invalid}(N, s_i^2)$ *s.t.* $\delta > 0$.

Intuitively, Claim 1 follows from the fact that the chance of an invalid block being accepted is more if a rational player decides not to validate a block before signing the block. This is because the size of the committee, N, is finite. We present the formal proof in the Appendix (Section A.2).

Theorem 1. *In ASHWAChain, if (i)* $n_B < n_C - n_{ag}$ *and (ii)* $\phi \leq \frac{\kappa \cdot \psi \delta}{c_{val}}$, *then for every rational player i,* $s_i^* = s_i^2$ *is a BNE. Here,* $\delta = P_{invalid}(N, s_i^1) - P_{invalid}(N, s_i^2)$ *s.t.* $\delta > 0$.

Theorem 1 follows from Claim 1 and by observing that (3) is greater than (2) when $\phi \leq \frac{\kappa \cdot \psi \delta}{c_{val}}$. The formal proof is provided in the Appendix (Section A.3). However, in a special case where $n_H \geq n_{ag}$, $S^* = (s_i^1)$ is the BNE for rational players where only valid blocks are accepted.

4 Conclusion

In this work, we introduced a blockchain consensus protocol, namely ASHWAChain that provides strong consistency and high scalability, by using a committee-based system for consensus (Section 2). Our security analysis shows that the system is in a secure state (Section 3.2), with high probability (Figure 2). Lastly, we proved that under certain assumptions on the protocol parameters, there exists a Bayesian Nash Equilibrium in which the rational players validate the transactions before signing any block (Theorem 1).

[4] For additional details, we refer the reader to Section A.1

References

1. Blockchain Charts (2020), https://www.blockchain.com/charts/transactions-per-second
2. Documentation/TechnicalWhitepaper.md (2020), https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md
3. Ethereum Daily Transactions Chart | Etherscan (2020), https://etherscan.io/chart/tx
4. Amitanand S. Aiyer, Lorenzo Alvisi, A.C.M.D.J.M., Porth, C.: Bar fault tolerance for cooperative services. In: Proceedings of the 20th ACM Symposium on Operating Systems Principles 2005, SOSP 2005, Brighton, UK, October 23-26, 2005. p. 45–58 (2005)
5. Bruno Biais, Christophe Bisière, M.B., Casamatta, C.: The blockchain folk theorem. In: The Review of Financial Studies (2019) (2019)
6. C. Decker, J.S., Wattenhofer, R.: Bitcoin meets strong consistency. In: In Proceedings of the 17th International Conference on Distributed Computing and Networking Conference (ICDCN). ICDCN (2016), https://tik-db.ee.ethz.ch/file/ed3e5da74fbca5584920e434d9976a12/peercensus.pdf
7. Carlsten, M., Kalodner, H., Weinberg, S.M., Narayanan, A.: On the instability of bitcoin without the block reward. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 154–167 (2016)
8. Castro, M., Liskov, B.: Practical byzantine fault tolerance. In: Third Symposium on Operating Systems Design and Implementation, New Orleans, USA, February 1999. Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139 (1999)
9. Chen, J., Micali, S.: Algorand: A secure and efficient distributed ledger. Theoretical Computer Science **777**, 155 – 183 (2019). https://doi.org/https://doi.org/10.1016/j.tcs.2019.02.001, in memory of Maurice Nivat, a founding father of Theoretical Computer Science - Part I
10. Chen, L., Xu, L., Gao, Z., Kasichainula, K., Shi, W.: Nonlinear blockchain scalability: a game-theoretic perspective. arXiv preprint arXiv:2001.08231 (2020)
11. Divya, M., Biradar, N.: Iota-next generation block chain. International Journal Of Engineering And Computer Science **7**, 23823–23826 (04 2018). https://doi.org/10.18535/ijecs/v7i4.05
12. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015. pp. 281–310. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
13. Halpern, J.Y., Vilaça, X.: Rational consensus: Extended abstract. In: Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing. p. 137–146. PODC '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2933057.2933088
14. Ittai Abraham, Dahlia Malkhi, K.N.L.R., Spiegelman, A.: Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. In: CoRR abs/1612.02916v1 (2016) (2016)
15. Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: A provably secure proof-of-stake blockchain protocol. pp. 357–388 (07 2017). https://doi.org/10.1007/978-3-319-63688-7$_1$2
16. Manshaei, M.H., Jadliwala, M., Maiti, A., Fooladgar, M.: A game-theoretic analysis of shard-based permissionless blockchains. IEEE Access **6**, 78100–78112 (2018). https://doi.org/10.1109/ACCESS.2018.2884764
17. Nakomoto, S.: Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org/bitcoin.pdf
18. Reiter, M.K.: A secure group membership protocol. In: Transactions on Software Engineering (1996)
19. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in bitcoin. In: International Conference on Financial Cryptography and Data Security. pp. 515–532. Springer (2016)
20. Siddiqui, S., Gujar, S.: Quicksync: A quickly synchronizing pos-based blockchain protocol. CoRR **abs/2005.03564** (2020), https://arxiv.org/abs/2005.03564
21. Siddiqui, S., Vanahalli, G., Gujar, S.: Bitcoinf: Achieving fairness for bitcoin in transaction fee only model. In: Seghrouchni, A.E.F., Sukthankar, G., An, B., Yorke-Smith, N. (eds.) Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020. pp. 2008–2010. International Foundation for Autonomous Agents and Multiagent Systems (2020)
22. Timo Hanke, M.M., Williams, D.: Dfinity technology overview series consensus system. CoRR abs/1805.04548 (2018) (2018), https://arxiv.org/pdf/1805.04548.pdf
23. Visa Inc.: Visanet booklet, https://usa.visa.com/dam/VCOM/download/corporate/media/visanet-technology/visa-net-booklet.pdf
24. Wikipedia contributors: Ashva — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Ashva&oldid=988520218 (2020)
25. Wikipedia contributors: Binomial distribution — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Binomial_distribution&oldid=989183881 (2020)
26. Yackolley Amoussou-Guenou, Bruno Biais, M.P.B., Tucci-Piergiovanni, S.: Rational vs byzantine players in consensus-based blockchains. In: In Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020, IFAAMAS, 9 pages. IFAAMAS (2020), http://ifaamas.org/Proceedings/aamas2020/pdfs/p43.pdf
27. Yaqin Wu, Pengxin Song, F.W.: Hybrid consensus algorithm optimization: A mathematical method based on pos and pbft and its application in blockchain. Mathematical Problems in Engineering, vol. 2020, Article ID 7270624, 13 pages, 2020 **2020** (2020). https://doi.org/https://doi.org/10.1155/2020/7270624
28. Yu, H., Nikolić, I., Hou, R., Saxena, P.: Ohie: Blockchain scaling made simple. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 90–105. IEEE (2020). https://doi.org/10.1109/SP40000.2020.00008

Appendix

A Proofs

In this section, we restate and present formal proofs of the results presented in the main paper.

A.1 Proposition 1 Proof

Proposition. The system will be in a secure state with high probability depending on $n_{threshold}$ and β if (i) α_A, the fraction of computational resources in the network controlled by the adversary A is upper bounded by a fraction β and (ii) $n_C \geq n_{threshold}$, where $n_{threshold} \in \mathbb{Z}^+$.

Proof. Since we consider PBFT, Byzantine identities should be strictly less than one-third of total identities, i.e., $n_B < \frac{1}{3} \cdot n_C$, for the system to work correctly. We know that, the probability that a miner mines a block is directly proportional to its fraction of computational power in the system, i.e., $Prob$(Miner to mine a block) $\propto \alpha_{miner}$. This follows by observing that the size of the committee, i.e., n_C is finite. Let X be the discrete random variable denoting the number of validators controlled by the adversary and consider the following binomial distribution to derive the probability of the number of validators controlled by the adversary A to be,

$$f(k, n_C, \alpha_A) = Pr(k; n_C, \alpha_A) = Pr(X = k) = \tbinom{n_C}{k}(\alpha_A)^k(1-\alpha_A)^{(n_C-k)}$$

Now consider the following Cumulative Distribution Function (CDF),

$$F(x) = Pr(X \leq x).$$

Note that, $F(\frac{1}{3} \cdot n_C)$ gives the probability that the system is in a secure state, i.e. $n_A < \frac{1}{3} \cdot n_C$, which depends on the value of n_C and α_A from the binary distribution. This proves the proposition. □

A.2 Claim 1 Proof

Claim. *In ASHWAChain, for every rational player's identity i, its belief regarding the probability of an invalid block being accepted will be more if it signs a block without validating, as compared to when it validates and then signs, i.e.,* $\delta = P_{invalid}(N, s_i^1) > P_{invalid}(N, s_i^2)$ *s.t.* $\delta > 0$.

Proof. Without loss of generality, for the proof we consider a rational player i. Let $P_{invalid}(N, s_i^1) = k_1$ and $P_{invalid}(N, s_i^2) = k_2$ s.t. $0 < k_1, k_2 < 1$. In the event when player i plays the strategy s_i^1, i.e., signs the block without validating, its belief regarding an invalid block being accepted can only increase. This follows by observing that the size of the committee, i.e., n_C is finite. Thus, player i not validating a block will directly imply that the chance of an invalid block being accepted will be more, i.e., $k_1 > k_2$. Now,

$$\begin{aligned} P_{invalid}(N, s_i^1) - P_{invalid}(N, s_i^2) &= k_1 - k_2 \\ &= \delta > 0. \end{aligned}$$

This proves the claim. □

A.3 Theorem 1 Proof

Theorem. *In ASHWAChain, if*
(i) $n_B < n_C - n_{ag}$ *and (ii)* $\phi \leq \frac{\kappa \cdot \psi\delta}{c_{val}}$,
then for every rational player i, such that $s_i^* = s_i^2$ *is BNE. Here,* $\delta = P_{invalid}(N, s_i^1) > P_{invalid}(N, s_i^2)$ *s.t.* $\delta > 0$.

Proof. The condition $n_B < n_C - nag$ should be true for PBFT to work correctly. For $s*_i = s_i^2$ to be a BNE for every rational player i, $u_i(R, s_i^1) \leq u_i(R, s_i^2)$ should satisfy according to Equation 1. On solving this condition using equations 2 and 3 we get,

$$\kappa \cdot (P_{invalid}(N, s_i^1) - P_{invalid}(N, s_i^2)) \geq \frac{\phi \cdot c_{val}}{\psi}.$$

Substituting $(P_{invalid}(N, s_i^1) - P_{invalid}(N, s_i^2))$ as δ, where $\delta \in R^+$ following from Claim 1, we get our result for the value of ϕ as:

$$\phi \leq \frac{\kappa \cdot \psi\delta}{c_{val}}. \quad \square$$

Name	State
PowChain	Proof of Work Blockchain
T	List of transaction validators
ComChain	Committee Blockchain
TxChain	Transaction Blockchain
O	Operation Log
B	Account Balances

Table A.1: Shared state constituents

Name	Operation
`powBlock`(b)	Adds PoW block b to PowChain
`txBlock`(t)	Adds transaction block t to T
`comBlock`(c)	Adds committee block c to ComChain

Table A.2: Shared state operations

B Notations

For readability, Table A.1 and Table A.2 summarize some of the notations presented in the main paper.

C Algorithms

In this section, we formally present the algorithms used in the main paper.

Algorithm 2: Agreement for identity p

```
begin
    if $p$ = Primary then
        Creates any operation and broadcasts
    else
        Validate operation, sign and broadcast
    wait($LatencyTime$)
    if signed by super-majority then
        Append operation to $O$
        Commit operation according to its commit rule
    else
        Nothing is committed, RESET
```