Brief Announcement: Best-Possible Unpredictable Proof-Of-Stake

Lei Fan
Shanghai Jiao Tong University, China

Jonathan Katz
Google, Washington DC, USA
University of Maryland, College Park, MD, USA

Zhenghao Lu
Shanghai Jiao Tong University, China

Phuc Thai
Sky Mavis, Ho Chi Minh City, Vietnam

Hong-Sheng Zhou
Virginia Commonwealth University, Richmond, VA, USA

Abstract

The proof-of-stake (PoS) protocols aim to reduce the unnecessary computing power waste seen in Bitcoin. Various practical and provably secure designs have been proposed, like Ouroboros Praos (Eurocrypt 2018) and Snow White (FC 2019). However, the essential security property of unpredictability in these protocols remains insufficiently explored. This paper delves into this property in the cryptographic setting to achieve the "best possible" unpredictability for PoS.

We first present an impossibility result for *all* PoS protocols under the *single-extension* design framework, where each honest player extends one chain per round. The state-of-the-art permissionless PoS protocols (e.g., Praos, Snow White, and more), are all under this single-extension framework. Our impossibility result states that, if a single-extension PoS protocol achieves the best possible unpredictability, then this protocol cannot be proven secure unless more than 73% of stake is honest. To overcome this impossibility, we introduce a new design framework called *multi-extension* PoS, allowing each honest player to extend *multiple* chains using a greedy strategy in a round. This strategy allows us to construct a class of PoS protocols that achieve the best possible unpredictability. It is noteworthy that these protocols can be proven secure, assuming a much smaller fraction (e.g., 57%) of stake to be honest.

2012 ACM Subject Classification Computing methodologies → Distributed computing methodologies

Keywords and phrases blockchain, consensus, proof-of-stake, unpredictability

Digital Object Identifier 10.4230/LIPIcs.DISC.2024.45

Related Version *Full Version*: `https://eprint.iacr.org/2021/660`

Funding Phuc Thai and Hong-Sheng Zhou were supported in part by NSF grant CNS-1801470.

Acknowledgements This project was conducted during Phuc Thai's time as a PhD student at Virginia Commonwealth University.

1 Introduction

Cryptocurrencies like Bitcoin [13] have proven to be a phenomenal success. These protocols are executed by a **large-size** peer-to-peer network of nodes using the proof-of-work (PoW) mechanism [9, 2]. They provide a trustworthy, append-only, and always-available public ledger, facilitating the implementation of a global payment system (e.g., Bitcoin) or a global computer (e.g., Ethereum). However, the PoW-based consensus requires substantial

computing power. Utilizing alternative resources like *coins (also known as stake)* to secure a blockchain is desirable. If successful, the new system would be environmentally friendly, as it would not rely on extensive computing power for security. Several attempts have been made, with PoS mechanisms widely discussed in the cryptocurrency community (e.g., [1, 12, 15, 4]). In a PoS-based blockchain protocol, players must prove ownership of a specified number of stakes; only those who can provide such proofs are permitted to participate in maintaining the blockchain. Compared with PoW mechanisms, the computational cost of finding solutions in PoS mechanisms is very "cheap."

Early PoS designs (e.g., [1, 12, 15, 4]) and PoW-based designs, such as the original Bitcoin, were initially crafted in an *ad hoc* style. However, the contemporary trend leans towards a more rigorous approach where security concerns are precisely defined, and the designed protocols undergo mathematical analysis. Notable contributions include the work by Garay et al. [10] and Pass et al. [14], analyzing the PoW-based blockchain in Bitcoin within the *cryptographic setting*. The analysis demonstrated that the Bitcoin blockchain can achieve crucial security properties, such as common prefix, chain quality, and chain growth. Indeed, research efforts have also been devoted to PoS-based and Bitcoin-like consensus, as seen in [8, 7, 3]. Nevertheless, these protocols are vulnerable to attacks due to predictability.

Intuitively, predictability in a protocol implies that certain players are aware they will be selected to generate blockchain blocks before actually doing so. Brown-Cohen et al. [5] explored the predictability of PoS in incentive-driven scenarios, where players may deviate from the protocol for higher profits. The power of predictability can be exploited by attackers to reduce the difficulty or cost of incentive-driven attacks like selfish-mining [5] or bribery [3]. Therefore, it is crucial for a PoS protocol to minimize predictability and mitigate the risks of these attacks. Ideally, a PoS protocol should aim for the *best possible* unpredictability, enabling effective counteraction of predictability-based attacks. Achieving this goal ensures the maintenance of blockchain fairness and incentivizes honest players to participate in the protocol.

Our first result is that we formally define (the best possible) unpredictability in the cryptographic setting. We assert that a protocol achieves the best possible unpredictability if it only allows players to predict whether they can generate the next block, and nothing more. Based on the definition of the best possible unpredictability, we identify an interesting impossibility for a class of PoS protocols following a *single-extension* design framework. Existing provably secure Bitcoin-like PoS protocols (e.g., [8, 7, 3]) are all within the single-extension framework. Finally, to overcome the impossibility, we develop a novel D-distance-greedy strategy in the multi-extension framework, which allows us to design a provably secure Bitcoin-like PoS protocol.

2 Security Model

The security of Bitcoin-like PoW-based protocols has been rigorously investigated by Garay et al. [10] and then by Pass et al. [14] in the cryptographic setting.

The execution of a PoS blockchain protocol. Following Canetti's formulation[6], we present an abstract model for a PoS blockchain protocol Π in the hybrid world of the semi-synchronous network communication functionality, the random oracles, and certain initialization functionality, similarly drawn from [14].

We consider the execution of blockchain protocol Π that is directed by an environment $\mathcal{Z}(1^\kappa)$, where κ is a security parameter. A necessary condition in all common blockchain systems is that all players agree on the first, i.e., the *genesis block*, which consists of the

identities (e.g., public keys) and the stake distribution of the players. The environment $\mathcal{Z}$ can "manage" players through an adversary $\mathcal{A}$ that can dynamically corrupt honest players. In any round r, each PoS-player $P \in \mathcal{P}$, with a local state st, receives a message from $\mathcal{Z}$, and potentially receives messages from other players. Then, it executes the protocol, broadcasts a message to other players, and updates its local state. Note that the network is under the control of $\mathcal{A}$, meaning that $\mathcal{A}$ is responsible for delivering all messages sent by players. Let $\mathsf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}}$ be a random variable denoting the joint VIEW of all players in the above protocol execution; note that this joint view fully determines the execution. More details of the formulation can be found in the full version of our paper.

Block and blockchain basics. A *blockchain* $\mathcal{C}$ consists of a sequence of ℓ concatenated blocks $B_0\|B_1\|B_2\|\cdots\|B_\ell$, where $\ell \geq 0$ and B_0 is the genesis block. We use $\mathsf{len}(\mathcal{C})$ to denote *blockchain length*, i.e., the number of blocks in blockchain $\mathcal{C}$; and here $\mathsf{len}(\mathcal{C}) = \ell$. (Note that since all chains must consist of the genesis block, we do not count it as part of the chain's length.) We use sub blockchain (or subchain) for referring to a segment of a chain; here for example, $\mathcal{C}[j, m]$, with $j \geq 0$ and $m \leq \ell$ would refer to a sub blockchain $B_j\|\cdots\|B_m$. We use $\mathcal{C}[i]$ to denote the i-th block, B_i in blockchain $\mathcal{C}$; here i denotes the *block height* of B_i in chain $\mathcal{C}$. If blockchain $\mathcal{C}$ is a prefix of another blockchain $\mathcal{C}_1$, we write $\mathcal{C} \preceq \mathcal{C}_1$.

Chain growth, common prefix, and chain quality. Previously, several fundamental security properties for Bitcoin-like PoW-based blockchain protocols have been defined: *chain growth* [11], *common prefix* [10, 14], and *chain quality* [10]. Intuitively, the chain growth property states that the chains of honest players should grow linearly to the number of rounds. The common prefix property indicates the consistency of any two honest chains except the last κ blocks. The chain quality property, characterized by the parameter $\mu \in (0, 1)$, aims to indicate the ratio of contributions from honest players that are contained in a sufficiently long and continuous part of an honest chain, is at least μ.

Unpredictability. At a high level, predictability means that (certain) protocol players are aware that they will be selected to generate blocks of the blockchain, *before* they actually generate the blocks. We investigate the unpredictability in the cryptographic setting.

Consider a malicious player $P \in \mathcal{P}$ at round r. Let VIEW^r be the view of all players at round r, and $\mathcal{C}^r$ be the best (valid) chain of all players in VIEW^r. At round r, the adversary $\mathcal{A}$ attempts to predict if the (malicious) player P can extend the best chain at a future round r', where $r' > r$. Let $z_P^{r'} \in \{0, 1\}$ be a prediction: $z_P^{r'} = 1$ means that $\mathcal{A}$ predicts that player P can extend the best chain at round r'. Now we introduce another random variable $\bar{z}_P^{r'}$ to indicate if P indeed can extend the best chain at round r' (as the adversary predicated at an early-round r) or not. Let $\text{VIEW}^{r'}$ be the view of all players at round r', and $\mathcal{C}^{r'}$ be the best valid chain of all players in $\text{VIEW}^{r'}$. We set $\bar{z}_P^{r'} = 1$ if there exists a chain $\mathcal{C} = \mathcal{C}^{r'}\|B$ in VIEW with a block B generated by player P at round r', otherwise we set $\bar{z}_P^{r'} = 0$.

Consider a view VIEW, protocol round r, and a malicious player P. For a prediction $z_P^{r'}$ where $r' > r$, we define the predicate $\mathsf{predictable}$ to be true if the prediction $z_P^{r'}$ accurately predicts whether or not player P can generate a new chain at round r' that is 2 blocks longer than the longest chain at round r. (In any PoS protocol, all players can *always* predict whether or not they can generate the next block, so we consider 2 blocks.) More concretely, we define $\mathsf{predictable}(\text{VIEW}, P, r, r', z_P^{r'}) = 1$ if and only if the following three conditions hold: *(i)* $r' > r$; *(ii)* $\mathsf{len}(\mathcal{C}^{r'}) + 1 - \mathsf{len}(\mathcal{C}^r) = 2$; and *(iii)* $z_P^{r'} = \bar{z}_P^{r'}$.

▶ **Definition 1** (The best possible unpredictability). *Consider a blockchain protocol* Π. *We say protocol* Π *achieves the best possible unpredictability if for all* PPT $\mathcal{Z}, \mathcal{A}$, *for any malicious player* P *at any round* r, *we have,*

$$\Pr\left[\ \textit{VIEW} \leftarrow \mathsf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}}; (r', z_P^{r'}) \leftarrow \mathcal{A}(P, r, \textit{VIEW}^r)\ \middle|\ \mathsf{predictable}(\textit{VIEW}, P, r, r', z_P^{r'}) = 0\ \right] > 1 - \mathsf{neg}(\kappa),$$

where $\mathsf{neg}(\cdot)$ *is a negligible function.*

3 An Impossibility Result

In this section, we present an impossibility result for a class of PoS protocols in the *single-extension* PoS framework. Intuitively, for Bitcoin-like PoS protocol in the *single-extension* framework, in each round, each honest player identifies only a single "best chain," and then extends this chain. The formal definition of this framework is presented in the full version of our paper. We remark that the state-of-the-art PoS protocols (e.g., [7, 8, 3]) can be categorized as single-extension PoS protocols.

Then we present an impossibility result for single-extension PoS protocols. Concretely, consider a PoS protocol in the single-extension framework, we can show that, if the PoS protocol achieves the best possible unpredictability, then the protocol cannot maintain security properties, such as the common prefix, when honest players control less than 73% of the stake. Let N be the number of players and ρ be the fraction of malicious players in the protocol execution. Let p be the probability that a player can extend a chain in a round. The probability that honest players extend a chain in a round is $\alpha = 1 - (1-p)^{N\cdot(1-\rho)}$. Similarly, the probability that the adversary extends a given chain is $\beta = 1 - (1-p)^{N\cdot\rho} \approx \frac{\rho}{1-\rho} \cdot \alpha$, if p is sufficiently small. The impossibility theorem is stated as follows, and the proof can be found in the full version.

▶ **Theorem 2.** *Consider a single-extension PoS protocol* Π *achieves the best possible unpredictability. If* $\alpha < e \cdot \beta$, *where* $e = 2.72$, *then* Π *cannot achieve common prefix property.*

4 Greedy Strategies: How to overcome the impossibility

In the previous section, we have obtained the impossibility of single-extension PoS protocols. In this section, we will introduce greedy strategies that follow a *multi-extension* framework. In these strategies, honest players are allowed to extend multiple chains that are "close" to each other. Our protocol can achieve the best possible unpredictability while requiring a much smaller fraction (e.g., 57%) of honest stake to achieve security properties.

Specifically, we allow the players to take a *greedy* strategy to extend the chains in a protocol execution: instead of extending a single best chain (i.e., the longest chain), the players are allowed to extend *a set of best chains*, expecting to extend the best chain faster. This is possible because extending chains in a PoS protocol is "very cheap." We remark that the set of best chains should be carefully chosen; otherwise, the protocol may not be secure. In our greedy strategy, the honest player extends the set of chains that share the same common prefix after removing the last few blocks. With this strategy, the security of the protocol is guaranteed.

Distance-greedy strategies. Consider a protocol execution. In each player's local view, there are multiple chains, which can be viewed as a tree. Concretely, the genesis block is the root of the tree, and each path from the root to a node is essentially a chain. The tree will "grow": the length of each existing chain may increase, and new chains may be created,

round after round. First, we define the "distance" between two chains in a tree. Intuitively, we say the distance from a "branch" chain to a "reference" chain is d if we can obtain a prefix of the reference chain by removing the last d blocks of the branch chain.

▶ **Definition 3** (Distance between two chains). *Let $\mathcal{C}$ be a chain of length ℓ, and $\mathcal{C}_1$ be a chain of length ℓ_1. We view $\mathcal{C}$ as the "reference" chain, and $\mathcal{C}_1$ to be the "branch" chain. Next, we define the distance between $\mathcal{C}$ and $\mathcal{C}_1$, and we use* $\mathsf{distance}(\textit{branch chain} \to \textit{reference chain})$*, i.e.,* $\mathsf{distance}(\mathcal{C}_1 \to \mathcal{C})$ *to denote the distance. More formally, if d is the smallest non-negative integer so that $\mathcal{C}_1[0, \ell_1 - d] \preceq \mathcal{C}$, then we say the distance between the reference chain $\mathcal{C}$ and the branch $\mathcal{C}_1$ is d, and we write* $\mathsf{distance}(\mathcal{C}_1 \to \mathcal{C}) = d$.

Now we are ready to define the distance-greedy strategies. Intuitively, a player following a distance-greedy strategy will try to extend a *set of best chains*, where the distance between the best chain and the chains in this set is quite small. Here, we consider the best chain as the branch chain and all other chains in the set of best chains as the reference chains. By the definition of the distance, we can obtain a common prefix of all reference chains by removing the last few blocks of the branch chain. Formally, we have the following definition.

▶ **Definition 4** (D-distance-greedy strategy). *Consider a blockchain protocol execution. Let P be a player of the protocol execution, and let $\mathbb{C}$ be the set of chains in player P's local view. Let $\mathcal{C}_{\text{best}}$ be the longest chain at round r, where $\ell = \mathsf{len}(\mathcal{C}_{\text{best}})$. Let D be non-negative integers. Define a set of chains $\mathbb{C}_{\text{best}}$ as*

$$\mathbb{C}_{\text{best}} = \left\{\mathcal{C} \in \mathbb{C} \;\middle|\; \mathsf{distance}(\mathcal{C}_{\text{best}} \to \mathcal{C}) \leq D\right\}.$$

We say P is D-distance-greedy *if, for all r, P makes attempts to extend all chains $\mathcal{C} \in \mathbb{C}_{\text{best}}$.*

Our protocol. We present a new protocol $\Pi^\bullet$ to achieve the best possible unpredictability while only requiring a much smaller fraction (e.g., 57%) of honest stake to achieve the security properties. For simplicity, we consider the payloads in all blocks to be empty. Protocol $\Pi^\bullet$ uses a unique digital signature scheme and a hash function as building blocks.

In the blockchain initialization phase, the genesis block B_0 will be created. Given a group of PoS-players $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$, a security parameter κ, and a unique digital signature scheme $(\mathsf{uKeyGen}, \mathsf{uKeyVer}, \mathsf{uSign}, \mathsf{uVerify})$, the initialization is as follows: each $P_j \in \mathcal{P}$ generates $(\mathrm{SK}_j, \mathrm{PK}_j) \leftarrow \mathsf{uKeyGen}(1^\kappa)$, publishes PK_j and keeps SK_j secret. The public keys are stored in B_0. In addition, an independent randomness $\mathsf{rand} \in \{0,1\}^\kappa$ will also be stored in B_0. That is $B_0 = \langle(\mathrm{PK}_1, \mathrm{PK}_2, \cdots, \mathrm{PK}_n), \mathsf{rand}\rangle$. For simplicity, we assume the flat model and omit the stake distribution in the genesis block.

In the blockchain extension phase, our protocol is parameterized by $\mathsf{Context}^\bullet$, $\mathsf{Mining}^\bullet$, $\mathsf{Validate}^\bullet$, and D-$\mathsf{BestChainSet}^\bullet$. The algorithm $\mathsf{Validate}^\bullet$ takes a chain $\mathcal{C}$ (with length ℓ) and the current round number r as inputs and evaluates every block of $\mathcal{C}$. Starting from the head of $\mathcal{C}$, for every block $\mathcal{C}[i]$, where $i \in [\ell]$, the procedure $\mathsf{Validate}^\bullet$ verifies that 1) $\mathcal{C}[i]$ is linked to the previous block $\mathcal{C}[i-1]$ correctly, 2) the hash inequality is correct, and 3) the signature is correct. The algorithm D-$\mathsf{BestChainSet}^\bullet$ selects the best (longest) chain $\mathcal{C}_{\text{best}}$ and iterates through the set of chains in the local state to find all the chains in which the distances from $\mathcal{C}_{\text{best}}$ to those chains do not exceed D, and outputs the set of best chains $\mathbb{C}_{\text{best}}$. For a chain $\mathcal{C} = B_0 \| B_1 \| B_2 \| \ldots \| B_i$ in $\mathbb{C}_{\text{best}}$, some honest player P, with key pair $(\mathrm{SK}, \mathrm{PK})$, tries to extend $\mathcal{C}$ at round r as follows. First, P computes the context $\eta := \mathsf{Context}^\bullet(\mathcal{C})$. Here, algorithm $\mathsf{Context}^\bullet$ returns the hash value of the last block on $\mathcal{C}$, i.e., $\mathsf{Context}^\bullet(\mathcal{C}) = h(B_i)$. Then, P tries to obtain a new block using the $\mathsf{Mining}^\bullet$ algorithm. Concretely, a new block

Algorithm 1 PROTOCOL $\Pi^\bullet$.

State : Initially, the set of chains $\mathbb{C}$ only consists of the genesis block. At round r, the PoS-player $P \in \mathcal{P}$, with (SK, PK) and local set of chains $\mathbb{C}$, proceeds as follows.
Upon receiving a chain $\mathcal{C}'$, set $\mathbb{C} := \mathbb{C} \cup \{\mathcal{C}'\}$ after verifying $\mathsf{Validate}^\bullet(\mathcal{C}', r) = 1$;
Compute $\mathbb{C}_{\text{best}} := D\text{-}\mathsf{BestChainSet}^\bullet(\mathbb{C})$;
for $\mathcal{C} \in \mathbb{C}_{\text{best}}$ **do**
  $\eta := \mathsf{Context}^\bullet(\mathcal{C})$; $B := \mathsf{Mining}^\bullet(\eta, r, \text{SK}, \text{PK})$;
  if $B \neq \perp$ **then**
    $\mathcal{C}_1 := \mathcal{C} \| B$; Broadcast $\mathcal{C}_1$;
  end
end

// Algorithms $\mathsf{Context}^\bullet$, $\mathsf{Mining}^\bullet$, $\mathsf{Validate}^\bullet$, and D-$\mathsf{BestChainSet}^\bullet$.
$\mathsf{Context}^\bullet(\mathcal{C})$:
  $\ell := \mathsf{len}(\mathcal{C})$; $\eta := h(\mathcal{C}[\ell])$; Return η;
$\mathsf{Mining}^\bullet(\eta, r, SK, PK)$:
  $\sigma := \mathsf{uSign}(\text{SK}, \langle \eta, r \rangle)$
  if $\mathsf{H}(\eta, r, \text{PK}, \sigma) < \mathtt{T}$ **then** Create new block $B := \langle \eta, r, \text{PK}, \sigma \rangle$; Return B;
  else Return $\perp$
$\mathsf{Validate}^\bullet(\mathcal{C}, r)$:
  Parse $\mathcal{C}$ into $B_0 \| B_1 \| \cdots \| B_\ell$;
  for $i \in [1, \ell]$ **do**
    Parse B_i into $\langle \eta_i, r_i, \text{PK}_i, \sigma_i \rangle$;
    if $h(B_{i-1}) \neq \eta_i$ or $\mathsf{H}(\eta_i, r_i, \text{PK}_i, \sigma_i) \geq \mathtt{T}$ or $\mathsf{uVerify}(\text{PK}_i, \langle \eta_i, r_i \rangle, \sigma_i) = 0$ or $r_i > r$ **then** Return 0;
  end
  Return 1;
D-$\mathsf{BestChainSet}^\bullet$:
  Set $\mathcal{C}_{\text{best}}$ as the longest chain in $\mathbb{C}$ and $\mathbb{C}_{\text{best}} = \{\mathcal{C}_{\text{best}}\}$;
  for $\mathcal{C} \in \mathbb{C}$ **do**
    if $\mathsf{distance}(\mathcal{C}_{\text{best}} \to \mathcal{C}) \leq D$ **then** $\mathbb{C}_{\text{best}} := \mathbb{C}_{\text{best}} \cup \{\mathcal{C}\}$;
  end

could be returned by $\mathsf{Mining}^\bullet$ if the following hash inequality holds: $\mathsf{H}(\eta, r, \text{PK}, \sigma) < \mathtt{T}$, where $\sigma := \mathsf{uSign}(\text{SK}, \langle \eta, r \rangle)$. The new block B_{i+1} is defined as $B_{i+1} := \langle \eta, r, \text{PK}, \sigma \rangle$. The pseudocode of our protocol $\Pi^\bullet$ can be found in Algorithm 1.

Security analysis. The security analysis techniques outlined in [10, 14, 8, 3] can offer valuable insights for analyzing the security properties of protocols based on the single-extension design framework. However, our protocol $\Pi^\bullet$ does not adhere to this framework, requiring new analysis techniques to establish its security properties.

We can prove the security properties of protocol $\Pi^\bullet$ under the assumption of honest majority of *effective stake*. Recall that in Section 3, we obtain that the adversary can amplify its stake by a factor $e = 2.72$, so we define the effective stake of the adversary as $\beta^\bullet = 2.72\beta$. Similarly, following the D-distance-greedy strategy, honest players can amplify their stake by an amplification ratio $\hat{\mathtt{A}}^\bullet_D$, and we define $\alpha^\bullet = \hat{\mathtt{A}}^\bullet_D \cdot \alpha$. Now, we formally state the theorem.

▶ **Theorem 5.** *Consider an execution of multi-extension protocol $\Pi^\bullet$ in the random oracle model, where honest players follow the D-distance-greedy strategy while adversarial players could follow any arbitrary strategy. Additionally, all players have their stake registered at the beginning of the execution. Assume* $(\mathsf{uKeyGen}, \mathsf{uKeyVer}, \mathsf{uSign}, \mathsf{uVerify})$ *is a unique digital signature scheme, and $\alpha^\bullet = \lambda\beta^\bullet$, $\lambda > 1$. Then protocol $\Pi^\bullet$ achieves 1) chain growth, 2) common prefix, 3) chain quality, and 4) the best possible unpredictability properties.*

The proof is shown in the full version of our paper.

References

1 NXT whitepaper, 2014. URL: https://www.dropbox.com/s/cbuwrorf672c0yy/NxtWhitepaper_v122_rev4.pdf.

2 Adam Back. Hashcash – A denial of service counter-measure, 2002. URL: http://hashcash.org/papers/hashcash.pdf.

3 Vivek Bagaria, Amir Dembo, Sreeram Kannan, Sewoong Oh, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Proof-of-stake longest chain protocols: Security vs predictability. *arXiv preprint*, 2019. arXiv:1910.02218.

4 Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Currencies without proof of work. In *Bitcoin Workshop*, 2016.

5 Jonah Brown-Cohen, Arvind Narayanan, Alexandros Psomas, and S Matthew Weinberg. Formal barriers to longest-chain proof-of-stake protocols. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 459–473, 2019. doi:10.1145/3328526.3329567.

6 Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, January 2000. doi:10.1007/s001459910006.

7 Phil Daian, Rafael Pass, and Elaine Shi. Snow White: Robustly reconfigurable consensus and applications to provably secure proof of stake. In Ian Goldberg and Tyler Moore, editors, *FC 2019*, volume 11598 of *LNCS*, pages 23–41. Springer, Heidelberg, February 2019. doi:10.1007/978-3-030-32101-7_2.

8 Bernardo David, Peter Gazi, Aggelos Kiayias, and Alexander Russell. Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 66–98. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78375-8_3.

9 Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 139–147. Springer, Heidelberg, August 1993. doi:10.1007/3-540-48071-4_10.

10 Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 281–310. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_10.

11 Aggelos Kiayias and Giorgos Panagiotakos. Speed-security tradeoffs in blockchain protocols. Cryptology ePrint Archive, Report 2015/1019, 2015. URL: https://eprint.iacr.org/2015/1019.

12 Jae Kwon. Tendermint: Consensus without mining, 2014. URL: https://tendermint.com/static/docs/tendermint.pdf.

13 Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. URL: https://bitcoin.org/bitcoin.pdf.

14 Rafael Pass, Lior Seeman, and abhi shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 643–673. Springer, Heidelberg, April / May 2017. doi:10.1007/978-3-319-56614-6_22.

15 Pavel Vasin. Blackcoin's proof-of-stake protocol v2, 2014. URL: http://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf.