

Unsealing the secrets of blockchain consensus: A systematic comparison of the formal security of proof-of-work and proof-of-stake

Iván Abellán Álvarez
Interdisciplinary Centre for Security,
Reliability and Trust, University of
Luxembourg
ivan.abellan@uni.lu

Vincent Gramlich
Branch Business & Information
Systems Engineering, Fraunhofer FIT
Germany
vincent.gramlich@fit.fraunhofer.de

Johannes Sedlmeir
Interdisciplinary Centre for Security,
Reliability and Trust, University of
Luxembourg
johannes.sedlmeir@uni.lu

ABSTRACT

With the increasing adoption of decentralized information systems based on a variety of permissionless blockchain networks, the choice of consensus mechanism is at the core of many controversial discussions. Ethereum’s recent transition from proof-of-work (PoW) to proof-of-stake (PoS)-based consensus has further fueled the debate on which mechanism is more favorable. While the aspects of energy consumption and degree of (de-)centralization are often emphasized in the public discourse, seminal research has also shed light on the formal security aspects of both approaches individually. However, related work has not yet comprehensively structured the knowledge about the security properties of PoW and PoS. Rather, it has focused on in-depth analyses of specific protocols or high-level comparative reviews covering a broad range of consensus mechanisms. To fill this gap and unravel the commonalities and discrepancies between the formal security properties of PoW- and PoS-based consensus, we conduct a systematic literature review over 26 research articles. Our findings indicate that PoW-based consensus with the longest chain rule provides the strongest formal security guarantees. Nonetheless, PoS can achieve similar guarantees when addressing its more pronounced tradeoff between safety and liveness through hybrid approaches.

CCS CONCEPTS

• **Security and privacy** → *Formal security models; Distributed systems security.*

KEYWORDS

Distributed ledger technology, dynamic availability, finality, liveness, PoS, PoW, safety

ACM Reference Format:

Iván Abellán Álvarez, Vincent Gramlich, and Johannes Sedlmeir. 2024. Unsealing the secrets of blockchain consensus: A systematic comparison of the formal security of proof-of-work and proof-of-stake. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC’24)*, April 8–12, 2024, Avila, Spain. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3605098.3635970>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SAC ’24, April 8–12, 2024, Avila, Spain

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0243-3/ 24/04.

<https://doi.org/10.1145/3605098.3635970>

1 INTRODUCTION

The proliferation of cryptocurrencies and decentralized applications following the introduction of Bitcoin [43] has spurred the need for effective designs of blockchain infrastructures with open (“permissionless”) participation. Blockchain technology provides a foundational data structure that enables the secure and intermediary-free transmission of both information and value in such decentralized networks [32]. At its core, a blockchain is a replicated, (probabilistically) immutable, ever-growing event log file that records all participants’ transactions in a well-defined total order [8, 40, 43]. Transactions are ordered and assigned to sequential batches known as blocks, each of which is cryptographically linked to the preceding block to achieve tamper evidence. This append-only structure facilitates an efficient synchronization process. Corresponding agreement rules that make synchronization robust and provide a shared and consistent view of the database even in the presence of partial system outages and malicious activities by some participants are called “consensus mechanisms” [19, 23, 67].

The consideration of faulty nodes that may crash or act maliciously in permissionless networks naturally involves the consideration of Sybil attacks: the ability of an entity to subvert and solely control many bogus identities at negligible costs that may participate in the blockchain network and, thus, influence the outcome of agreement processes [14, 51]. In general, consensus mechanisms for permissionless blockchains hence combine decision rules with voting procedures and Sybil resistance mechanisms that linearly couple voting weight to a scarce resource [60]. Bitcoin introduced a novel combination of a cryptographic data structure, Sybil resistance mechanism, economic incentives for participation, and agreement rules to implement such a blockchain network functioning in an open and decentralized setting [43]. More specifically, provably invested computational power dictates a node’s voting weight in its proof-of-work (PoW)-based longest chain consensus. However, while offering convincing heuristics and examples for security conditions like the maximum adversarial tolerance threshold for computational power acceptable for ensuring consistency (e.g., to prevent double-spending with high probability), the original Bitcoin paper did not derive formal security guarantees [43]. Similar observations hold for the originally proposed proof-of-stake (PoS) in Peercoin [29], which achieves Sybil resistance by coupling a node’s voting weight to its cryptocurrency coin holdings.

Contrary to permissionless blockchains, there are also “permissioned” constructions in which the number and identity of nodes are specified at the start of the protocol and that involve well-defined

processes to add or remove nodes during operation. Corresponding deterministic consensus mechanisms that tolerate a maximum number of faulty nodes are also known as Byzantine fault-tolerant (BFT) protocols and have been well-studied since the 1980s [38]. Permissionless blockchains can be considered a strict generalization of permissioned constructions, with a considerably lower degree of control over participants and their behavior, which brought new opportunities and challenges to distributed systems designs [43, 68]. Many established approaches to formalization and formal security analyses for permissioned networks turned out not to adequately accommodate these novel, naturally non-deterministic constructions. Consequently, alternative models and notions on the ideal functionality of distributed systems [23] have been proposed, alongside novel formal security properties permissionless blockchains should meet [19].

A significant challenge for organizations is the effective selection and management of permissionless blockchain-based infrastructures, which is manifested in discussions around corresponding design principles [55, 57]. The choice of the consensus protocol plays a crucial role. This role manifests in an ongoing controversial debate between the arguably two most prominent designs, namely PoW and PoS [26, 38]. Permissionless blockchains are restricted by several impossibility results (see Section 2.1 for details), such as the incompatibility of finality (safety) and dynamic availability (liveness) under non-synchronous network conditions. Many design choices have a significant impact on the consensus protocol's security properties [7, 36, 46, 52, 58]. In general, PoW is often touted for its security [21, 49] but faces significant criticism due to its high energy consumption [60]. This debate has witnessed significant contention particularly since the Ethereum blockchain introduced the "Merge", which marked its transition from PoW to PoS in September 2022. In particular, while PoS-based consensus undisputedly improves energy consumption substantially [10, 54], its impact on security is controversial [46, 47, 67]. However, while there is broad agreement that blockchain security aspects are crucial for ensuring reliability and trustworthiness [e.g., 12, 19, 20, 23, 37, 38, 46], security often seems overlooked.

We observe two main dimensions of the prevailing academic discussion on the security aspects of PoW and PoS in permissionless blockchain designs. The first dimension is a dedicated stream in the economic literature that considers the degree of decentralization and the characteristics of the time evolution of the distribution of the scarce resource that impacts an entity's voting weight in consensus: Computational power ("hash rate") in PoW and capital ("staked crypto-assets") in PoS [e.g., 2, 56]. The second dimension, the focus of this work, represents formal security properties, many of which are based on a given distribution of voting weights and threshold assumptions on honest protocol execution. Some related studies already offer a high-level comparative analysis of characteristics of permissioned and permissionless consensus protocols, including security aspects [e.g., 26, 69]. Two recent works [62, 70] provide a comprehensive survey of various consensus mechanisms and corresponding key components, including PoW and PoS-based Sybil resistance mechanisms and permissioned consensus approaches. [62] further extends and categorizes the literature while highlighting the main advantages, limitations, and applicability of various proof-of-x mechanisms. However, these works do not focus on

comparing the formal security aspects related to computational limitations, incompatibilities, known vulnerabilities, and associated mitigation strategies, and thus do not provide an in-depth understanding of the nuanced differences between PoW and PoS. On the other hand, seminal research on the formal security properties of permissionless consensus has primarily focused on analyzing individual blockchains and their consensus mechanisms, e.g., for PoW [19, 20] and PoS [12, 44]. On the other hand, [23] contributes a formal, consensus-agnostic framework to model protocols for security analysis, and [20] provides an analytical approach to describing the security properties of a PoW consensus protocol that is readily generalizable to cover also PoS-based constructions. However, there seems to be no general agreement on the appropriate collection of core security properties for permissionless consensus. In particular, a comprehensive and detailed comparison between prominent design choices for permissionless consensus mechanisms, and in particular between popular instantiations of PoW and PoS, is lacking. This paper closes this research gap by systematically analyzing the commonalities and differences between formalization aspects and formal security properties to comprehensively capture and compare the trade-offs inherent to PoW and PoS-based consensus mechanisms for permissionless blockchains. We hence ask the following two research questions:

RQ1. *What are commonly considered security properties for consensus mechanisms in permissionless blockchains?*

RQ2. *What are the commonalities and differences between PoW- and PoS-based consensus mechanisms regarding those formal security properties?*

To answer these research questions, we ground our study in peer-reviewed journals and conferences, following established guidelines for conducting systematic literature reviews (SLRs) [5, 30].

2 BACKGROUND

2.1 Historical Overview of Decentralized Systems Security

To understand the functionalities and security characteristics of information systems, research often employs formalization that takes into account the behavior of the system's individual components. A common formalization model for blockchains is state machine replication (SMR), which characterizes the behavior of every individual node of a distributed system [59]. SMR thus provides an abstract model for a system of deterministic machines ("nodes") that handle information processing by individual storage and mutual communication [59]. SMR captures an ordered sequence of inputs ("transactions"), with the goal of ensuring a consistent and logical execution such that from the perspective of clients, the decentralized and in particular distributed system consisting of many nodes behaves like a highly reliable centralized system (e.g., running on a faultless server). SMR is widely used to formalize and prove the security of permissioned consensus protocols [59]. The consensus protocol accordingly updates the current state of each local SMR node [59]. SMR formally ensures consensus protocol correctness (i.e., qualifies as BFT) if a system satisfies the following key properties [33]: 1) *Safety*: The protocol does not produce contradictory states among non-faulty nodes, thus ensuring a consistent and persistent view [19, 38]; and 2) *Liveness*: correct

processing of transactions will eventually happen upon correct input [1, 50], i.e., the distributed system keeps making meaningful progress over time and is useful to interact with [37]. Whether or not a distributed system satisfies the safety and liveness properties can vary depending on the number of faulty nodes, the use of additional building blocks (e.g., digital signatures and public key infrastructure for authenticated messaging [15, 24]), and other conditions, such as network reliability and performance [21, 34]. For representing the mentioned network properties, three models are widely used, namely synchrony, partial synchrony, and asynchrony. A synchronized network communicates reliably with a bounded maximum transmission time [37, 61]. Asynchrony, on the other hand, considers potentially unbounded communication time delays and therefore offers no control on the number of lost messages [37, 61]. Partial synchrony is between these two extremes, denoting two synchronization states: Periods of asynchrony that last for an unknown time, and periods of synchrony that eventually occur [15, 37]. Partial synchrony is widely considered to be an appropriate model for the Web [22].

The Dolev-Strong protocol presented in 1983 [13] marks one of the first solutions for permissioned consensus under synchrony. By making use of authenticated messaging, it can handle any number of faulty nodes, as digital signatures on all messages ensure that malicious nodes that supply different nodes contradictory information are detected. The protocol involves honest nodes incrementally affirming a decision by adding digital signatures, resulting in a BFT solution. However, distributed systems with potentially faulty nodes are fundamentally constrained when facing unreliable network conditions. In 1985, [18] proved the FLP theorem that states that in a permissioned system under asynchronous network assumptions, no deterministic solution for consensus exists even if only a single node may crash. However, solutions that achieve both safety and liveness with high probability exist when non-deterministic components are used. HoneyBadgerBFT [42], proposed in 2016, is an example of such a non-deterministic permissioned consensus protocol with practical performance under asynchronous network conditions. Furthermore, the CAP theorem dictates fundamental security principles for distributed systems under stronger synchronicity assumptions. After the initial conjecture by Brewer in 2000 [6], the CAP theorem was formalized and proved in 2002 [22]. The CAP acronym stands for *Consistency* (i.e., safety in the sense that all operations execute in the same order for every available honest node), *Availability* (i.e., liveness on messages that are eventually delivered, thus implying that the execution terminates), and *Partition tolerance* (i.e., tolerating certain network failure events where lost messages lead to a split of the network into isolated parts) [6, 22]. The theorem states that distributed systems can only fulfill only two out of these three properties in the partially synchronous (and, therefore, also in the asynchronous) setting [22]. Yet, solutions exist in partial synchrony if the consistency requirement is weakened [18, 22].

While some recent advances have been made in permissioned consensus, e.g., substantial performance improvements by reductions of message complexity in non-deterministic BFT protocols for asynchronous networks [42] or deterministic BFT protocols under partial synchrony [45, 72], research in the last decade has primarily focused on understanding the emerging permissionless systems. It turns out that in the permissionless setting, there is an analogous

impossibility result to the CAP theorem [36, 58]. It shows fundamental incompatibilities between finality and dynamic availability properties under a partially synchronous network. This incompatibility is an essential characteristic that distinguishes PoW- and PoS-based consensus protocols [37, 38] (see Section 4.1). *Finality* defines the consistency of states and irreversibility of transactions or blocks. While finality is naturally satisfied in deterministic BFT protocols, many non-deterministic consensus protocols only provide *probabilistic finality*, e.g., with the probability of reversibility decreasing exponentially with each new block added to the blockchain in longest-chain rule PoW [19, 36] (see Section 4.3). *Dynamic availability*, on the other hand, refers to a network characteristic that can provide liveness even in the presence of arbitrary fluctuations in the participating nodes or their voting power [36]. Intuitively, the reason behind the dynamic availability-finality dilemma is the fact that from the perspective of an honest node, network partitions are indistinguishable from diminishing network participation [36]. Consequently, dynamically available networks must “keep growing the chain” even in the case of a network partition [58], which may produce a split view [58]. Such a split view, by definition, compromises consistency (safety) if both sides are finalizing transactions [36].

2.2 Proof-of-Work and Proof-of-Stake Constructions

Permissionless settings need to ensure the consistency of blockchain nodes among honest participants under an honest-majority assumption (in some metric). Consequently, they must account for faulty nodes and in particular for Sybil attacks (see Section 1). Permissionless blockchain consensus hence relies on the integration of a Sybil resistance mechanism in addition to a rule for choosing the valid state among potentially multiple options (“forks”). These mechanisms require the expenditure or investment of a scarce resource for participation in consensus [21, 64]. PoW and PoS are Sybil resistance mechanisms that also incentivize honest behavior in consensus through rewards. Honest behavior is typically rewarded in the form of fixed block rewards and variable transaction fees for block producers, as well as compensation for further activities in consensus. These incentives are distributed as tokens of the native cryptocurrency that every permissionless blockchain network needs as the basis for its compensation mechanism. Adversarial attempts are typically penalized by depriving rewards for the expenditure of the scarce resource or corresponding opportunity costs, as well as potential capital forfeiture in PoS [44, 46].

The first ever introduced permissionless blockchain, Bitcoin, is governed by PoW in combination with the longest-chain rule (“Nakamoto consensus”) [43]. PoW defines a verifiably computationally intensive mechanism that on average proves the provisioning of a certain amount of computing hardware and electric energy [16] to regulate the block production rate (i.e., block proposers and the time intervals at which blocks are appended to the global ledger). Blocks represent a solution to a cryptographic puzzle that is used to determine the average amount of computational power provided by a participant. Solving this puzzle makes the node operators (“miners”) eligible to append the block under consideration to the existing chain. The canonical longest chain is defined to be the

chain that requires the largest computational effort for construction, i.e., “length” is determined according to a weighted sum over all blocks, where a block’s weight is determined by the difficulty of the puzzle solved by the block.

Permissionless blockchain transaction throughput rate is well-known to be low: Throughput defines the computational, bandwidth, and storage resource requirements of each node; such that only a low throughput keeps the barrier to participation sufficiently moderate to make decentralization in the long run possible [57]. Moreover, low throughput usually involves small blocks (i.e., fast propagation) with a slow block production rate, which is beneficial for security: It decreases the probability of unintentional forking, where honest nodes try to build on a block that is not the most recent one because they have not yet received the latest block [19, 21]. However, this low throughput of blockchains compared to centralized systems restricts the scalability [63] and is hardly compatible with the real-time requirements in organizations [25]. A faster block production can help both to reduce transaction confirmation latencies and achieve slightly higher transaction throughput by facilitating a more continuous use of computation and bandwidth resources as long as storage is not the bottleneck. To increase the block production rate without compromising security, the Greedy Heaviest Observed Subtree (GHOST) protocol was proposed [63] for permissionless blockchain consensus protocols. In GHOST, new blocks are appended to the previous most voted block (i.e., higher weight) while stale but valid blocks still influence the chain [44, 45, 47, 67]. The canonical chain is, therefore, the heaviest in terms of votes in contrast to the longest in terms of counting or weighted by the difficulty of the PoW puzzle [47]. These constructions by which honest nodes decide where to append a new block they propose are often referred to as “fork-choice rules”.

In PoW, miners face direct costs for participation in consensus through their contribution of hardware and electricity, which puts a strong incentive for them to behave honestly: The cryptographic puzzle is dependent on a specific batch of transactions and a previous block, i.e., the chain a miner decides to extend [19]. Consequently, miners have to wisely choose how to use their resources, and if they use them on a block deemed invalid by other miners, or for extending a chain that is not currently the longest one, they face a substantial risk that their block will not be respected by the majority of other miners, i.e., they gain no rewards and their resources are wasted [21]. PoS replaces hardware and electric energy as a scarce resource to which voting power is coupled by capital in the form of ownership of native cryptocurrency tokens, which is also publicly verifiable through the transparent accounting in the permissionless blockchain network [7]. Participants deposit capital (“stake”) to signal their willingness to participate in consensus. They are then selected to act as block proposers or to validate and attest to blocks as part of a committee, typically with probability equal or close to their share of the total stake, using some source of (pseudo-) randomness generated in the protocol [7, 28].

Consequently, contrary to PoW, PoS faces the issue of “costless simulation”, leading to the *nothing-at-stake* problem: Any node can at negligible cost create different blocks at the same height (“equivocation”) that could potentially be both deemed correct and included by honest nodes [44]. In the case of the existence of forks,

i.e., alternative chains with equal height, nodes even have an incentive to engage in equivocation because this makes them eligible for rewards in any future scenario chosen by the majority of nodes. Hence, the costless simulation issue would imply that a fork may never be resolved. To address this shortcoming, miners can be held accountable for observed misconduct, including equivocation, through their staked capital (“slashing”) [7, 44, 46]. Another way to address costless simulation and the risks of equivocation is to achieve immediate finality by using the Sybil resistance mechanism only to determine a subset of nodes that subsequently run a (permissioned) BFT protocol [53, 58]. This approach and check-pointing services that continuously mark a set of blocks as finalized after a relatively short time can also effectively prevent long-range attacks in which attackers have already disposed of their collateral at the time of launching an attack with an alternative chain [58]. Hence, regardless of the fork-choice rule, many common constructions of PoS systems employ permissioned BFT protocols to guarantee safety, either for immediate finalization or check-pointing services [7, 45, 58]. Alternatively, modifications to the fork-choice rule can yield secure constructions if the leader election process is sound [28]. The current state of the Ethereum consensus protocol combines the GHOST protocol with Casper the Friendly Finality Gadget (FFG) (this combination is often called “Gaspar”) [45, 67]. Casper (FFG) combines a PoS-based fork choice rule, weighted by attached attestations’ stake, and a BFT-style protocol that provides a check-pointing service for finalizing blocks [7, 46]. The eligibility of block proposers and validators is drawn from the required deposited stake from which smaller-sized committees are formed every round [7, 45] or randomly selected by weighted stake [28] to validate blocks.

3 METHOD

To answer our research questions (see Section 1), we comprehensively collected relevant academic works by conducting an SLR following the guidelines of Kitchenham [30]. Based on a basket of literature we had already collected and investigated in a preliminary study, we defined our search string: (“*security analysis*” OR “*adversarial attacks*” OR *liveness* OR *finality* OR *safety*) AND (proof*of*stake OR proof*of*work OR *Nakamoto* OR *Bitcoin* OR *Ethereum*) AND (protocol OR consensus). We then applied this search string to a set of academic databases based on their relevance to the study topic [30, 31]. We selected four popular computer science databases that encompass journals and conference proceedings: ACM Computing Library, IEEE Xplore, ScienceDirect, and SpringerLink. For ACM Digital Library and IEEE Xplore, we applied the search string as specified. To keep the effort for searches manageable, we further tailored the search string for ScienceDirect and SpringerLink: We excluded the keywords *liveness*, *finality*, and *safety* to reduce the otherwise $n = 1763$ results in SpringerLink down to 558. Due to limitations in the number of Boolean operators supported in ScienceDirect (max. 8), we ran two independent queries, including either “*protocol*” or “*consensus*”. These two queries yielded 38 and 39 papers, respectively. Through building the union of these results (i.e., removing duplicates), we arrived at an initial selection of 41 publications from ScienceDirect.

We illustrate the subsequent paper selection process in Figure 1. To guide the SLR, we defined a set of inclusion and exclusion criteria that we applied to the screening and filtering of title, abstract, and full-text [5, 30, 31]. We include works formalizing blockchain characteristics and properties, e.g., by describing an ideal functionality, analyses of consensus protocols that involve PoW- and PoS-based Sybil resistance mechanisms, investigations of corresponding security implications, attacks against such systems, and corresponding mitigation approaches. On the other hand, we exclude publications that 1) put an exclusive focus on permissioned consensus or Sybil resistance mechanisms beyond PoW and PoS, 2) consider only alternative blockchain constructions unrelated to PoW or PoS-based consensus mechanism, 3) represent high-level surveys on a broad set of consensus mechanisms, or 4) lack a formal evaluation method of the prescribed model. We carried out the filtering process by applying the inclusion and exclusion criteria at every step. Initially, our search string yielded a total of 746 results across all four databases. A subsequent title screening resulted in 91 remaining papers. We then evaluated these publications' abstracts and removed duplicates, narrowing our selection down to 48 publications. Lastly, a full-text analysis of these papers yielded our final selection of 26 publications.

We classified the selected literature according to the topics it addresses among three groups: blockchain formalization, PoW-based blockchain constructions, and PoS-based ones. We then extracted the security properties discussed in these groups and mapped them into related groups to reflect, for instance, the close connection between safety, consistency, and finality; as well as the tight relationship between liveness, dynamic availability, chain quality, and chain growth (see Section 4.1). We thus formed a structured overview of blockchain security properties that we use as a basis for answering RQ2.

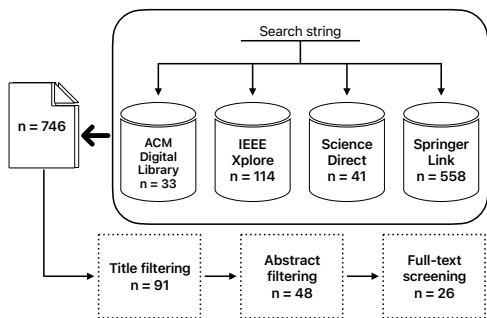


Figure 1: Systematic literature review process.

4 RESULTS

The following section presents the results we extracted from the selected literature. In Table 1, we categorize all the relevant literature with respect to the consensus protocol constructions it analyzes and the considered security properties. Table 1 represents the possible construction of consensus protocols via combinations of Sybil resistance mechanisms and a fork-choice rule according to what we have defined in our literature selection criteria, namely PoW

and PoS. It also features the relation of all papers with the identified security properties. Section 4.1 discusses abstract models of the functionalities and the security properties of permissionless blockchains. We then provide a comprehensive overview of formal security properties of PoW and PoS-based consensus mechanisms in Section 4.3 and Section 4.4, respectively.

4.1 Blockchain security formalization

Blockchain formalization and abstraction are fundamental in characterizing the architectural model of blockchain to prove security properties. In their seminal work, Garay et al. [19] provide the first formal proof and formalize blockchain security properties under synchronous network conditions drawing from the longest-chain fork-choice rule design (i.e., everyone agrees to append blocks to the longest chain seen, weighted by difficulty) and the PoW Sybil resistance mechanism. The authors introduce three essential security properties that represent necessary conditions to guarantee safety and liveness: 1) *common prefix* describes the existence of a large commonly agreed sub-chain and provides probabilistic safety guarantees, 2) *chain quality* describes the ratio at which honest blocks are included in the chain and represents a liveness property, and 3) *chain growth*, the speed at which the chain grows, i.e., keeps recording blocks.

Common prefix with security parameter $k \in \mathbb{N}$ [19, 28]: For any node l , let C_l the current view of the chain and $C_l^{[k]}$ denote this chain with the last k blocks removed. Then any two honest nodes i and j have consistent view of the chain up to the k last blocks, i.e., $C_i^{[k]} \leq C_j^{[k]}$ or $C_j^{[k]} \leq C_i^{[k]}$, where \leq denotes “is prefix of”.

Chain-quality with parameter $\mu \in [0, 1]$ [28]: For any reported chain from the common prefix of an honest node, the ratio of adversarial blocks is at most $1 - \mu$.

Chain-growth with parameter $\tau \in [0, 1]$ [19, 28, 46]: Let C_t be the chain at time t for $t \geq 0$ and $|C_t|$ the length of C_t . Then C has chain growth τ if $|C_t| - |C_s| \geq \tau \cdot (t - s)$ for all $0 \leq s \leq t$.

The formalization and blockchain security notions in [19] are also leveraged by many other works for analyzing the formal security aspects of different blockchain consensus designs [e.g., 12, 21, 27, 40, 46, 52, 53, 58, 64]. [3] complements previous work by constructing a UC-secure PoW longest chain blockchain under synchrony. The UC framework [9] serves as an abstract and general model to define composable protocol functionalities, providing strong security guarantees also in concurrent protocol execution. The ideal functionality in [3] enables the analysis of various customizable properties, including synchrony assumptions and adversarial capabilities. [23] generalize the ideal functionality of blockchains, including [3], in the form of an ideal ledger that reflects the broad spectrum of blockchain properties (e.g., consensus mechanism and synchronization models). Their ideal ledger comprises a globally ordered list of transactions on a global state, which can be interacted with through various actionable subroutines (e.g., read/write operations). Alternative forms of abstraction, such as automata, are also suitable to prove safety and liveness properties [4]. To further evaluate security notions, [40] strengthens previously defined security properties, namely *common prefix*, *chain quality*, *safety*, *liveness*, and *chain growth* [19].

Table 1: Classification of the literature selected in our SLR.

Fork-choice rule	Consensus protocol								Formalization
	Sybil-resistant mechanism								
	Proof-of-Work				Proof-of-Stake				
	GHOST	Longest-chain			GHOST	BFT/Quorum			
BFT/Quorum-checkpointed									
Security properties									
Safety (consistency, persistence)	[12, 20, 21, 52]	[19, 21, 40, 52, 71]	[7, 27, 45, 46, 53, 58]	[7, 41, 45, 46]	[12, 20, 28, 64]	[39, 44–47, 67]	[4, 24]		
Liveness	[12, 20, 52]	[19, 40, 52]	[7, 27, 45, 46, 53, 58]	[7, 45, 46]	[12, 20, 28, 64]	[44–47, 67]	[4, 24]		
Common prefix, chain-quality, chain growth	[12, 20, 21, 52]	[12, 19, 21, 40, 52]	[27, 46, 53, 58]	[45–47]	[12, 20, 28, 64]	[39, 45–47]			
Finality			[7, 27, 45, 46, 53, 58]	[7, 45–47]		[39, 44–46, 67]	[4, 24]		[3, 23]
Dynamic availability		[19, 73]	[27, 45, 46, 53, 58]	[45–47]	[28]	[45–47]			
Other formal properties	[21] ¹ , [52] ² , [66] ⁶	[21] ¹ , [35] ² , [52] ² , [71] ^{1,2} , [73] ²	[27] ³ , [46] ³ , [53] ³ , [58] ⁴	[46] ³	[73] ²	[46] ³	[4] ⁷		

¹ Block size, network delay, stale blocks rate.² Reward distribution.³ Availability-accountability dilemma.⁴ Availability-finality dilemma.⁵ Order-fairness.⁶ Bootstrapping.⁷ System convergence.

PoS addresses safety against long-range attacks by offering finality guarantees (either immediate, through check-pointing [7], or probabilistic [28]). As the current total stake is a publicly visible figure [36], they must involve a BFT-style routine with “honest supermajority” assumption for immediate finality or “honest majority” for probabilistic finality guarantees. Consequently, they cannot tolerate an unknown threshold of stake represented by inactive or disconnected participants [37]. On the other hand, the total hash rate of a PoW-based permissionless blockchain system is unknown and can fluctuate unpredictably with nodes joining, leaving, or adapting their mining efforts. Therefore, prioritizing dynamic availability seems natural for longest-chain PoW constructions.

One property of blockchains that prioritize safety is to guarantee that no two different blocks at the same height are finalized. In general, BFT protocols such as Casper identify adversarial behavior and forfeit their stake in the event of equivocation as long as the majority stake is controlled by honest nodes [46]. Generally, the nodes in control of the majority stake would be in a position to signal adversarial nodes’ behavior and hold them accountable for their acts. However, [46] shows conflicts between dynamic availability and accountability with respect to safety and liveness, thus defining another dilemma. On the one hand, permissioned BFT protocols provide safety and can tolerate up to one-third of adversarial participants [4, 46] in non-synchronous networks while remaining accountable, as participants are identified within the network [46]. In contrast, dynamically available blockchains provide liveness regardless of the specific Sybil resistance mechanism [46], with the constraint that the majority of the resource must be controlled by honest nodes [12, 19, 40]. Common PoS gadgets such as Casper, and Gasper satisfy safety and provide finality and accountability but lack strong liveness guarantees, as demonstrated by known attacks [39, 44, 46, 47]. More details on PoS gadgets can be found in Section 4.4.

As the dynamic availability-finality dilemma and the dynamic availability-accountability dilemma provably cannot be resolved with a single consensus design, hybrid solutions utilizing dual ledgers emerge as a viable approach to tackle this trade-off. Dual ledgers leverage two different, “user-dependent” sets of consensus rules. For instance, while one main chain is dynamically available, tolerating nodes leaving and joining, the other is a check-pointed prefix of the previous consisting of finalized blocks, thus prioritizing safety in the event of de-synchronization [46, 58]. [46] shows that existing accountable and safe BFT protocols can also be used

all together as part of the consensus rules pursuing dual ledger strategies [45, 46].

4.2 Bootstrapping and blockchain continuity

Permissionless blockchains’ inherent properties are not limited to security aspects regarding continuous operation but also the seamless process of synchronizing nodes. Any de-synchronized node – whether because of a newly spawned node or temporarily in a partition – must be able to obtain a verifiable latest state of the protocol, i.e., to reach the subset of blocks that form the common prefix. Bootstrapping is the process by which nodes synchronize their local state with the globally agreed state [53]. The ability of a blockchain to allow such verifiable bootstrapping with minimal trust assumptions is that the blockchain is objective [65]. One powerful notion of permissionless PoW blockchains with the longest-chain rule weighted by difficulty is that they are objective [65]. Nodes can reliably reach the latest chain state by locally reconstructing and verifying the chain without any external contribution as long as the node is connected to at least one other honest node. Note that all permissionless blockchains are required to agree on an initial trusted source for the genesis block and node software as public given parameters. In contrast, PoS blockchains are weakly subjective: Nodes need external sources of information such as an additional set of recent blocks agreed to be valid (i.e., check-pointed) to determine which is the latest agreed state and, thus, to identify the canonical chain [65].

Bootstrapping a blockchain is an important characteristic that can be negatively affected by eclipse attacks [66] or forking events, e.g., in long-range attacks in PoS [11, 53]. In eclipse attacks, adversarial nodes supply invalid blocks to their neighboring nodes, therefore disrupting the synchronization process [53]. Check-pointing methods as suggested in [58] can address this issue [53]. Check-pointing and, thus, relying on an oracle to query external messages for validation is a requirement for safety in PoS protocols [37]. Therefore, there seems to be a correspondence in blockchain design, where objective blockchains (e.g., PoW) only provide probabilistic immutability, in other words, probabilistic finality, whereas weakly subjective ones such as PoS can achieve finality.

4.3 Formal security analysis of PoW

Going beyond the original heuristics presented in the Bitcoin Whitepaper [43], [19] formally proves that safety (i.e., *common prefix*) is satisfied with high probability only under the assumption

of honest majority ($> 50\%$, weighted by hash rate), and a tightly connected network, i.e., message delivery of blocks is fast compared to the block production rate, which represents a stronger assumption than synchrony. [19] also proves that there is a maximum number of blocks, and waiting time after which an honest transaction is guaranteed to be included, thus defining the liveness (i.e., *chain quality*) property [71]. [20], which builds upon [19], then proves the common prefix and the liveness property including *chain growth* under partial synchrony.

In PoW, a node's share of total invested computational power in the blockchain network is equal to the probability at which a new block building on a given latest block can be found by either party (both for honest and adversarial participants) [19]. However, the provable *chain quality* starts to degrade significantly when the adversarial threshold approaches 50% , suggesting a potentially asymmetric increase in the share of blocks proposed in comparison to the share of computational power [19]. Indeed, [17] shows that by adopting "selfish mining" where an entity mines on a private chain and strategically delays the release of blocks, adversarial nodes can gain a disproportionate advantage (i.e., contribute more blocks to the canonical chain) and, therefore, also earn more rewards than honest nodes [21, 71]. As such, PoW longest-chain protocols fail to appropriately reward participation according to their portion of the share of the hash rate [19]. The probability of successfully publishing a private chain, thus inducing a reorganization of blocks, still increases exponentially with the adversary's share of voting power [21, 52]. More precisely, the relative revenue of selfish mining depends on the fork-choice rule followed by honest nodes [71] and other factors such as the stale block rate (i.e., valid blocks that collide with others resulting in non-inclusion) and network partitions, i.e., forks [21, 71]. These findings show the need for tighter upper bounds on the adversarial threshold of computational power given the side effects of message delays, and the throughput of block creation to guarantee the common prefix and chain quality [19]. Simulation results in [71] confirm these upper bounds. As the share of the hash rate increases beyond 33% , the relative revenue of selfish mining increases disproportionately [21, 71]. On the other hand, the capacity of successfully and selfishly mining multiple consecutive blocks, thus causing a reorganization, still decreases exponentially with each new included block in the canonical chain [19, 40, 43]. Reorganizations benefit adversaries either by earning additional block rewards (degrading the chain quality) or as a consequence of completing successful double spends (compromising safety) [19]. [12] shows that selfish mining is indeed the worst possible adversarial attack on the longest-chain-based PoW consensus. Incorporating random choice in the fork-choice rule for a node that learns about two different longest (in particular, equally long) chains at roughly the same time mitigates the "worst-case" achievable through selfish-mining [17], with a corresponding upper bound of $\approx 33\%$ on the adversarial threshold.

Other factors, such as network latency and block propagation time, also influence the security of PoW-based consensus protocols [21]. [12] demonstrates that the probability of adversarial events is a function of the block propagation rate and the adversary's share of the computational power. A slow block production rate offers higher levels of security [52] as it benefits consistency [21], ensures the common prefix security property [19], and

keeps the probability of successful double-spends low [52]. [21] also shows that lowering the propagation rate of valid block inclusion up to a certain threshold that depends on the average propagation time (e.g., approx. 1 minute in Bitcoin) does not considerably affect the security assumptions.

4.4 Formal security analysis of PoS

PoS protocols have been proposed and implemented similarly to the original PoW in Bitcoin by heuristically assuming its security as in [29]. Several attacks that severely affect the common prefix of the ledger need to be accounted for in PoS blockchains [41]; most prominently, *nothing-at-stake* and *long-range* attacks (see Section 2.2). Additionally, PoS can be vulnerable to *grinding* attacks, where nodes exploit weaknesses of pseudo-random number generators used for electing block proposers in order to gain an advantage in the probability of being elected [11, 28]. [41] shows two ways of tackling such attacks: by authenticating and binding nodes' identities to proposed blocks, and by using trusted hardware for block production. [28] presented the first formalization for a PoS blockchain and proved *consistency* and *liveness* guarantees with respect to an adversarial threshold close to 50% .

Subsequent works on PoS constructions have been centered around mitigating or preventing these attacks to guarantee the security properties of blockchain such as safety (*consistency* and *common prefix*) and liveness (*chain-quality* and *chain growth*). PoS in synchronous networks with the longest-chain rule, when assuming that the election process of block proposers is a random process [28], maintains similar security properties to PoW with respect to safety and liveness, including *chain quality*, *chain growth*, and *common prefix* [12, 20, 64]. The key difference lies in the honest majority requirement, with PoS necessitating $> 66\%$ of the stake to be controlled by honest and participating peers [44, 64] compared to $> 50\%$ in PoW [19]. PoS necessitates the depositing of a stake to an intrinsic public key infrastructure (PKI). The stake is relative to the known number of public keys in the system, thereby heuristically fostering a BFT protocol. Further, its dynamic availability can be considered analogous to a non-synchronous network. Interpreting the underlying protocol as BFT-style, consensus can be achieved if at least $\approx 66\%$ of the voting power is controlled by honest nodes. Alternatively, consensus protocols that integrate several key components such as a new rewarding scheme, a modified longest-chain fork-choice rule, and forward-secure signatures can result in an approximate Nash equilibrium that disincentivizes validators from deviating from the protocol and thus reduce the tolerable adversarial threshold to $< 50\%$ [28]. Note that these schemes differ substantially from the ones coined from the PoS variants proposed by Peercoin [29] and implemented in Ethereum [45]. In particular, [28] incorporates a publicly verifiable random function that ensures the generation of a globally verifiable random value, effectively tackling grinding attacks.

[7] introduce some modifications to the PoS-based longest-chain protocols in the form of Casper (see Section 2.2) to address long-range attacks. Casper finalizes blocks of the canonical chain and holds adversarial nodes accountable (e.g., through slashing), mitigating the nothing-at-stake problem [7, 46]. [44] and [39] present a set of formal proofs, grounded in prior reasoned arguments by

[7] on the safety of Gasper. The authors show safety for Gasper for up to 33 % of adversarial stake. As Gasper employs the GHOST protocol, orphaned/stale blocks that are valid and received votes still influence the chain [47, 71]. These events influence the security of the system by facilitating new attacks that may harm the liveness of the protocol [71]. Indeed, [47, 67] show a lack of liveness of this implementation. [47] further finds vulnerabilities in two variants of the GHOST protocol that can be exploited with even less than 33 % of adversarial stake. Both designs suffer from variants of long-range attacks and equivocation in combination with leveraging the influence of orphaned blocks to displace or split the canonical chain. Owing to costless simulation, nodes can generate several blocks in Gasper and equivocally vote for them, which potentially leads to an “avalanche” or “balancing” attack. As a result, the canonical chain can be displaced or forked for an undefined amount of time [45, 47]. Displacement shifts grow the chain horizontally and vertically by leveraging the voting weights of equivocating blocks. Withheld blocks are altogether released horizontally on top of the same privately mined block (i.e., in the form of an “avalanche”), thus shifting (i.e., forking) the canonical chain. This results in a lack of safety and liveness of the protocol [47]. To address this issue, a modification called Latest Message Driven (LMD) was introduced [47]. The LMD modification affects the decision on GHOST by accounting only for the latest voted block instead of an unbounded number of previous blocks so that it tackles equivocating on multiple equal height blocks [45]. However, [47] and [45] describe a “balancing” attack that affects the LMD GHOST implementation, thus demonstrating a potential lack of safety as a consequence of an undefined but constant chain split. An adversary with a small fraction of the stake can timely equivocate on blocks to make both chains grow at the same time. Notably, only the first attempt at forking the chain view by equivocating represents a slashable action. Both [67] and [47] describe scenarios where in the event of a chain fork, an adversary is able to keep a split view of the chain such that half of the *validators* see either side. As a consequence, no chain is ever finalized, thus breaking the liveness property.

5 DISCUSSION AND CONCLUSION

This paper focuses on comparing the formal security aspects of PoW- and PoS-based consensus mechanisms. We answer *RQ1* by consolidating established security notions and corresponding distributed computing impossibility results. The formal blockchain security properties we identified are *safety*, *consistency*, *common prefix*, *finality*, *liveness*, *chain quality*, *chain growth*, and *dynamic availability*. Additionally, the paper highlights that *safety* is related to the *common prefix*, *consistency*, and *finality* properties. We discuss impossibility results including FLP [18], CAP [6, 22] to emphasize key theoretical limitations of distributed networks and outline “upper bounds” on security properties of PoW- and PoS-based permissionless blockchains. We also point out that these limitations can, to some extent, be overcome by defining security properties probabilistically with respect to a security parameter [18, 36, 37] (e.g., probabilistic finality instead of absolute finality [36]) and by strengthening assumptions about network partitions, i.e., synchrony properties [36]. Due to the discussed impossibility results and dilemmas, modeling security in permissionless networks needs

relatively strong assumptions on network synchrony to assure safety and liveness [38, 46, 58]. Furthermore, non-deterministic protocols are the only known approaches to solving consensus in permissionless networks with inherently dynamic voting power distribution [38]. [67] notes some evaluation disparity between academic and practical design implementations of consensus protocols concerning security properties. While the former often reason based on tight assumptions to prove security properties rigorously, the latter tend to relax the adversarial capabilities to accommodate the desired proofs.

Our path toward answering *RQ2* involves several key observations. For both PoW and PoS-based consensus with the longest-chain rule, under specific honest majority assumptions after some reasonable time period, blocks are part of *common prefix* [19, 47]. In general, blocks are included in the canonical chain in PoW with overwhelming probability (i.e., *probabilistic finality*). Similar patterns can be observed in probabilistic PoS constructions such as [28], while finality is immediate or satisfied after a certain finite time in PoS constructions using BFT protocols. Every honest node ends up sharing a common subset of blocks (i.e., *consistency*) [12, 19, 47]. PoW and probabilistic PoS variants achieve these properties if honest nodes control the majority of the voting power ($> 50\%$) when assuming tight and favorable network conditions. BFT-based PoS, on the other hand, requires honest nodes to control over 66 % of the voting power. However, PoW also requires an honest 66 % majority to ensure sufficient chain growth and quality due to the profitability of selfish mining with a voting power above 33 % [19, 71]. PoS longest-chain protocols can achieve comparable thresholds in networks where synchronicity is guaranteed [28, 64]. PoS constructions that depend on accountable gadgets such as Casper can guarantee safety regardless of the fork-choice rule even under powerful adversaries [67]. Liveness was initially posited to be guaranteed in a static validator setting and a supermajority of honest controllable voting power [44] and later proven under tightly synchronized networks assuming $< 50\%$ of adversarial controlled stake in [64].

We also found some inherent trade-offs affecting both PoW and PoS, such as the *dynamic availability-finality* dilemma [37, 38, 58] and the *availability-accountability* [46] dilemma. Balancing these trade-offs involves prioritizing specific designs based on the desired capabilities of the network. PoS with dual ledgers are placed in as a potential solution to overcome such trade-offs between safety and liveness. Particularly, Gasper provides liveness under a simple honest majority assumption ($> 50\%$) and satisfies the *common prefix* property if a supermajority ($> 66\%$) of stake is honest [47]. Similar thresholds have been shown for other accountable and safe BFT protocols [45, 46].

Researchers also incorporate slightly less common formal security or other blockchain properties, such as the *network delay* on message communication, the *stale block ratio*, and the *order-fairness* of transactions [27]. Our review indicates that if the block propagation rate is sufficiently slow to ensure a low stale block ratio, PoW-based Nakamoto consensus offers stronger formal security guarantees regarding *common prefix*, *consistency*, and *liveness* [21, 71] than PoS. This property can be traced back to PoW’s choice of the utilization of computational resources in achieving Sybil resistance, as investing computational resources is inherently

dynamic in nature. PoW causes substantial costs for block production that inherently avoids the need to address equivocation and long-range attacks. Nevertheless, longest-chain consensus variants with PoS-based construction provide similar guarantees when specific attacks are addressed [7, 64]. On the other hand, while PoS with GHOST allows for higher block production rates and, therefore, lower latencies and higher throughput when storage is not the bottleneck [21, 71], it comes at the expense of increased complexity and weaker formal security guarantees [47]. PoS gadgets aim to ensure *safety* by providing *finality*, but can be compromised with non-zero probability with a small fraction of adversarially controlled stake [47].

According to our SLR, no single solution currently addresses all of the desired security properties. It seems that PoW and PoS-based consensus protocols have already approached their optimal design, given the constraints posed by the impossibility results and dilemmas surveyed. Consequently, there is a line of work on gadgets that aim to satisfy various blockchain security properties in the form of dual ledgers to circumvent the impossibility results and dilemmas [45, 46, 58]. Each ledger prioritizes one security property, e.g., *safety* during partitions (i.e., finalized blocks) or liveness (i.e., the chain keeps growing and is dynamically available), and each user can pick their priority depending on the intended transaction.

We emphasize that in practice, formal security guarantees should not only be reduced to the adversarial hash power or stake threshold. To give a recent example, a bug in a very common Ethereum client implementation recently caused the check-pointed chain to stop finalizing blocks for around an hour [48], thus inducing a state where liveness was not guaranteed in the safety-prioritizing chain. Nevertheless, as a consequence of the dual-ledger constructions, the dynamically available chain kept growing with transactions being included, so Ethereum did not suffer a full liveness issue [46]. After the issue was resolved the network recovered to its normal state, indicating a high degree of resilience [46]. This incident suggests that analyzing the levels of decentralization and setting them in relationship with the tolerable faulty thresholds in the employed consensus mechanism is not only critical in the distribution of computational power or stake but also on other layers [57]. Finally, consensus designs for permissionless blockchains do not only impact formal security properties but account for and balance also other important issues, such as economic security aspects (including long-term (de-)centralization tendencies) and performance. A broader consideration of PoW and PoS is, therefore, required for a holistic perspective of security in permissionless blockchain designs.

ACKNOWLEDGMENTS

This research was funded in part by the Luxembourg National Research Fund (FNR) through the PABLO project (grant reference 16326754), the FiReSpARX Project (grant reference 14783405), by PayPal, grant reference “P17/IS/13342933/PayPal-FNR/Chair in DFS/Gilbert Fridgen” (PEARL), and by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the project “Fraunhofer Blockchain Center (20-3066-2-6-14)”. For the purpose of open access, and in fulfillment of the obligations arising from the FNR grant agreement, the authors have applied a Creative

Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission. We also thank Joaquín D. Fernández, Orestis Papageorgiou, and Nadia Pocher for their valuable feedback.

REFERENCES

- [1] Bowen Alpern and Fred B Schneider. 1987. Recognizing safety and liveness. *Distributed Computing* 2, 3 (Sept. 1987), 117–126.
- [2] Nick Arnosti and S Matthew Weinberg. 2022. Bitcoin: A natural oligopoly. *Management Science* 68, 7 (Jan. 2022), 4755–4771.
- [3] Christian Badertscher, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. 2017. Bitcoin as a transaction ledger: a composable treatment. In *Advances in Cryptology – CRYPTO 2017*. Vol. 10401. Springer International Publishing, Cham, 324–356.
- [4] Nathalie Bertrand, Vincent Gramoli, Igor Konnov, Marijana Lazić, Pierre Tholomiat, and Josef Widder. 2022. Holistic verification of blockchain consensus. In *36th International Symposium on Distributed Computing*, Vol. 246. ACM, New York, NY, USA, 10:1–10:24.
- [5] Pearl Brereton, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. 2007. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software* 80, 4 (April 2007), 571–583.
- [6] Eric A. Brewer. 2000. Towards robust distributed systems. In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*. ACM, New York, NY, USA, 7.
- [7] Vitalik Buterin, Daniel Reijnders, Stefanos Leonardos, and Georgios Piliouras. 2019. Incentives in Ethereum’s hybrid Casper protocol. In *IEEE International Conference on Blockchain and Cryptocurrency*. IEEE, Seoul, Korea (South), 236–244.
- [8] Bert-Jan Butijn, Damian A. Tamburri, and Willem-Jan van den Heuvel. 2020. Blockchains: A systematic multivocal literature review. *Comput. Surveys* 53 (July 2020), 37 pages. Issue 3.
- [9] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. 2007. Universally composable security with global setup. In *Theory of Cryptography: 4th Theory of Cryptography Conference*. Springer, Amsterdam, The Netherlands, 61–85.
- [10] CCRI. 2022. The Merge: Implications on the electricity consumption and carbon footprint of the Ethereum network. <https://carbon-ratings.com/dl/eth-report-2022>
- [11] Evangelos Deirmentzoglou, Georgios Papakyriakopoulos, and Constantinos Patsakis. 2019. A survey on long-range attacks for proof of stake protocols. *IEEE Access* 7 (2019), 28712–28725.
- [12] Amir Dembo, Sreeram Kannan, Ertem Nusret Tas, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. 2020. Everything is a race and Nakamoto always wins. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, USA, 859–878.
- [13] D. Dolev and H. R. Strong. 1983. Authenticated algorithms for Byzantine agreement. *SIAM J. Comput.* 12, 4 (1983), 656–666.
- [14] John R Douceur. 2002. The sybil attack. In *Peer-to-Peer Systems: First International Workshop*. Springer, Springer, Berlin, Heidelberg, 251–260.
- [15] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. 1988. Consensus in the presence of partial synchrony. *J. ACM* 35, 2 (April 1988), 288–323.
- [16] Cynthia Dwork and Moni Naor. 1992. Pricing via processing or combatting junk mail. In *Advances in Cryptology – CRYPTO ’92*. Springer, Berlin, Heidelberg, 139–147.
- [17] Ittay Eyal and Emin Gün Sirer. 2018. Majority is not enough: Bitcoin mining is vulnerable. *Commun. ACM* 61, 7 (June 2018), 95–102.
- [18] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. 1985. Impossibility of distributed consensus with one faulty process. *J. ACM* 32, 2 (April 1985), 374–382.
- [19] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The Bitcoin backbone protocol: analysis and applications. In *Advances in Cryptology - EUROCRYPT*. Springer, Berlin, Heidelberg, 281–310.
- [20] Peter Gaži, Aggelos Kiayias, and Alexander Russell. 2020. Tight consistency bounds for Bitcoin. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, USA, 819–838.
- [21] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the security and performance of proof of work blockchains. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, USA, 3–16.
- [22] Seth Gilbert and Nancy Lynch. 2002. Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant Web services. *SIGACT News* 33, 2 (June 2002), 51–59.
- [23] Mike Graf, Daniel Rausch, Viktoria Ronge, Christoph Egger, Ralf Küsters, and Dominique Schröder. 2021. A security framework for distributed ledgers. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, USA, 1043–1064.

- [24] Rachid Guerraoui, Petr Kuznetsov, Matteo Monti, Matej Pavlovič, and Dragos-Adrian Seredinschi. 2019. The consensus number of a cryptocurrency. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*. ACM, New York, NY, USA, 307–316.
- [25] Tobias Guggenberger, Johannes Sedlmeir, Gilbert Fridgen, and André Luckow. 2022. An in-depth investigation of the performance characteristics of Hyperledger Fabric. *Computers and Industrial Engineering* 173 (Nov. 2022), 108716.
- [26] Huaqun Guo and Xingjie Yu. 2022. A survey on blockchain technology and its security. *Blockchain: Research and Applications* 3, 2 (June 2022), 100067.
- [27] Mahimna Kelkar, Soubhik Deb, and Sreeram Kannan. 2022. Order-fair consensus in the permissionless setting. In *Proceedings of the 9th ACM on ASIA Public-Key Cryptography Workshop*. ACM, New York, NY, USA, 3–14.
- [28] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliyniykov. 2017. Ouroboros: a provably secure proof-of-stake blockchain protocol. In *Advances in Cryptology – CRYPTO 2017*. Vol. 10401. Springer International Publishing, Cham, 357–388.
- [29] Sunny King and Scott Nadal. 2012. PPCoin: Peer-to-peer crypto-currency with proof-of-stake. <https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf>
- [30] Barbara Kitchenham. 2004. *Procedures for performing systematic reviews*. Technical Report. Keele University & NICTA. <https://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf>
- [31] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 51, 1 (Jan. 2009), 7–15.
- [32] Mary C Lacity. 2022. Blockchain: From Bitcoin to the Internet of value and beyond. *Journal of Information Technology* 37, 4 (2022), 326–340.
- [33] L. Lamport. 1977. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering* SE-3, 2 (March 1977), 125–143.
- [34] Leslie Lamport. 1978. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM* 21, 7 (July 1978), 558–565.
- [35] Suhyeon Lee and Seungjoo Kim. 2019. Countering block withholding attack efficiently. In *IEEE Conference on Computer Communications Workshops*. IEEE, Paris, France, 330–335.
- [36] Andrew Lewis-Pye and Tim Roughgarden. 2020. Resource pools and the CAP theorem. <http://arxiv.org/abs/2006.10698>
- [37] Andrew Lewis-Pye and Tim Roughgarden. 2021. How does blockchain security dictate blockchain implementation?. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, USA, 1006–1019.
- [38] Andrew Lewis-Pye and Tim Roughgarden. 2023. Byzantine generals in the permissionless setting. <https://arxiv.org/pdf/2101.07095v7.pdf>
- [39] Elaine Li, Traian Serbanuta, Denisa Diaconescu, Vlad Zamfir, and Grigore Rosu. 2020. Formalizing correct-by-construction Casper in Coq. In *IEEE International Conference on Blockchain and Cryptocurrency*. IEEE, Toronto, ON, Canada, 1–3.
- [40] Jing Li and Dongning Guo. 2019. On analysis of the Bitcoin and Prism backbone protocols in synchronous networks. In *57th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, Monticello, IL, USA, 17–24.
- [41] Wenting Li, Sébastien Andreina, Jens-Matthias Bohli, and Ghassan Karamé. 2017. Securing proof-of-stake blockchain protocols. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Vol. 10436. Springer International Publishing, Cham, 297–315.
- [42] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The honey badger of BFT protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (Vienna Austria)*. ACM, New York, NY, USA, 31–42. <https://doi.org/10.1145/2976749.2978399>
- [43] Satoshi Nakamoto. 2009. Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>
- [44] Ryuya Nakamura, Takayuki Jimba, and Dominik Harz. 2019. Refinement and verification of CBC Casper. In *Crypto Valley Conference on Blockchain Technology*. IEEE, Rotkreuz, Switzerland, 26–38.
- [45] Joachim Neu, Ertem Nusret Tas, and David Tse. 2021. Ebb-and-Flow protocols: A resolution of the availability-finality dilemma. In *IEEE Symposium on Security and Privacy*. IEEE, San Francisco, CA, USA, 446–465.
- [46] Joachim Neu, Ertem Nusret Tas, and David Tse. 2022. The availability-accountability dilemma and its resolution via accountability gadgets. In *Financial Cryptography and Data Security*. Springer International Publishing, Cham, 541–559.
- [47] Joachim Neu, Ertem Nusret Tas, and David Tse. 2022. Two more attacks on proof-of-stake GHOST/Ethereum. In *Proceedings of the ACM Workshop on Developments in Consensus*. ACM, New York, NY, USA, 43–52.
- [48] Margaux Nijkerk. 2023. Ethereum Briefly Stopped Finalizing Transactions. What Happened? <https://www.coindesk.com/tech/2023/05/17/etheriums-loss-of-finality-what-happened/>
- [49] Zhiqiang Ouyang, Jie Shao, and Yifeng Zeng. 2021. PoW and PoS and related applications. In *Proceedings of the International Conference on Electronic Information Engineering and Computer Science*. IEEE, Changchun, China, 59–62.
- [50] Susan Owicki and Leslie Lamport. 1982. Proving liveness properties of concurrent programs. *ACM Transactions on Programming* 4, 3 (July 1982), 455–495.
- [51] Moritz Platt and Peter McBurney. 2023. Sybil in the haystack: A comprehensive review of blockchain consensus mechanisms in search of strong Sybil attack resistance. *Algorithms* 16 (Jan. 2023), 34. Issue 1.
- [52] Gholamreza Ramezan and Cyril Leung. 2020. Analysis of proof-of-work-based blockchains under an adaptive double-spend attack. *IEEE Transactions on Industrial Informatics* 16, 11 (Nov. 2020), 7035–7045.
- [53] Ranvir Rana, Dimitris Karakostas, Sreeram Kannan, Aggelos Kiayias, and Pramod Viswanath. 2022. Optimal bootstrapping of PoW blockchains. In *Proceedings of the 23rd International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. ACM, New York, NY, USA, 231–240.
- [54] Alexander Rieger, Tamara Roth, Johannes Sedlmeir, and Gilbert Fridgen. 2022. We need a broader debate on the sustainability of blockchain. *Joule* 6, 6 (June 2022), 1137–1141.
- [55] Marten Risius and Kai Spohrer. 2017. A blockchain research framework. *Business & Information Systems Engineering* 59, 6 (Dec. 2017), 385–409.
- [56] Ioanid Roşu and Fahad Saleh. 2020. Evolution of shares in a proof-of-stake cryptocurrency. *Management Science* 67, 2 (Nov. 2020), 661–672. <https://doi.org/10.1287/mnsc.2020.3791>
- [57] Ashish Rajendra Sai, Jim Buckley, Brian Fitzgerald, and Andrew Le Gear. 2021. Taxonomy of centralization in public blockchain systems: A systematic literature review. *Information Processing & Management* 58, 4 (July 2021), 102584.
- [58] Suryanarayana Sankagiri, Xuechao Wang, Sreeram Kannan, and Pramod Viswanath. 2021. Blockchain CAP theorem allows user-dependent adaptivity and finality. In *Financial Cryptography and Data Security*. Springer, Berlin, Heidelberg, 84–103.
- [59] Fred B. Schneider. 1990. Implementing fault-tolerant services using the state machine approach: A tutorial. *Comput. Surveys* 22, 4 (Dec. 1990), 299–319.
- [60] Johannes Sedlmeir, Hans Ulrich Buhl, Gilbert Fridgen, and Robert Keller. 2020. The energy consumption of blockchain technology: Beyond myth. *Business & Information Systems Engineering* 62 (2020), 599–608. Issue 6.
- [61] Barbara Simons, Jennifer L. Welch, and Nancy Lynch. 2005. An overview of clock synchronization. In *Fault-Tolerant Distributed Computing*. Springer New York, New York, NY, USA, 84–96.
- [62] Arshdeep Singh, Gulshan Kumar, Rahul Saha, Mauro Conti, Mamoun Alazab, and Reji Thomas. 2022. A survey and taxonomy of consensus protocols for blockchains. *Journal of Systems Architecture* 127 (2022), 102503.
- [63] Yonatan Sompolsinsky and Aviv Zohar. 2015. Secure high-rate transaction processing in Bitcoin. In *Financial Cryptography and Data Security*. Springer, Berlin, Heidelberg, 507–527.
- [64] Soren Eller Thomsen and Bas Spitters. 2021. Formalizing nakamoto-style proof of stake. In *IEEE 34th Computer Security Foundations Symposium*. IEEE, Dubrovnik, Croatia, 1–15.
- [65] Oleksandr Vashchuk and Roman Shuwar. 2018. Pros and cons of consensus algorithm proof of stake. Difference in the network safety in proof of work and proof of stake. *Electronics and Information Technologies* 9 (2018), 10.
- [66] Matthew Walck, Ke Wang, and Hyong S. Kim. 2019. TendrilStaller: Block delay attack in Bitcoin. In *IEEE International Conference on Blockchain*. IEEE, Atlanta, GA, USA, 1–9.
- [67] Yongge Wang. 2022. Byzantine fault tolerance for distributed ledgers revisited. *Distributed Ledger Technologies: Research and Practice* 1, 1 (2022), 26 pages.
- [68] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151, 2014 (2014), 1–43.
- [69] Yang Xiao, Ning Zhang, Wenjing Lou, and Y. Thomas Hou. 2020. A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials* 22, 2 (2020), 1432–1465.
- [70] Ashok Kumar Yadav, Karan Singh, Ali H. Amin, Laila Almutairi, Theyab R. Alsenani, and Ali Ahmadian. 2023. A comparative study on consensus mechanism with security threats and future scopes: Blockchain. *Computer Communications* 201 (March 2023), 102–115.
- [71] Runkai Yang, Xiaolin Chang, Jelena Mišić, and Vojislav B. Mišić. 2020. Assessing blockchain selfish mining in an imperfect network: Honest and selfish miner views. *Computers & Security* 97 (Oct. 2020), 101956.
- [72] Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. 2019. HotStuff: BFT consensus with linearity and responsiveness. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*. ACM, New York, NY, USA, 347–356.
- [73] Guangsheng Yu, Xuan Zha, Xu Wang, Wei Ni, Kan Yu, J. Andrew Zhang, and Ren Ping Liu. 2020. A unified analytical model for proof-of-x schemes. *Computers & Security* 96 (Sept. 2020), 101934.