The Power of Random Symmetry-Breaking in Nakamoto Consensus

Lili Su ✉ 🏠
Northeastern University, Boston, MA, USA

Quanquan C. Liu ✉ 🏠
Massachusetts Institute of Technology, Cambridge, MA, USA

Neha Narula ✉ 🏠
Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

Nakamoto consensus underlies the security of many of the world's largest cryptocurrencies, such as Bitcoin and Ethereum. Common lore is that Nakamoto consensus only achieves consistency and liveness under a regime where the difficulty of its underlying mining puzzle is very high, negatively impacting overall throughput and latency. In this work, we study Nakamoto consensus under a wide range of puzzle difficulties, including very easy puzzles. We first analyze an adversary-free setting and show that, surprisingly, the common prefix of the blockchain grows quickly even with easy puzzles. In a setting with adversaries, we provide a small backwards-compatible change to Nakamoto consensus to achieve consistency and liveness with easy puzzles. Our insight relies on a careful choice of *symmetry-breaking strategy*, which was significantly underestimated in prior work. We introduce a new method – *coalescing random walks* – to analyzing the correctness of Nakamoto consensus under the uniformly-at-random symmetry-breaking strategy. This method is more powerful than existing analysis methods that focus on bounding the number of *convergence opportunities*.

2012 ACM Subject Classification Theory of computation → Distributed computing models

Keywords and phrases Nakamoto consensus, Byzantine consensus, blockchain, symmetry-breaking, coalescing random walks

Digital Object Identifier 10.4230/LIPIcs.DISC.2021.39

Related Version *Full Version*: https://arxiv.org/pdf/2108.09604.pdf

1 Introduction

Nakamoto consensus [19], the elegant blockchain protocol that underpins many cryptocurrencies, achieves consensus in a setting where nodes can join and leave the system without getting permission from a centralized authority. Instead of depending on the identity of nodes, it achieves consensus by incorporating computational puzzles called proof-of-work [9] (also known as *mining*) and using a simple longest-chain protocol.[1] Nodes in a network maintain a local copy of an append-only ledger and gossip messages to add to the ledger, collecting many into a block. A block consists of the set of records to add, a pointer to the previous block in the node's local copy of the ledger, and a nonce, which is evidence the node has done proof-of-work, or solved a computational puzzle of sufficient difficulty, dependent on the block. The node then broadcasts its local chain to the network. Honest nodes choose a chain they see with the most proof-of-work to continue building upon.

Previous work defined correctness and liveness in proof-of-work protocols (also referred to as the *Bitcoin backbone*) using three properties: *common-prefix*, *chain-quality*, and *chain-growth* [12, 15, 21]. Informally, common-prefix indicates that any two honest nodes share a

[1] We use "longest chain" to mean the one with the most proof-of-work given difficulty adjustments, not necessarily the one with the most blocks, though without considering difficulty adjustments they are the same.

35th International Symposium on Distributed Computing (DISC 2021).
Editor: Seth Gilbert; Article No. 39; pp. 39:1–39:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

common prefix of blocks, chain-growth is the rate at which the common prefix grows over time, and chain-quality represents the fraction of blocks created by honest nodes in a chain. In previous work, achieving these properties critically relied on the setting of the difficulty factor in the computational puzzles. We express this as p, the probability that any node will solve the puzzle in a given round. Previous work analyzing Nakamoto consensus has shown that for consistency and liveness p should be very small in relation to the expected network delay and the number of nodes [12, 21]. For example, mining difficulty in Bitcoin is set so that the network is only expected to find a puzzle solution roughly once every ten minutes.

Requiring a small p increases block time, removing a parameter for improving transaction throughput. One way to compensate is by increasing block size, which could result in burstier network traffic and longer transaction confirmation times for users. Newer chains which do not use proof-of-work seem to favor short block times, probably because users value a fast first block confirmation: in EOS, blocks are proposed every 500 milliseconds [10] and Algorand aims to achieve block finality in 2.5 seconds [18], whereas in Bitcoin blocks only come out every ten minutes.

Common belief is that larger p fundamentally constrains chain growth (i.e., the growth of the common prefix), even in the absence of an adversary, due to the potential of increased *forking*: nodes will find puzzle solutions (and thus blocks) at the same time; because of the delay in hearing about other nodes' chains nodes will build on different chains, delaying agreement. Another common conjecture, explicitly mentioned in [12], is that the choice of *symmetry-breaking strategies*, or ways honest nodes choose among multiple longest chains, is not relevant to correctness.

In this paper, we show that these common beliefs are incorrect. In particular, we show that when p is beyond the well-studied region even the simple strategy of choosing among chains of equal length randomly fosters chain growth, especially in the absence of adversaries.

Contributions. In this work, we formally analyze Nakamoto consensus under a wide range of p including large p. We confirm previous (informal) analysis that Nakamoto consensus requires small p in the presence of adversaries, but show that surprisingly, it does not in a setting without adversaries, even if $p = 1$ (all nodes mine blocks every round) with a minor change in nodes' symmetry-breaking strategy. Previous work assumed the requirement of *convergence opportunities*, a period when only one honest node mines a block, in order to achieve consistency [17, 21]; we show that in fact convergence opportunities are *not* required for common-prefix and chain growth. With an additional backwards-compatible modification to Nakamoto consensus, we can derive a bound on the chain growth for a wider range of p (including large p) in a setting with adversaries. Our key idea in this modification is to introduce a *verifiable delay function* [5] to prevent the adversaries from extending a chain by multiple blocks in a round. Our analysis is based on a new application of a well-known technique, coalescing random walks. To our knowledge this is the first application of coalescing random walks to analyze the common-prefix and chain quality of Bitcoin and other proof-of-work protocols. We thoroughly analyze Nakamoto consensus with the *uniformly-at-random* symmetry-breaking strategy and discuss different symmetry-breaking strategies including *first-seen*, *lexicographically-first*, and *global-random-coin*.

In summary, our contributions are as follows:

- A new approach for analyzing the confirmation time of the Bitcoin protocol under the uniformly-at-random symmetry-breaking strategy in the adversarial-free setting via *coalescing random walks*. Our analysis works for a new region of p, and shows that previous works' requirement for *convergence opportunities* was unneeded.

- New notions of *adversarial advantages* and *coalescing opportunities* to provide a more general analysis of common-prefix and chain growth in Nakamoto consensus in the presence of adversaries.

Related Work. Proofs-of-work were first put forth by Dwork and Naor [9]. Garay, Kiayias, and Leonardas [12] provided the first thorough analysis of Nakamoto's protocol in a synchronous static setting, introducing the ideas of *common-prefix*, *chain quality* and *chain growth*. Later work [15] extended the analysis to a variable difficulty function. Pass, Seeman, and shelat [21] extended the idea of common-prefix to *future self-consistency*, and provided an analysis of Nakamoto consensus in the semi-synchronous setting with an adaptive adversary. Several additional papers used this notion of future self-consistency [17, 27]. [17, 21] relied on *convergence opportunities*, or rounds where only one node mines a block, to analyze chain growth. In this work we show that convergence opportunities are *not* required for chain growth, and relying on them underestimates chain growth with high p; in the adversary-free setting we show chain growth even with $p = 1$ (no convergence opportunities; all nodes mine a block every round). Other work considered the tradeoffs between chain growth and chain quality [15, 16, 21, 23, 26]; however, to the best of our knowledge, none of these works considered different symmetry breaking strategies to enable faster chain growth while maintaining chain quality. In our paper, we thoroughly explore this domain. Another line of work [11, 25] considers how the uniformly-at-random symmetry breaking strategy affects incentive-compatible selfish mining attacks; our analysis applies to general attacks.

Random walks have been used to analyze the probability of consistency violations in proofs-of-stake protocols [3]; ours is the first work that uses coalescing random walks to analyze the common-prefix and chain quality of Bitcoin and other proof-of-work protocols.

2 Model and Definitions

In this section, we present the specific model we use and briefly describe the Bitcoin cryptosystem. We follow the formalization presented in [15, 17, 21].

Network and Computation Model. Following previous work [12, 14, 15, 21, 24, 27], we consider a synchronous network where nodes send messages in synchronous rounds, i.e., $\Delta = 1$; equivalently, there is a global clock and the time is slotted into equal duration rounds. Each node has identical computing power. Notably, the synchronous rounds assumption is significantly more relaxed than assuming $\Delta = 0$.[2] Our model operates in the *permissionless setting*. This means that any miner can join (or leave) the protocol execution without getting permission from a centralized or distributed authority. For ease of exposition, we assume the number of participants remains n. Our results can be easily generalized to handle perturbation in the population size by a stochastic dominance argument as long as the population size does not deviate too far from n, and the proportion of Byzantine participants does not increase due to the perturbation.

Adversary Model. Throughout this paper, we assume that all Byzantine nodes are controlled by a *probabilistic polynomial time (PPT) adversary* $\mathcal{A}$ that can coordinate the behavior of all such nodes. $\mathcal{A}$ operates in PPT which means they have access to random coins but can only

[2] In fact, the analysis based on Poisson race [2, 20] essentially assumes all mined blocks can be ordered in a globally consistent way, i.e., $\Delta = 0$, which does not hold in our synchronous network model.

use polynomial time to perform computations. At any time during the run of the protocol, $\mathcal{A}$ can corrupt up to b nodes at any point in time where b is a parameter that is an input to the protocol. The corrupted nodes remain corrupted for the remainder of the protocol. Finally, $\mathcal{A}$ cannot modify or delete the messages sent by honest nodes, but can read all messages sent over the network and arbitrarily order the messages received by any honest nodes.

2.1 Bitcoin Cryptosystem

A *blockchain protocol* is a stateful algorithm wherein each node maintains a local version of the blockchain $\mathcal{C}$. Each honest node runs its own homogeneous version of the blockchain protocol. Nodes receive messages from the *environment* $\mathcal{Z}(1^\lambda)$, where λ is the security parameter chosen based on the population size n. The environment is responsible for all the external factors related to a protocol's execution. For example, it provides the value of b to the nodes. Detailed description of the environment can be found in [21].

The protocol begins by having the environment $\mathcal{Z}$ initialize n nodes. The protocol proceeds in synchronous rounds; at each round r, each node receives a message from $\mathcal{Z}$. In each round, an honest node attempts to mine a block containing its message to add to its local chain. We provide formal definitions of the Bitcoin cryptosystem below.

Blocks and Blockchains. A blockchain $\mathcal{C} \triangleq B_0B_1B_2\cdots B_\ell$ for some $\ell \in \mathbb{N}$ is a chain of blocks. Here B_0 is a predetermined *genesis block* that all chains must build from. A *block* B_ℓ, for $\ell \geq 1$, is a triple $B_\ell = \langle s, x, \mathsf{nce}\rangle$, where s, x, $\mathsf{nce} \in \{0,1\}^*$ are three binary strings of arbitrary length. Specifically, s is used to indicate this block's predecessor, x is the text of the block containing the message (e.g. transactions) and other metadata, and nce is a *nonce* chosen by a node.

Proofs-of-Work. The Bitcoin cryptosystem crucially uses nonces as *proofs-of-work* for determining whether a block can be legally added to a chain.[3] Proof-of-work (PoW) is rigorously defined in previous work [12, 14, 15, 21, 24, 27] based on the use of the *random oracle model.*

▶ **Definition 1** (Random Oracle Model). *A random oracle $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$ on input $x \in \{0,1\}^*$ outputs a value selected uniformly at random from $\{0,1\}^\lambda$ if x has never been queried before. Otherwise, it returns the previous value returned when x was queried last.*

▶ **Definition 2** (Bitcoin PoW). *All nodes access a common random oracle $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$. We say a node successfully performs a PoW with* proof *$x \in \{0,1\}^*$ if $\mathcal{H}(x) \leq D$.*

▶ **Definition 3** (Valid Chain). *A blockchain $\mathcal{C} = B_0B_1\cdots B_\ell = B_0\langle s_1, x_1, \mathsf{nce}_1\rangle \cdots \langle s_\ell, x_\ell, \mathsf{nce}_\ell\rangle$ is* valid *with respect to a given puzzle difficulty level $D \in \{1, \cdots, 2^\lambda\}$ if the following hold: (1) $\mathcal{H}(B_0) = s_1$ and $\mathcal{H}(B_{\ell'}) = s_{\ell'+1}$ for $\ell' = 1, \cdots, \ell - 1$; and (2) $\mathcal{H}(B_{\ell'}) \leq D$ for $\ell' = 0, \cdots, \ell$.*

Longest Chain Rule. The length of a valid chain C is the number of blocks it contains. We refer to the local version of the blockchain kept by node i as the local chain at node i, denoted by $\mathcal{C}_i$. In each round r, node i tries to mine a block via solving a PoW puzzle with

[3] Note that in practice, the nonce is effectively concatenated with a miner's public key (included in the *coinbase* transaction) to ensure unique queries. The public key does *not* need to be verified. Importantly, this means that the miner can just generate a (pk, sk) pair on their local computer without the need to verify that identity with a third-party authority.

the specified difficulty D. If a block is successfully mined, then node i extends its local chain with this block and broadcasts its updated local chain to all other nodes in the network, which will be delivered at each node at the beginning of the next round. At the beginning of the next round, before working on PoW, node i updates its local chain to be the longest chain it has seen. If there are many longest chains, node i chooses one of them uniformly at random.

For ease of exposition, henceforth, C_i is referred to the local chain at the end of a round; $C_i(t)$ is the local chain of node i at the end of round t. Equivalent to using the difficulty parameter D, one can instead consider $p \triangleq D/2^{\lambda}$. The notion of p used in lieu of D has been considered in [12, 14, 15, 17, 21, 24] to simplify notation. Henceforth, we will quantify the algorithm performance in terms of p rather than D and λ.

We use the phrase *with overwhelming probability* throughout this paper. *With overwhelming probability* is defined as with probability at least $1 - \frac{1}{\text{poly}(\lambda)^c}$ for any constant $c \geq 1$. We use the phrase *with all but negligible probability in* λ to mean that the probability is upper bounded by some negligible function $\nu(\lambda)$ on λ (defined in Definition 4).

▶ **Definition 4** (Negligible Probability). *A function ν is* negligible *if for every polynomial $p(\cdot)$, there exists an N such that for all integers $n > N$, it holds that $\nu(n) < \frac{1}{p(n)}$. We denote such a function by* negl*. An event that occurs with* negligible probability *occurs with probability* negl(n).

2.1.1 Properties of the Protocol

In this paper, we will analyze the Nakamoto consensus in terms of two characteristics (generalized from definitions in [12, 17, 27]). The *common prefix* is defined as a sub-chain that is a common prefix of the local chains of all honest nodes at the end of a round. The two properties *maximal common prefix* and *maximal inconsistency* are defined intuitively as: the maximal prefix that is the same across all honest chains and the maximal number of blocks in any honest chain that is not shared by all other honest chains, respectively.

▶ **Property 5** (Maximal common-prefix and maximal inconsistency). *Given a collection of chains $\mathcal{C} = \left\{\tilde{C}_1, \cdots, \tilde{C}_m\right\}$ that are kept by honest nodes, the maximal common-prefix of chain set $\mathcal{C}$, denoted by $P_{\mathcal{C}}$, is defined as the longest common-prefix of chains $\tilde{C}_1, \cdots, \tilde{C}_m$. The maximal inconsistency of $\mathcal{C}$, denoted by $I_{\mathcal{C}}$, is defined as*

$$\max_{i:1 \leq i \leq m} \left|\tilde{C}_i - P_{\mathcal{C}}\right|, \tag{1}$$

where $\tilde{C}_i - P_{\mathcal{C}}$ is the sub-chain of $\tilde{C}_i$ after removing the prefix $P_{\mathcal{C}}$ and $|\cdot|$ denotes the length of the chain, i.e., the number of blocks in the chain.

3 Fundamental Limitations of Existing Approaches

To the best of our knowledge, existing work assumes extremely small p. In fact, the seemingly mild *honest majority assumption* in [13, 22] also implicitly assumes small p.

▶ **Proposition 6.** *If the* honest majority assumption *in [13] holds, then $p \leq \frac{n-2b}{2(n-b)^2}$.*

A formal statement of the honest majority assumption and the proof of Proposition 6 can be found in the full version. Note that the upper bound in this proposition is only a necessary condition. Having p satisfy this condition does not guarantee protocol correctness.

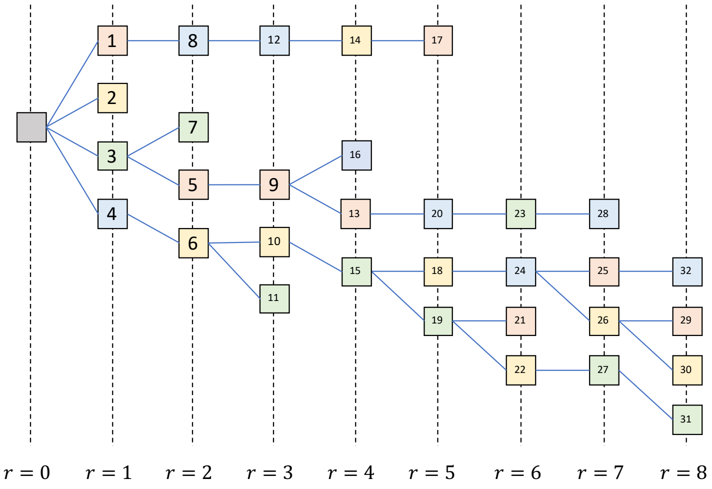

Figure 1 Example growth of a set of chains starting with the genesis block at round $r = 0$. Here, in this example $p = 1$, $n = 4$, and $b = 0$.

▶ Remark 7. Proposition 6 implies that in the vanilla Nakamoto consensus protocol, unless $\frac{b}{n}$ is *non-trivially* bounded above from $\frac{1}{2}$, p needs to be extremely low – even much lower than the commonly believed $\Theta(\frac{1}{n})$. See the full version. for detailed arguments.

To the best of our knowledge, most of the existing analyses focus on bounding the number of “convergence opportunities”, which for $\Delta = 1$ is defined as the number of rounds in which *exactly* one honest node mines a block, and for general Δ, it is defined as the global block mining pattern that consists of (i) a period of Δ rounds where no honest node mines a block, (ii) followed by a round where a single honest player mines a block, (iii) and, finally, another Δ rounds of silence from the honest nodes [17, 21]. Obviously, guaranteeing sufficiently many convergence opportunities necessarily requires p to be small; in the extreme case when $p = 1$ there will be no convergence opportunities at all. An important insight from our results is that *convergence opportunities are not necessary for common-prefix growth*. This is illustrated Fig. 1 which depicts the chain growth when there are 4 honest nodes and $p = 1$. Each node mines a block every round and each is associated with a color. In particular, blocks $1, 5, 9, 13, 17, 21, 25, 29$ are mined by the pink node, blocks $4, 8, 12, 16, 20, 24, 28, 32$ are mined by the blue node, etc. In each round, each node chooses one of the existing longest chains uniformly at random to extend. As shown in Fig. 1, there are no convergence opportunities in any of these 8 rounds and the four nodes never choose the same chain to extend. However, instead of the trivial common prefix (the genesis block) the longest chains at the end of round 8 (the four chains ending with blocks 32, 29, 30, and 31, respectively) share the common prefix $genesis \to 4 \to 6 \to 10 \to 15$. In general, as we show in Section 4, even for the extreme case when $p = 1$, the common prefix of the longest chains still grows as time goes by.

4 Uniformly-at-Random Symmetry-Breaking Strategy

Bitcoin uses the *first-seen* symmetry-breaking strategy; nodes will only switch to a new chain with more proof-of-work than their current longest chain. In this section, we investigate the power of the uniformly-at-random symmetry-breaking strategy, in which each honest node chooses one of its received longest chains uniformly at random to extend upon – independently of other nodes and independently across rounds. We choose to start with the uniformly-at-random strategy because (1) it is easy to implement, especially in a distributed fashion, and (2) despite its simplicity, it is very powerful in fostering chain growth.

For ease of exposition, we first present our results in the adversary-free setting (Sections 4.1 and 4.2) and then in the adversary-prone setting (Section 4.3).

4.1 Warmup: $p = 1$ and Adversary-Free

Even the adversary-free setting (i.e., $b = 0$) is surprisingly non-trivial to analyze. Hence we build insights by first considering the simpler setting where $p = 1$ as a warmup.

▶ **Theorem 8.** *Suppose that $p = 1$ and $b = 0$. Then for any given round index $t \geq 1$, in expectation, the local chains at the honest nodes share a common prefix of length $t + 1 - O(n)$.*

▶ Remark 9. In Theorem 8, the expectation is taken w. r. t. the randomness in the symmetry breaking strategy. Theorem 8 says that large p indeed boosts the growth of the common prefix among the local chains kept by the honest nodes, and that, though temporal forking exists among local chains kept by the honest nodes, such forking can be quickly resolved by repetitive symmetry-breaking across rounds.

The following definition and theorem are useful to see the intuitions of Theorem 8.

▶ **Definition 10** (Coalescing Random Walks [1][4])**.** *In a coalescing random walk, a set of particles make independent random walks on an undirected graph $G = (V, E)$ with self-loops. Whenever one or more particles meet at a vertex, they unite to form a single particle, which then continues the random walk through the graph. We define the* coalescence time*, denoted by C_G, to be the number of steps required before all particles merge into one particle.*

▶ **Theorem 11** ([1, 7])**.** *If $G = (V, E)$ is complete, then $\mathbb{E}[C_G] = O(n)$.*

In the proof of Theorem 8, we build up the connection between the longest chains and the backwards coalescing random walks on complete graphs, and show that the maximal inconsistency among n longest chains turns out to be the same as the number of steps it takes n random walks on the n-complete graph to coalesce into one. Finally, we use the existing results on coalescing random walks to conclude.

Main proof ideas of Theorem 8. We cast our proof insights via an example presented in Fig. 1. In this figure, there are four miners. For ease of exposition, we use the colors *pink*, *yellow*, *green*, and *blue* to represent each of the miners, respectively. As shown in Fig. 1, there are 4 longest chains at the end of round 8 and these chains share a maximal common prefix ending at block 15. The maximal inconsistency of these 4 longest chains is 4; that is, these 4 longest chains are NOT inconsistent with each other until the most recent 4 blocks of each chain. For expository convenience below, instead of using numbers to represent each of the blocks, we use the tuple (color, r) to represent a block that is mined by a certain miner at round r. The maximal inconsistency of the longest chains can be characterized by the coalescing time on complete graphs. To see this, let's consider the four longest chains held by honest miners during round 8 backwards.

Backwards-Chain #1: (**blue**, 8) → (pink, 7) → (**blue**, 6) → (yellow, 5) → (**green**, 4) → (yellow, 3) → (yellow, 2) → (**blue**, 1) → (gray, 0), which can be read as "block (**blue**, 8) is attached to block (pink, 7) which is further attached to block (**blue**, 6) ... attached to the genesis block (gray, 0). "

[4] The original definition given in [1] assumes no self-loops, but its analysis applies to the graphs with self-loops.

Backwards-Chain #2: $(\text{pink}, 8) \rightarrow (\text{yellow}, 7) \rightarrow (\text{pink}, 6) \rightarrow (\textbf{green}, 5) \rightarrow (\textbf{green}, 4) \rightarrow (\text{yellow}, 3) \rightarrow (\text{yellow}, 2) \rightarrow (\textbf{blue}, 1) \rightarrow (\text{gray}, 0)$.
Backwards-Chain #3: $(\text{yellow}, 8) \rightarrow (\text{yellow}, 7) \rightarrow (\text{pink}, 6) \rightarrow (\textbf{green}, 5) \rightarrow (\textbf{green}, 4) \rightarrow (\text{yellow}, 3) \rightarrow (\text{yellow}, 2) \rightarrow (\textbf{blue}, 1) \rightarrow (\text{gray}, 0)$.
Backwards-Chain #4: $(\textbf{green}, 8) \rightarrow (\textbf{green}, 7) \rightarrow (\text{yellow}, 6) \rightarrow (\textbf{green}, 5) \rightarrow (\textbf{green}, 4) \rightarrow (\text{yellow}, 3) \rightarrow (\text{yellow}, 2) \rightarrow (\textbf{blue}, 1) \rightarrow (\text{gray}, 0)$.

Since $p = 1$ and there is no adversary, the number of longest chains received by each honest node at each round is n. Under our symmetry-breaking rule, in each round t, each miner chooses which of the longest chains received at the beginning of round t to extend on uniformly-at-random. Thus, neither the previous history up to round t nor the future block attachment choices after round t affects the choice of the chain extension in round t. Reasoning heuristically[5], we can view each of the backwards-chain as a random walk on a 4-complete graph with vertex set $\{pink, yellow, green, blue\}$. In particular, Backwards-Chain #1 can be viewed as a sample path of a random walk starting at the blue vertex, then moves to the pink vertex, then back to the blue vertex etc., and finally to the blue vertex. Similarly, Backwards-Chains $\#2, \#3$, and $\#4$ can be viewed as the sample paths of three random walks starting at the pink vertex, yellow vertex, and green vertex, respectively. These four random walks (starting at four different vertices) are not completely independent. For any pair of random walks, before they meet, they move on the graph independently of each other; whenever they meet, they move together henceforth. Concretely, backwards-chains 2 and 3 meet at $(\text{yellow}, 7)$ and these chains are identical starting from block $(\text{yellow}, 7)$; this holds similarly for other pairs of backwards chains. Finally, these four backward chains all meet at the block $(\textbf{green}, 4)$ and move together henceforth. Notably, this block is exactly the last block in the maximal common prefix of the four longest chains of round 8. Thus, the maximal inconsistency among the longest chains of round 8 is identical to the number of backwards steps it takes for all these four random walks to coalesce into one. This relation is not a coincidence. It can be shown (detailed in the proof of Theorem 8) that this identity holds for general n. Formal proof of Theorem 8 can be found in Appendix 7.

4.2 General p: Adversary-Free

The analysis for general p is significantly more challenging than that of $p = 1$ in two ways: (1) we need to repeatedly apply coupling arguments; and (2) we need to characterize the coalescence time of a new notion of coalescing random walks (the lazy coalescing random walks), the latter of which could be of independent interest for a broader audience.

▶ **Theorem 12.** *Suppose that $np = \Omega(1)$. If $p < \frac{4\ln 2}{n}$, in expectation, at the end of round t, the local chains at the nodes share a common prefix of length $(1 + (1 - (1 - p)^n)\, t) - O(\frac{1}{npe^{-np}})$. If $p \geq \frac{4\ln 2}{n}$, in expectation, at the end of round t, the local chains at the nodes share a common prefix of length $(1 + (1 - (1 - p)^n)\, t) - O\left(\frac{2np}{(1-2\exp(-\frac{1}{3}np))}\right)$.*

▶ Remark 13. The expression of the common prefix length in Theorem 12 contains two terms with the first term (i.e., $(1 + (1 - (1 - p)^n)\, t)$) being the only term that involves t. Intuitively, from this term, we can read out the common prefix length growth rate w.r.t. t. The second term (which is expression in terms of Big-O notation) can be interpreted as a quantification of the maximal inconsistency of the honest chains.

[5] Formally shown in the proof of Theorem 8 via introducing an auxiliary process.

Now we further interpret these two terms via simplifying the expression using the inequalities $(1-np) \le (1-p)^n \le \exp(-np)$.

(1) When $np = o(1)$, it is true that $(1-p)^n \approx (1-np)$ for large n, which implies that $(1-(1-p)^n)\, t \approx npt = o(t)$, i.e., the common prefix grows at a speed $o(t)$. The maximal inconsistency bound $O(\frac{1}{npe^{-np}})$ is not tight. Nevertheless, via a straightforward calculation, we know that the maximal inconsistency is $O(1)$.

(3) When $np = \omega(1)$, we have $0 \le (1-p)^n \le \exp(-np) \to 0$ as $np \to \infty$. Thus the common-prefix grows at the speed $(1-(1-p)^n)\, t \approx t = \Omega(t)$ with maximal inconsistency $O(np)$ for sufficiently large np.

(4) When $np = c \in (0,1)$, it is true that $(1-p)^n = (1-c/n)^n \to \exp(-c)$ as $n \to \infty$. The common-prefix grows at the speed of $\Theta(t)$ for sufficiently large n and the maximal inconsistency is $O(1)$.

Overall, when np gets larger, the common-prefix growth increases and the maximal inconsistency grows at a much slower rate.

The following definition and lemma are used in proving Theorem 12. This lemma could be of independent interest to a broader audience and its proof can be found in the appendix.

▶ **Definition 14** (Lazy coalescing random walk)**.** *For any fixed $u \in (0,1)$, we say n* particles *are u-lazy coalescing random walks if for each step: with probability $(1-u)$, each particle stays at its current location; with probability u, each particle moves to an adjacent vertex picked uniformly at random. If two or more particles meet at a location, they unite into a single particle and continue the procedure. The* coalescence time *is the same as that in Definition 10.*

▶ **Lemma 15.** *Suppose that G is a complete graph of size $|V| = n_g$ (where $n_g \ge 2$) with self-loops. For any $u \in (0,1)$, the coalescence time of the u-lazy coalescing random walks is $C_G(n_g) = O(n_g/u)$.*

Proof Sketch of Theorem 12. When $p < \frac{4\ln 2}{n}$, we can use Poisson approximation to approximate the distribution of number of blocks in each round. A straightforward calculation shows that the probability of having exactly one block in a round is $np\exp(-np)$. Thus, in expectation, the maximal inconsistency is $O\left(\frac{1}{np\exp(-np)}\right)$. Henceforth, we restrict our attention to the setting where $p \ge \frac{4\ln 2}{n}$ and quantify the expected maximal inconsistency among the longest chains of round t. It is attempting to apply arguments similar to that in the proof of Theorem 8 and derive a bound on the maximal inconsistency via stochastic dominance. However, the obtained bound on the maximal inconsistency is $O(n)$ which could be extremely loose for a wide range of p. Nevertheless, based on the insights obtained in this coarse analysis, we can come up with a much finer-grained analysis and obtain the bound in Theorem 12. Similar to the proof of the special case when $p = 1$, in our fine-grained analysis for general $p \in (0,1)$, we couple the growth of the common prefix in Nakamoto protocols with the coalescing time random walks on complete graphs. The major differences from the proof of $p = 1$ are: (1) instead of the standard coalescing random walks, we need to work with a lazy version of it, formally defined in Definition 14; (2) there is no fixed correspondence between a color and a node – in our proof of general p, the correspondence is round-specific rather than fixed throughout the entire dynamics; (3) there is no bijection between a sample path of the Nakamoto dynamics and that of the backwards coalescing random walks, thus, we need to rely on stochastic dominance to build up the connection of these two dynamics.

4.3 General p: Adversary-Prone

Throughout this section, we assume $p < 1$. In this subsection, we consider adversary-prone systems, i.e., $b > 0$. Simple concentration arguments show that when $bp \geq (1 + 2c)$ for any given $c \in (0, 1)$, using vanilla Nakamoto consensus the chain quality could be near zero. To make larger p feasible, we introduce a new assumption – Assumption 16 – which we then remove in Section 5 by providing a construction that ensures Assumption 16 with all but negligible probability. Specifically, we use a cryptographic tool called a VDF to ensure that over a sufficiently long time window, the corrupt nodes can only collectively extend a chain by more than one block in a round with negligible probability.

▶ **Assumption 16.** *In each round, a chain can be extended by at most 1 block.*

To strengthen the protocol robustness, we make the additional minor modification requiring each honest node to selectively relay chains at the *beginning* of a round.

Selective relay rule. At each honest node i, for each iteration $t \geq 1$: Node i looks at the chains it received in the previous round $t - 1$, and if any of them are longer than its own local longest chain, it not only chooses one of the longest chains to replace its local one, it also broadcasts it to other nodes before it begins mining in round t.

As implied by our proof, this modification can reduce the maximal difference between the lengths of the longest chains kept by the honest nodes and by the corrupt nodes. Intuitively, if the adversary sends two chains of different lengths to two different groups of honest nodes, with the selective relay rule, only the longer chain would survive in this round. Notably, it is possible that none of them survive in this round. Even with the assurance guaranteed by Assumption 16, compared with the adversary-free settings, the analysis for the adversary-prone setting is challenging. This is because the corrupt nodes could deviate from the specified symmetry breaking rule. For example, a corrupt node can choose not to extend its longest chain, or can choose from its set of longest chains in any way that provides advantage. In addition, a corrupt node can hide blocks it has mined from the honest nodes for as long as it wants, or from some subset of the honest nodes during a round.

For simplicity and for technical convenience, we assume that a corrupt node randomly chooses among longest chains that end with an honest block. This assumption is only imposed in the rare event when simultaneously both the adversary has no adversary advantage (see Definition 17) and only honest nodes mine blocks in the most recent nonempty round.

In contrast to the adversary-free setting where the lengths of honest nodes' local chains differ by at most 1, in the presence of an adversary, such difference could be large. To precisely bound this difference, we introduce a random process we call *adversary advantage*:

▶ **Definition 17** (Adversary advantage). *Let* $\{\mathcal{N}(t)\}_{t=0}^{\infty}$ *be the random process defined as*

- $\mathcal{N}(0) = 0$*, and*
- *for* $t \geq 1$*,*

$$\mathcal{N}(t) = \begin{cases} \mathcal{N}(t-1) + 1, & \textit{if only corrupt nodes found blocks in round } t; \\ \max\{\mathcal{N}(t-1) - 1,\ 0\}, & \textit{if only honest nodes found blocks in round } t; \\ \mathcal{N}(t-1), & \textit{otherwise.} \end{cases}$$

Note that the random process $\{\mathcal{N}(t)\}_{t=0}^{\infty}$ is independent of the adversarial behaviors of the corrupt nodes. To make the discussion concrete, we introduce the following definition.

▶ **Definition 18.** *The length of the longest chains kept by the honest nodes* **at round** t *is defined as the length of the longest local chains kept by honest nodes* at the end *of round* t*.*

▶ **Lemma 19.** *For any $t \geq 1$, at the end of round t, the length of the longest chains kept by the adversary – henceforth referred to as an adversarial longest chain of round t – is at most $\mathcal{N}(t)$ longer than the length of a chain kept by an honest node.*

Proof of Lemma 19 can be found in the full version. From its proof, we can deduce an attacking strategy of the adversary that meets the upper bound in Lemma 19. The following lazy random walk, referred to as *coalescing opportunities*, is important in our analysis. It can also be used to quantify the chain quality.

▶ **Definition 20.** *Let $t_1, t_2, \cdots$ be the rounds in which at least one node mines a block with the understanding that $t_0 = 0$. Let $\mathcal{J}(m)$ be a random walk defined as*

$$\mathcal{J}(m) = \begin{cases} 0, & \textit{if } m = 0; \\ \mathcal{J}(m-1) + 1, & \textit{if only honest nodes mine a block during round } t_k; \\ \mathcal{J}(m-1) - 1, & \textit{if only corrupt nodes mine a block during round } t_k; \\ \mathcal{J}(m-1), & \textit{otherwise.} \end{cases}$$

▶ Remark 21. A couple of interesting facts on the coalescing opportunities dynamics are: Among the most recent m blocks in a longest chain, there are at least $\mathcal{J}(m)$ blocks mined by the honest nodes. In addition, regardless of the behaviors of the adversary, for any two longest chains, there are at least $\mathcal{J}(m)$ block positions each of which has non-zero probability of being in the common prefix of these two chains.

Let $p_{+1} = \mathbb{P}\{\mathcal{J}(m) = \mathcal{J}(m-1) + 1\}$ and $p_{-1} = \mathbb{P}\{\mathcal{J}(m) = \mathcal{J}(m-1) - 1\}$, i.e., p_{+1} (resp. p_{-1}) is the probability for $\mathcal{J}(m)$ to move up (resp. down) by 1. We have

$$p_{+1} = \frac{(1-p)^b \left(1 - (1-p)^{n-b}\right)}{1 - (1-p)^n} \quad \text{and} \quad p_{-1} = \frac{\left(1 - (1-p)^b\right)(1-p)^{n-b}}{1 - (1-p)^n}. \tag{2}$$

It is easy to see that when $b > \frac{1}{2}n$, it holds that $p_{+1} > p_{-1}$. For ease of exposition, let $p^* = \mathbb{P}\{\mathcal{J}(t) \neq \mathcal{J}(t-1)\} = p_{+1} + p_{-1}$.

▶ **Lemma 22.** *With probability at least $\left(1 - \exp\left(-\frac{(p_{+1}-p_{-1})^2 M}{16p^*}\right) - \exp\left(-\frac{(p^*)^2 M}{2}\right)\right)$, it holds that $\mathcal{J}(M) \geq \frac{(p_{+1}-p_{-1})M}{4}$.*

Lemma 22 gives a high probability lower bound on the number of coalescing opportunities during M nonempty rounds.

▶ **Theorem 23.** *For any given $T \geq 1$ and $M \geq \frac{4}{\beta(p_{+1}-p_{-1})}$ where $\beta = \frac{(n-b)p}{2(3np)^2}$, at the end of round T, with probability at least*

$$1 - \exp\left(-\frac{(p^*)^2 M}{2}\right) - \exp\left(-\frac{(p_{+1}-p_{-1})^2 M}{16p^*}\right) - \frac{2}{\beta} \exp\left(-\frac{1}{2}(n-b)\right)$$

over the randomness in the block mining, the expected maximal inconsistency among a given pair of honest nodes is less than M, where the expectation is taken over the randomness in the symmetry breaking.

▶ Remark 24. It is worth noting that $\beta = \frac{(n-b)p}{2(3np)^2} = \frac{1}{18}\frac{(n-b)}{n}\frac{1}{np}$, i.e., β is a function of the fraction of honest nodes and the total mining power of the nodes in the system.

Suppose that $n \ge 2\log\frac{4}{\epsilon\beta}$ for any given $\epsilon \in (0,1)$. Let

$$M^* = \max\left\{\frac{4\log 1/\epsilon}{(p^*)^2}, \frac{4}{\beta(p_{+1}-p_{-1})}, \frac{16p^*}{(p_{+1}-p_{-1})^2}\log\frac{4}{\epsilon}\right\}.$$

From Theorem 23, we know that with probability at least $1-\epsilon$, the maximal inconsistency is less than M^*. Roughly speaking, when b gets smaller, M^* mainly gets smaller.

Proof of Theorem 23. We use N_t to denote the number of blocks generated during round t and associate each node with a distinct color in $\{c_1, \cdots, c_n\}$. If node i mines a block during round t, we use (c_i, t) to denote this block. The genesis block is denoted as $(c_1, 0)$. Recall that the blocks mined during round t are collectively referred to as the block layer t. As the randomness in the block generation (i.e., puzzle solving of individual nodes) is independent of the adversarial behaviors of the corrupt nodes and is independent of which chain an honest node chooses to extend, we consider the auxiliary process wherein the nodes mine blocks for the first T rounds, and then the corrupt nodes and honest nodes sequentially decide on block attachments. Let $\{i_1, \cdots, i_K\}$ be the set of rounds such that $N_{i_k} \neq 0$ for each $i_k \in \{i_1, \cdots, i_K\}$. Let j_1 and j_2 be any two honest nodes whose chains at the end of round T are denoted by $C_1(T)$ and $C_2(T)$, respectively. For each of these chains, we can read off a sequence of colors

$$\text{for Chain } C_1(T): \ c_1c(1,2)c(1,3)\cdots c(1,\ell_1), \quad \text{and}$$
$$\text{for Chain } C_2(T): \ c_1c(2,2)c(2,3)\cdots c(2,\ell_2),$$

where ℓ_1 and ℓ_2, respectively, are the lengths of chains $C_1(T)$ and $C_2(T)$, c_1 is the color of the genesis block, $c(1,k)$ for $k \in \{2, \cdots, \ell_1\}$ is the color of the k–th block in $C_1(T)$ and $c(2,k)$ for $k \in \{2, \cdots, \ell_2\}$ is the color of the k–th block in $C_2(T)$. If $\ell_1 \neq \ell_2$, without loss of generality, we consider the case that $\ell_1 < \ell_2$; the other case can be handled similarly. We augment the color sequence $c_1c(1,2)c(1,3)\cdots c(1,\ell_1)$ to the length ℓ_2 sequence as

$$c_1c(1,2)c(1,3)\cdots c(1,\ell_1)c(1,\ell_1+1)\cdots c(1,\ell_2),$$

by setting $c(1,k) = c_0$ for $k = \ell_1+1, \cdots, \ell_2$ where $c_0 \notin \{c_1, \cdots, c_n\}$ is a special color that never shows up in a real block. It is easy to see that $C_1(T)$ and $C_2(T)$ *start* to be inconsistent at their k-th block if and only if $c(1,k') \neq c(2,k')$ for each $k' \in \{k, \cdots, \ell_2\}$. Let $\{i_{h_1}, \cdots, i_{h_R}\} \subseteq \{i_1, \cdots, i_K\}$ such that for each $i_{h_r} \in \{i_{h_1}, \cdots, i_{h_R}\}$ it holds that

- Only honest nodes successfully mined blocks;
- $\mathcal{N}(i_{h_r-1}) = 0$.

For ease of exposition, we refer to each of i_{h_r} as *a coalescing opportunity.* Recall that each of the honest nodes extends one of the longest chains it receives. By Lemma 19, we know that each of $C_1(T)$ and $C_2(T)$ contains a block generated during round i_{h_r}. Let (c_1', i_{h_r}) and (c_2', i_{h_r}) be the blocks included in $C_1(T)$ and $C_2(T)$, respectively. If (c_1', i_{h_r}) is in the k-th position in $C_1(T)$, then (c_2', i_{h_r}) is also in the k-th position in $C_2(T)$. For each i_{h_r}, we denote the set of chains (including the forwarded chains) received by j_1 and j_2 at round i_{h_r}, denoted by $\mathcal{C}_1^r$ and $\mathcal{C}_2^r$. Since the adversary can hide chains to a selective group of honest nodes, $\mathcal{C}_1^r$ and $\mathcal{C}_2^r$ could be different. The probability of j_1 and j_2 extending the same chain at round i_{h_r} is

$$\frac{|\mathcal{C}_1^r \cap \mathcal{C}_2^r|}{|\mathcal{C}_1^r|\,|\mathcal{C}_2^r|} \ge \frac{\mathrm{NB}(i_{h_r-1})}{\Big(\mathrm{NB}(i_{h_r-1}) + \mathrm{AB}(i_{h_r-1}) + \mathrm{AB}(\widetilde{i_{h_r-1}})\Big)^2} \tag{3}$$

where the inequality follows from Lemma 33 of the the full version. By Lemma 22, we know that in the M non-empty block layers that are most recent to round T,

$$R \geq \mathcal{J}(M) \geq \frac{(p_{+1} - p_{-1})M}{4}$$

holds with probability at least $\left(1 - \exp\left(-\frac{(p^*)^2 M}{2}\right) - \exp\left(-\frac{(p_{+1}-p_{-1})^2 M}{16 p^*}\right)\right)$. In addition, it can be shown that for each of the r ensured by Lemma 22 we have

$$\max\{|\mathcal{C}_1^r|, |\mathcal{C}_2^r|\} \leq \mathrm{NB}(i_{h_r-1}) + \mathrm{AB}(i_{h_r-1}) + \mathrm{AB}(\widetilde{i_{h_r-1}})\mathbb{1}\{\mathrm{AB}(i_{h_r-1}) = 0\}.$$

For any i_k, let X_k be the number of blocks mined by the honest nodes during round i_k such that $X_k \neq 0$. Using conditioning and Hoeffding's inequality, the following holds with probability at least $\left(1 - 2\exp\left(-\frac{1}{2}(n-b)\right)\right)$,

$$X_k \geq \frac{1}{2}(n-b)p \;\text{ and }\; X_k + Y_k + Y_{k-1}\mathbb{1}\{Y_k = 0\} \leq 3np,$$

which implies that $\frac{X_k}{X_k + Y_k + Y_{k-1}\mathbb{1}\{Y_k=0\}} \geq \frac{(n-b)p}{2(3np)^2} \triangleq \beta$. On average over the random symmetry breaking, it takes at most $1/\beta$ coalescing opportunities backwards for chains $C_1(T)$ and $C_2(T)$ to coalesce into one. Thus, we need $\frac{(p_{+1}-p_{-1})M}{4} \geq \frac{1}{\beta}$. ◀

5 VDF-Based Scheme

In this section, we present a scheme to ensure Assumption 16. The key cryptographic tool we use in the following scheme is the construction of the *verifiable delay function*, $\mathcal{F}(x)$, which we define informally below. Please refer to [4] for the formal definition (also defined formally in the full version of our paper).

▶ **Definition 25** (Verifiable Delay Function (informal)). *Let λ be our security parameter. There exists a function $\mathcal{F}$ with difficulty $X = O(poly(\lambda))$ where the output $y \leftarrow \mathcal{F}(x)$ (where $x \in \{0,1\}^\lambda$) cannot be computed in less than X sequential computation steps, even provided $poly(\lambda)$ parallel processors, with probability at least $1 - \mathsf{negl}(\lambda)$. The VDF output can be verified, quickly, in $O(\log(X))$ time.*

We set the difficulty of the VDF to the duration of a round; in other words, the difficulty is set such that the VDF produces exactly one output at the end of each round. We amend default Nakamoto consensus by adding the following procedure. We believe this could be added in a backwards-compatible way to existing Nakamoto implementations, like Bitcoin. Backwards-compatibility is desirable in decentralized networks because it means that a majority of the network can upgrade to the new protocol and non-upgraded nodes can still verify blocks and execute transactions. Below we describe a scheme that, when added to Nakomoto consensus, assures Assumption 16. The proof of the following theorem is in the full version of our paper.

▶ **Theorem 26.** *Assumption 16 is satisfied by our VDF-based scheme.*

VDF-Scheme Overview. The VDF-scheme works intuitively as follows. We number the rounds beginning with round 0. All nodes have the genesis block B_0 in their local chains in round 0 and starting mining blocks in round 1. In round 0, the VDF output is computed using 0 as the input. During each round $j > 0$, each node computes a VDF output, y_j, (using $\mathcal{F}$) for the current round j where the input to $\mathcal{F}$ is the output of the VDF, y_{j-1}, from the previous round concatenated with the round number, j. Both inputs are necessary; the output of the VDF from the previous round ensures that we cannot compute the VDF output for this round until we have obtained the output for the previous round, and the round number is necessary to ensure that the output is *not* used for a future round. Once the VDF output is computed, each honest node attempts to mine a block using the VDF output as part of the input to the mining attempt. This also ensures that the block generation rate of honest nodes is upper bounded by np. Then, each node which successfully mines a block sends the new chain to all other nodes.

All honest nodes verify that each chain satisfies two conditions:

1. Let $o_1, \dots, o_\ell$ be the VDF outputs contained in blocks $B_1, \dots, B_\ell$, respectively, of a chain C (the genesis block does not contain a VDF output). Let $r_1, \dots, r_\ell$ be the rounds where $o_1, \dots, o_\ell$ were computed, respectively. Then, $r_1 < \cdots < r_{\ell-1} < r_\ell$.
2. o_i is the VDF output computed from round $r_i \geq i - 1$.

The honest nodes also check all proofs included in the chains, confirming that the VDF outputs are correctly computed and the blocks are correctly mined using the VDF outputs. An honest node discards any chain which does not pass verification.

Pseudocode. The precise pseudocode of our VDF-based scheme is given below. Using $\mathcal{F}$, each honest node i performs the following:

1. Initially, all honest nodes use input 0 at the start of the protocol to obtain output $y_0 = \mathcal{F}(0)$ for round 0.
2. Let $d_j = \mathcal{F}(y_{j-1})$ be the output of the VDF for round j and $y_j = d_{j-1}|j$.[6] i stores y_j.
3. When i mines a block B_j, i includes the output $y_{j-1} = d_{j-1}|j$ from the previous round in B_j, ie. B_j is mined with y_{j-1} as part of the input.
4. Each node which successfully mines a block adds the mined block to its local chain. Then, it broadcasts its local chain to all other nodes.
5. For each longest chain received, each node verifies the following:
   a. Let $o_1, \dots, o_\ell$ be the VDF outputs stored in each block in order starting with the first block and ending with the ℓ-th block. Let $r_1, \dots, r_\ell$ be the rounds associated with the VDF output. Then, $r_\ell > r_{\ell-1} > \cdots > r_1$.
   b. The k-th block in the chain (starting from the genesis block) is mined using $y_{k'}$ from round $k' \geq k - 1$.
   c. The proofs of the VDF output and the mining output are correct, i.e. the block is correctly mined using the corresponding VDF output.
6. If i receives a chain where more than one block in the chain is mined with the same y_j (for any j smaller than the current round), the node discards the chain.
7. At the end of round j, i sets $y_{j+1} \leftarrow \mathcal{F}(y_j)|j+1$ and begins computing the next value $\mathcal{F}(y_{j+1})$ using y_{j+1} as input.

Due to space constraints, we do not include the proof of Theorem 26; please find the full proofs in the full version of our paper. However, the intuition for our proof is straightforward. Items 5a and 5b ensure that no chain accepted by an honest node contains more than one block per VDF output. Setting the difficulty of the VDF to the duration of the round ensures that at most one VDF output is produced during a round. Together, these two observations prove Theorem 26, namely, that any chain held by an honest node can be extended by at most one block each round.

[6] Here, $a|b$ is the commonly used notation indicating concatenation between a and b.

6 Discussion

Validation and Communication Costs. A higher p means a faster block rate and thus more blocks. The validation and bandwidth complexity of Nakamoto protocols are proportional to block size and the number of blocks that are mined, since each miner validates and then communicates every mined block to all other miners (in practice, nodes do not necessarily gossip shorter chains, and taking advantage of nodes' memory overlap can help reduce block transfer size [8]). One needs to determine the optimal value of p that trades off validation and bandwidth complexity and chain growth. This work expands the space of p to consider.

Other Symmetry-Breaking Strategies. Here we consider three other symmetry-breaking strategies with high p. *First-seen* is where all honest nodes take the first chain out of the longest-length chains they see, and *lexicographically-first* is where honest nodes take the lexicographically-first chain of the set of longest chains according to some predetermined ordering, for example alphabetically. Intuitively, the adversary can control the network and thus cause different honest nodes to see different chains of the same length first for first-seen, impacting common-prefix, or grind on blocks to always produce the lowest lexicographically-ordered chain for lexicographically-first, impacting chain-quality. A third strategy is to use a *global-random-coin*: Suppose that all nodes have access to a permutation oracle $\mathcal{P}$ that returns a permutation sampled uniformly at random of a number of elements passed into it *where any subset of elements obey the same partial ordering.* With $\mathcal{P}$ symmetry-breaking is trivial since all honest nodes will agree on the result of the coin flip. Furthermore, if the coin is fair, then the number of honest blocks added to the chain is proportional to the fraction of honest nodes. However, in reality, it is difficult and oftentimes infeasible to ensure such a strong guarantee.

Conclusion. In this work we show that unlike previously thought, convergence opportunities are not necessary to make chain progress. We use *coalescing random walks* to analyze the correctness of Nakamoto consensus under a regime of puzzle difficulty previously thought to be untenable, expanding the space of p for protocol designers.

References

1 David Aldous and Jim Fill. Reversible markov chains and random walks on graphs, 2002.

2 Vivek Bagaria, Sreeram Kannan, David Tse, Giulia Fanti, and Pramod Viswanath. Prism: Deconstructing the blockchain to approach physical limits. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 585–602, 2019.

3 Erica Blum, Aggelos Kiayias, Cristopher Moore, Saad Quader, and Alexander Russell. *The Combinatorics of the Longest-Chain Rule: Linear Consistency for Proof-of-Stake Blockchains*, pages 1135–1154. SIAM, 2020. doi:10.1137/1.9781611975994.69.

4 Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, pages 757–788, 2018. doi:10.1007/978-3-319-96884-1_25.

5 Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. Cryptology ePrint Archive, Report 2018/601, 2018. URL: https://eprint.iacr.org/2018/601.

6 Colin Cooper, Robert Elsasser, Hirotaka Ono, and Tomasz Radzik. Coalescing random walks and voting on connected graphs. *SIAM Journal on Discrete Mathematics*, 27(4):1748–1758, 2013.

7 Colin Cooper, Alan Frieze, and Tomasz Radzik. Multiple random walks in random regular graphs. *SIAM Journal on Discrete Mathematics*, 23(4):1738–1761, 2010.

8 Matt Corallo. Compact block relay, 2016. URL: https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki.

9 Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Annual international cryptology conference*, pages 139–147. Springer, 1992.

10 EOS. v2.0 consensus protocol, 2021. URL: https://developers.eos.io/welcome/v2.0/protocol/consensus_protocol.

11 Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security*, pages 436–454. Springer, 2014.

12 Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 281–310, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

13 Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

14 Juan Garay, Aggelos Kiayias, and Nikos Leonardos. Full analysis of nakamoto consensus in bounded-delay networks. Cryptology ePrint Archive, Report 2020/277, 2020. URL: https://eprint.iacr.org/2020/277.

15 Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol with chains of variable difficulty. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 291–323. Springer, 2017. doi:10.1007/978-3-319-63688-7_10.

16 Aggelos Kiayias and Giorgos Panagiotakos. Speed-security tradeoffs in blockchain protocols. *IACR Cryptol. ePrint Arch.*, 2015:1019, 2015. URL: http://eprint.iacr.org/2015/1019.

17 Lucianna Kiffer, Rajmohan Rajaraman, and abhi shelat. A better method to analyze blockchain consistency. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 729–744, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3243734.3243814.

18 Silvio Micali. Algorand 2021 performance, 2020. URL: https://www.algorand.com/resources/blog/algorand-2021-performance.

19 Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009. URL: http://www.bitcoin.org/bitcoin.pdf.

20 Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system.(2008), 2008.

21 Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 643–673, Cham, 2017. Springer International Publishing.

22 Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 643–673. Springer, 2017.

23 Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '17, page 315–324, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3087801.3087809.

24 Ling Ren. Analysis of nakamoto consensus. *IACR Cryptol. ePrint Arch.*, 2019:943, 2019. URL: https://eprint.iacr.org/2019/943.

25 Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2016.

26 R. Zhang and B. Preneel. Lay down the common metrics: Evaluating proof-of-work consensus protocols' security. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 175–192, 2019. doi:10.1109/SP.2019.00086.

27 Jun Zhao, Jing Tang, Zengxiang Li, Huaxiong Wang, Kwok-Yan Lam, and Kaiping Xue. An analysis of blockchain consistency in asynchronous networks: Deriving a neat bound. In *40th IEEE International Conference on Distributed Computing Systems, ICDCS 2020, Singapore, November 29 - December 1, 2020*, pages 179–189. IEEE, 2020. doi:10.1109/ICDCS47774.2020.00039.

7 Proof of Theorem 8

Proof of Theorem 8. We formalize the arguments of the main proof ideas in Section 4.1. Let $\{c_1, \cdots, c_n\}$ be a set of n different colors. We associate each node in the system with a color. We use (c_i, t) to denote the block generated by honest node i during round t and $(c_0, 0)$ to denote the genesis block. We use $(c_i, t) \to (c_{i'}, t-1)$ to denote the event that block (c_i, t) is attached to block $(c_{i'}, t-1)$, which occurs with probability $\frac{1}{n}$ under our symmetry-breaking rule. To quantify the maximal inconsistency of the longest chains of round T, we consider the following auxiliary random process. It can be easily shown that there is a bijection between the sample paths of the Bitcoin blockchain protocol and the sample paths of this auxiliary process, and that the auxiliary process and the original blockchain protocol with random symmetry breaking have the same probability distribution.

Auxiliary random procedure: For any given $T \geq 1$, do the following:

(i) Let each color generate a block for each of the rounds in $\{1, 2, \cdots, T\}$;

(ii) Attach each of the block $(c_i, 1)$ for $i = 1, \cdots, n$ to the genesis block $(c_0, 0)$;

(iii) For each $t \geq 2$ and each (c_i, t), attach it to one of the blocks $\{(c_i, t-1), i = 1, \cdots, n\}$ uniformly at random (i.e., with probability $1/n$).

Connecting to coalescing random walks: Here, we formally quantify the connection between the maximal inconsistency among the longest chains of round T with the coalescing time of n random walks on an n-complete graph. Since $p = 1$ and there is no adversary, the number of longest chains received by each honest node at each round is n. Let $C(T, c_1), \cdots, C(T, c_n)$ be the n longest chains of round T ending with blocks $(c_1, T), \cdots, (c_n, T)$, respectively. We first show that each of these n chains can be coupled with a random walk on the n-complete graph. Without loss of generality, let's consider $C(T, c_1)$ which can be expanded as

$$C(T,c_1) := (c_0,0) \leftarrow (c_{i_1},1) \leftarrow \cdots \leftarrow \left(c_{i_{t-1}},t-1\right) \leftarrow (c_{i_t},t) \leftarrow \cdots \leftarrow \left(c_{i_{T-1}},T-1\right) \leftarrow (c_1,T), \quad (4)$$

where c_t is the color of the $(t+1)$-th block in the chain. Note that the chain $C(T, c_1)$ is random because the sequence of block colors $c_0 c_{i_1} \cdots c_{i_{t-1}} c_{i_t} \cdots c_{i_{T-1}} c_1$ is random. Moreover, the randomness in $C(T, c_1)$ is fully captured in the randomness of the block colors. We have

$$\begin{aligned}
&\mathbb{P}\left\{C(T,c_1) = (c_0,0) \leftarrow (c_{i_1},1) \leftarrow \cdots \leftarrow \left(c_{i_{t-1}},t-1\right) \leftarrow (c_{i_t},t) \leftarrow \cdots \leftarrow \left(c_{i_{T-1}},T-1\right) \leftarrow (c_1,T)\right\} \\
&\overset{(a)}{=} \mathbb{P}\left\{(c_0, 0) \leftarrow (c_{i_1}, 1)\right\} \prod_{t=2}^{T} \mathbb{P}\left\{\left(c_{i_{t-1}}, t-1\right) \leftarrow (c_{i_t}, t)\right\} \\
&= \prod_{t=2}^{T} \mathbb{P}\left\{\left(c_{i_{t-1}}, t-1\right) \leftarrow (c_{i_t}, t)\right\},
\end{aligned}$$

where the last equality is true as $\mathbb{P}\{(c_0, 0) \leftarrow (c_{i_1}, 1)\} = 1$, and the equality (a) holds because under our symmetry-breaking rule, neither the previous history up to round t nor the future block attachment choices after round t affects the choice of the chain extension in round t.

Moreover, the probability of any realization of the color sequence $c_0 c_{i_1} \cdots c_{i_{t-1}} c_{i_t} \cdots c_{i_{T-1}} c_1$ (i.e., a sample path on the block colors in Bitcoin) is $\left(\frac{1}{n}\right)^{T-1}$. Let's consider the complete graph with vertex set $\{c_1, c_2, \cdots, c_n\}$. Under our symmetry breaking rule, the backwards color sequence $c_1 c_{i_{T-1}} \cdots c_{i_t} c_{i_{t-1}} \cdots c_{i_1}$ (without considering the genesis block) is a random walk on the n-complete graph starting at vertex c_1. Similarly, we can argue that $C(T, c_2), \cdots, C(T, c_n)$ correspond to $n-1$ random walks on the n-complete graphs starting at vertices $c_2, \cdots, c_n$, respectively. As argued in the **main proof ideas** paragraph, these n random walks are not fully independent. In fact, they are coalescing random walks, and their coalescence is exactly the maximal inconsistency among the longest chains of round T.

With the above connection of the longest chain protocol augmented by uniformly-at-random symmetry breaking with coalescing random walks. We conclude by applying Theorem 11. ◀

8 Proof of Lemma 15

Proof of Lemma 15. To characterize the coalescence time, similar to the analysis in [6], for any given $k \in \{1, \cdots, n_g\}$, we construct a larger graph $Q = Q_k = (V_Q, E_Q)$, where $V_Q = V^k$ and two vertices $\boldsymbol{v}, \boldsymbol{w} \in V^k$ if $\{v_1, w_1\}, \cdots, \{v_k, w_k\}$ are edges of G. Let M_k be the time until the first meeting in the original graph G. Let $S \subseteq V_Q$ denote the set of all possible configurations of the locations of the n_g random walks at the first meeting,

$$S_k = \{(v_1, \cdots, v_k) : v_i = v_j \quad \text{for some } 1 \le i < j \le k\}. \tag{5}$$

It is easy to see that there is a direct equivalence between the u-lazy random walks on G and the single u-lazy random walk on Q. Since Q is a complete graph with self-loops, the limiting distribution of lazy random walk on Q is the same as the standard random walk on Q. Let $\boldsymbol{\pi}^Q \in \mathbb{R}^{|V^k|}$ be the stationary distribution of a standard random walk on Q and let $\pi^Q_{S_k} = \sum_{\boldsymbol{v} \in S_k} \pi^Q_{\boldsymbol{v}}$. By [6, Lemma 4], we know that for any $1 \le k \le k^*$ where $k^* \triangleq \max\{2, \log n_g\}$, it holds that

$$\pi^Q_{S_k} \ge \frac{k^2}{8 n_g}.$$

Let $H_{\boldsymbol{v}, S_k}$ denote the hitting time of vertex set S_k starting from vertex v and let

$$H^Q_{\boldsymbol{\pi}}(H_{S_k}) = \sum_{\boldsymbol{v} \in V^k} \boldsymbol{\pi}^Q_{\boldsymbol{v}} H_{\boldsymbol{v}, S_k}$$

denote the expected hitting time of S_k from the stationary distribution $\boldsymbol{\pi}^Q$. From [1, Lemma 2.1] and the fact we can contract the vertex set S_k into one pseudo vertex, similar to [6, proof of Theorem 2], we have that

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\pi}^Q}[H_{S_k}] &= \frac{\sum_{t=0}^{\infty} \left(P^t_{S_k}(S_k) - \boldsymbol{\pi}^Q_{S_t}\right)}{\boldsymbol{\pi}^Q_{S_k}} = \frac{\sum_{t=0}^{\infty} \left((1-u)^t + (1-(1-u)^t)\, \boldsymbol{\pi}^Q_{S_k} - \boldsymbol{\pi}^Q_{S_k}\right)}{\boldsymbol{\pi}^Q_{S_k}} \\ &\le \frac{8 n_g}{k^2} \frac{1}{u} \left(1 - \boldsymbol{\pi}^Q_{S_k}\right) \le \frac{8 n_g}{u k^2}. \end{aligned}$$

In addition, by conditioning on whether the particles stay at their initial locations or not, we have

$$\mathbb{E}[M_k] = (1-u)(1 + \mathbb{E}[M_k]) + u\left(1 + \mathbb{E}_{\boldsymbol{\pi}^Q}(H_{S_k})\right),$$

which implies that

$$\mathbb{E}\left[M_k\right] \leq \frac{1}{u}\left(1+\frac{8n_g}{k^2}\right) = O\left(\frac{n_g}{uk^2}\right).$$

Thus, for any k such that $1 \leq k \leq k^* = \{2, \log n_g\}$, we have

$$\mathbb{E}\left[C_k\right] \leq \sum_{s=2}^{k} \mathbb{E}\left[M_s\right] \leq O(n_g/u).$$

Let $\mathcal{W}_u$ be a lazy random walk on the complete graph G with initial location u. In each round, with probability $(1-u)$, $\mathcal{W}_u$ stays at its current location and with probability u it moves to one of the current neighbors (including self-loops) uniformly at random. Let $\boldsymbol{\pi}^G$ the limiting distribution of the location vertex of $\mathcal{W}_u$. By [6, Eq.(8)], its mixing time is $t_{mix} = \frac{3\log n_g}{\log(1/(1-u))}$, i.e., for any given $u \in V$, when $t \geq \lceil \frac{3\log n_g}{\log(1/(1-u))} \rceil$,

$$\begin{aligned}\|P_u^t - \boldsymbol{\pi}^G\|_1 &= \sum_{v\in V} \left|P_u^t(v) - \boldsymbol{\pi}_v^G\right| \\ &= \left|1-\pi_u^G\right|(1-u)^t + \sum_{v:v\in V, v\neq u} \left|\left(1-(1-u)^t\right)\boldsymbol{\pi}_v^G - \boldsymbol{\pi}_v^G\right| \\ &\leq 2(1-u)^t \leq \frac{2}{n_g^3} \leq \frac{1}{n_g^2}.\end{aligned}$$

Here, with a little abuse of notation, we use P_u^t to denote the distribution of the state of $\mathcal{W}_u$ at round t. Let $t^* = k^* \log n_g \left(k^* t_{mix} + 3\mathbb{E}_{\boldsymbol{\pi}^Q}\left(H_{S_{k^*}}\right)\right)$. Following the arguments in [6, Section 5], we have

$$\begin{aligned}C(n_g) &\leq 4t^* + \mathbb{E}\left[C_{k^*}\right] \\ &\leq 4\log n_g \left(k^* t_{mix} + 3\mathbb{E}_{\boldsymbol{\pi}^Q}\left(H_{S_{k^*}}\right)\right) + O(n_g/u) \\ &\leq \frac{4\log^4 n_g}{\log\frac{1}{1-u}} + 12\log^2 n_g \frac{8n_g}{u\log^2 n_g} + O(n_g/u) \\ &\leq \frac{4\log^4 n_g}{u} + \frac{96n_g}{u} + O(n_g/u) \\ &= O(n_g/u),\end{aligned}$$

where the last inequality follows from $\log 1/(1-u) \geq u$. ◀