Everything is a Race and Nakamoto Always Wins

Amir Dembo
Stanford University
amir@math.stanford.edu

Sreeram Kannan
University of Washington
ksreeram@uw.edu

Ertem Nusret Tas
Stanford University
nusret@stanford.edu

David Tse
Stanford University
dntse@stanford.edu

Pramod Viswanath
University of Illinois
Urbana-Champaign
pramodv@illinois.edu

Xuechao Wang
University of Illinois
Urbana-Champaign
xuechao2@illinois.edu

Ofer Zeitouni
Weizmann Institute of Science
ofer.zeitouni@weizmann.ac.il

ABSTRACT

Nakamoto invented the longest chain protocol, and claimed its security by analyzing the private double-spend attack, a race between the adversary and the honest nodes to grow a longer chain. But is it the worst attack? We answer the question in the affirmative for three classes of longest chain protocols, designed for different consensus models: 1) Nakamoto's original Proof-of-Work protocol; 2) Ouroboros and SnowWhite Proof-of-Stake protocols; 3) Chia Proof-of-Space protocol. As a consequence, exact characterization of the maximum tolerable adversary power is obtained for each protocol as a function of the average block time normalized by the network delay. The security analysis of these protocols is performed in a unified manner by a novel method of reducing all attacks to a race between the adversary and the honest nodes.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**.

KEYWORDS

Proof-of-Work; Proof-of-Stake; Proof-of-Space; Security Analysis

ACM Reference Format:
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 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20), November 9–13, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3372297.3417290

The authors are listed alphabetically. For correspondence on the paper, please contact DT at dntse@stanford.edu.

CCS '20, November 9–13, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7089-9/20/11…$15.00
https://doi.org/10.1145/3372297.3417290

1 INTRODUCTION

1.1 Background

In 2008, Satoshi Nakamoto invented the concept of *blockchains* as a technology for maintaining decentralized ledgers [Nak08]. A core contribution of this work is the *longest chain* protocol, a deceptively simple consensus algorithm. Although invented in the context of Bitcoin and its Proof-of-Work (PoW) setting, the longest chain protocol has been adopted in many blockchain projects, as well as extended to other more energy-efficient settings such as Proof-of-Stake (PoS) (eg. [BPS16], [KRDO17],[DGKR18],[BGK$^+$18],[FZ18]) and Proof-of-Space (PoSpace) (eg. [AAC$^+$17, CP19, PKF$^+$18]).

Used to maintain a ledger for a valued asset in a permissionless environment, the most important property of the longest chain protocol is its *security*: how much resource does an adversary need to attack the protocol and revert transactions already confirmed? Nakamoto analyzed this property by proposing a specific attack: the private double-spend attack (Figure 2(a)). The adversary grows a private chain of blocks in a race to attempt to outpace the public longest chain and thereby replacing it after a block in the public chain becomes k-deep. Let λ_h and λ_a be the rate at which the honest nodes and the adversary mine blocks, proportional to their respective hashing powers. Then it is clear from a law of large numbers argument that if $\lambda_a > \lambda_h$, then the adversary will succeed with high probability no matter how large k is. Conversely, if $\lambda_a < \lambda_h$, the probability of the adversary succeeding decreases exponentially with k. When there is a network delay of Δ between honest nodes, this condition for security becomes:

$$\lambda_a < \lambda_{\text{growth}}(\lambda_h, \Delta), \tag{1}$$

where $\lambda_{\text{growth}}(\lambda_h, \Delta)$ is the growth rate of the honest chain under worst-case forking. In a fully decentralized setting with many honest nodes each having small mining power, [SZ15] calculates this to be $\lambda_{\text{growth}} = \lambda_h/(1 + \lambda_h\Delta)$. If we let β to be the adversary fraction of power, then (1) yields the following condition:

$$\beta < \frac{1-\beta}{1+(1-\beta)\lambda\Delta}. \tag{2}$$

Here, λ is the total mining rate, and $\lambda\Delta$ is the number of blocks mined per network delay. $1/(\lambda\Delta)$ is the block speed normalized by the network delay. Solving (2) at equality gives a security threshold $\beta_{\text{pa}}(\lambda\Delta)$. When $\lambda\Delta$ is small, $\beta_{pa}(\lambda\Delta) \approx 0.5$, and this leads to

Nakamoto's main claim in [Nak08]: the longest chain protocol is secure as long as the adversary has less than 50% of the total hashing power and the mining rate is set to be low. A more aggressive mining rate to speed up the blockchain reduces the security threshold. Hence (2) can be viewed as a tradeoff between security and block speed.

The private double-spend attack is a *specific* attack, and Nakamoto claimed security based on the analysis of this attack alone. But what about other attacks? Are there other worse attacks? A pertinent question after Nakamoto's work is the identification of the *true* security threshold $\beta^*(\lambda\Delta)$ in the face of the *worst* attack. The groundbreaking work [GKL15] first addressed this question by formulating and performing a formal security analysis of the Proof-of-work longest chain protocol. They used a lock-step round-by-round synchronous model, and the analysis was later extended to the more realistic Δ-synchronous model [PSS17]. The results show that when $\lambda\Delta \to 0$, indeed $\beta^*(\lambda\Delta)$ approaches 50%, thus validating Nakamoto's intuition. However, for $\lambda\Delta > 0$, there is a gap between their bounds and the private attack security threshold, and this gap grows when $\lambda\Delta$ grows.

1.2 Main contribution

The main contribution of this work is a new approach to the security analysis of longest chain protocols. This approach is driven by the question of whether the private attack is the worst attack for longest chain protocols in a broad sense. Applying this approach to analyze three classes of longest chain protocols in the Δ−synchronous model[PSS17], we answer this question in the affirmative in all cases: **the *true* security threshold is the same as the private attack threshold**:

$$\beta^*(\lambda\Delta) = \beta_{\text{pa}}(\lambda\Delta) \quad \text{for all } \lambda\Delta \geq 0 \tag{3}$$

(Figure 1). The three classes are: 1) the original Nakamoto PoW protocol; 2) Ouroboros Praos [DGKR18] and SnowWhite [PS17, BPS16] PoS protocols; 3) Chia PoSpace protocol [CP19]. They all use the longest chain rule but differ in how the lotteries for proposing blocks are run. (Figure 4) In the first two protocols, we close the gap between existing bounds and the private attack threshold by identifying the true threshold to be the private attack threshold at all values of $\lambda\Delta$. For Chia, the adversary is potentially very powerful, since at each time, the adversary can mine on every block of the blocktree, and each block provides an independent opportunity for winning the lottery. It was not known to be secure for *any* non-zero fraction of adversary power. (More specifically, while [CP19] proved the chain growth and chain quality properties for the Chia protocol, the crucial common prefix property is missing.) Our result not only says that Chia is secure, but it is secure all the way up to the private attack threshold (although the private attack threshold is smaller for Chia than for the other two classes of protocols due to the increased power of the adversary).

That the true security threshold matches the private attack threshold in all these protocols is not a coincidence. It is due to an intimate connection between the private attack and any general attack. Our approach exposes and exploits this connection by defining two key concepts: *blocktree partitioning* and *Nakamoto blocks*. Through these concepts, we can view *any* attack as a race between adversary and honest chains, not just the private attack.

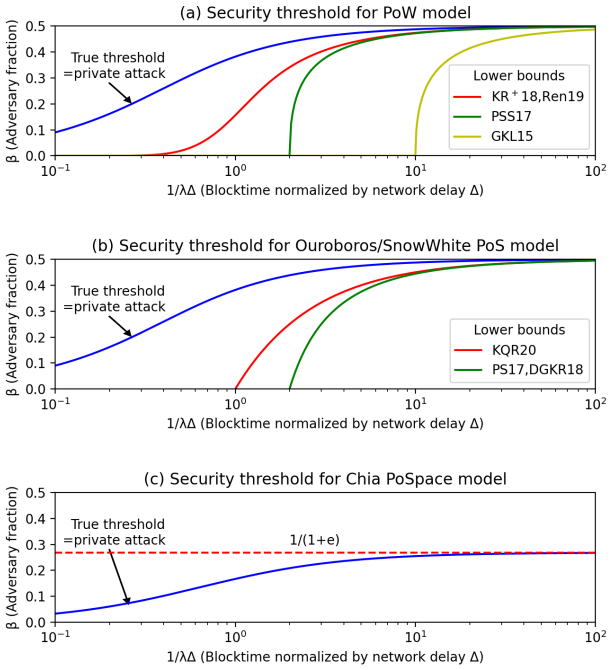

Figure 1: True security threshold as a function of normalized block speed, compared to bounds in the literature. (a) Proof-of-work model; (b) Ouroboros/SnowWhite Proof-of-Stake model; (c) Chia Proof-of-Space model. In (a) and (b), the blue curve represents $\beta^*(\lambda\Delta) = \beta_{\text{pa}}(\lambda\Delta)$; both PoW and PoS have the same (true) security threshold. In (a), the red, green and yellow curves are obtained by solving $\beta = (1-\beta)e^{-2(1-\beta)\lambda\Delta}$, $\beta = (1-\beta)(1-2\lambda\Delta(1-\beta))$ and $\beta = (1-\beta)(1-10\lambda\Delta(1-\beta))$ respectively. In (b), the red and green curves are $(1-\beta)/(1+\lambda\Delta) = 1/2$ and $(1-\beta)(1-\lambda\Delta) = 1/2$ respectively. In (c), the blue curve is the solution of $e\beta = \frac{1-\beta}{1+(1-\beta)\lambda\Delta}$, the true threshold, and also that of private attack. Unlike in (a) and (b), the true threshold does not reach 0.5 when $\lambda\Delta \to 0$, but reach $1/(1+e)$ instead. Note that while in all cases , the true security threshold equals the private attack threshold, the threshold is different for Chia than for the other two.

However, unlike the private attack, a general attack may send many adversary chains to simultaneously race with the honest chain.

The entire blocktree, consisting of both honest and adversary blocks, public or private, is particularly simple under a private attack: it can be partitioned into two chains, one honest and one adversary (Figure 2(a)). In contrast, under a general attack where the adversary can make public blocks at multiple time instances, a much more complex blocktree can emerge (Figure 2(b)). However, what we observe is that by partitioning this more complex tree into sub-trees, each rooted at a honest block and consisting otherwise entirely of adversary blocks, one can view the general attack as initiating *multiple* adversary sub-trees to race with a single fictitious chain consisting of only honest blocks (Figure 3). The growth rate of each of these adversary sub-trees is upper bounded by the growth rate of the adversary chain used in the private attack. Therefore, if the private attack is unsuccessful, we know that the growth

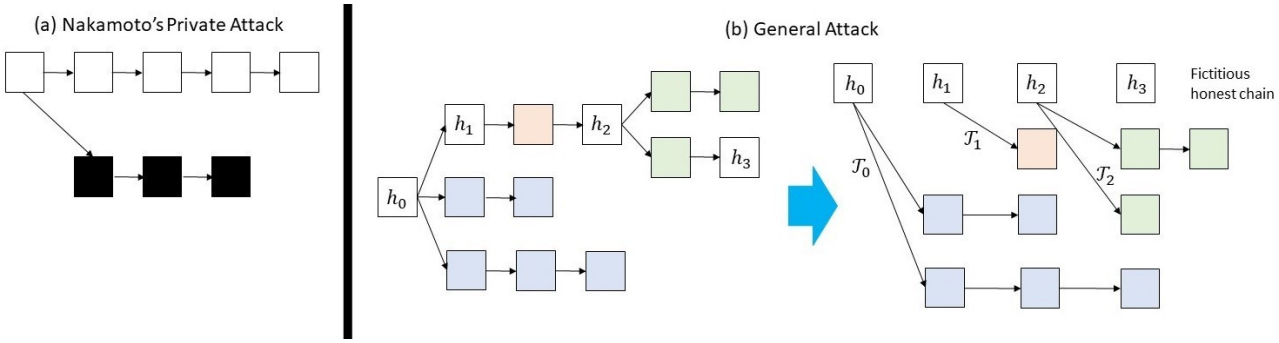

Figure 2: (a)Nakamoto's private attack as a race between a single adversary chain and the honest chain. (b) By blocktree partitioning, a general attack is represented as multiple adversary chains simultaneously racing with a fictitious honest chain. Note that this fictitious chain is formed by only the honest blocks, and may not correspond to the longest chain in the actual system. However, the longest chain in the actual system must grow no slower than this fictitious chain.

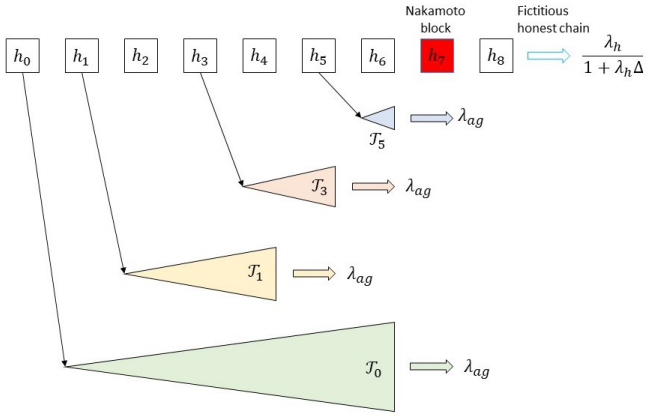

Figure 3: Race between the adversary trees and the fictitious honest chain. While there may be multiple adversary trees simultaneously racing with the honest chain, the growth rate of each tree is bounded by the growth rate of the adversary chain in the private attack. An honest block is a Nakamoto block when all the previous adversary trees never catch up with the honest chain past that block.

rate of each of the adversary trees must be less than that of the fictitious honest chain. What we show, for each of the three classes of protocols, is that under that condition, there must exist honest blocks, which we call *Nakamoto blocks*, each having the property that *none* of the past adversary trees can *ever* catch up after the honest chain reaches the block. These Nakamoto blocks serve to stabilize the blockchain: when each such block enters the blocktree, complex as it may be, we are guaranteed that the entire prefix of the longest chain up to that block remains immutable in the future[1]. When Nakamoto blocks occur and occur frequently, the persistence and liveness of the blockchain is guaranteed.

1.3 Related works

There have been several significant ideas that have emerged from the security analysis of blockchains in the past few years, and below we put our contribution in the perspective of these ideas.

[GKL15] initiated blockchain security analysis through defining key *backbone* properties[2] of chain common prefix, chain quality and chain growth. Applying this framework to analyse the PoW longest chain protocol in the lock-step round-by-round model, it is shown that the common prefix property, the most difficult property to analyze, is satisfied if the number of adversary blocks over a long window is less than the number of uniquely successful honest blocks[3]. A similar block counting analysis is conducted by [PSS17] in the $\Delta-$ synchronous model, with the notion of uniquely successful blocks replaced by the notion of *convergence opportunities*. The resulting bound is tight when $\lambda\Delta$ is small but loose in general. Moreover, the block-counting technique completely breaks down for analyzing PoS longest chain protocols because of the notorious *Nothing-at-Stake* problem: winning one lottery can yield a very large number of blocks for the adversary. To overcome this issue, two new ideas were invented. In the Ouroboros line of work [KRDO17, DGKR18, BGK+18], a new notion of *forkable strings* was invented and a Markov chain analysis was performed to show convergence of the longest chain regardless of adversary action if the adversary stake is below a certain threshold. Sleepy Consensus and SnowWhite [PS17, BPS16] took a different approach and defined a notion of *a pivot*, which is a time instance t such that in all time intervals around t, there are more honest convergence opportunities than the number of adversary slots. They showed that a pivot forces convergence of the longest chain up to that time, and moreover if the adversary stake is less than a certain threshold, then these pivots must occur and they must occur often.

Despite this impressive stream of ideas, the true security threshold was still unknown for both the PoW and PoS longest chain protocols. Moreover, the analysis techniques seem very tied to the specific longest chain protocol under study. The definition of a pivot in [PS17], for example, is tied to the specific longest chain protocol, SnowWhite, they designed. In contrast, the notion of Nakamoto blocks in our approach can be viewed as a more general notion of pivots, but defined for general longest chain protocols and designed to tie the problem back to the private attack. Even though the analysis method in [PS17] has already evolved (or, shall we say, pivoted) from the analysis method in [GKL15], the influence of the block

[1]Thus, Nakamoto blocks have a god-like permanence, they exist, but nobody knows which block is a Nakamoto block.

[2]Properties of the blocktree, independent of the content of the blocks.

[3]A uniquely successful honest block is one that is the only honest block mined in a round.

counting method is still felt in the definition of a pivot. We depart from this method by defining a Nakamoto block directly in terms of structural properties of the evolving blocktree itself. In fact, our approach was motivated from analyzing a protocol like Chia, where the rate of adversary winning slots grows exponentially over time and hence a condition like the one used in [PS17] does not give non-trivial bounds.

The present paper is an extension of an earlier version [BDK+19], where we introduced and applied this approach to analyze a PoS longest chain protocol [FZ18] similar to the Chia protocol. Since we released that early version, we became aware of an independent work [KQR20], which obtains the true security threshold as well as linear consistency for the Ouroboros Praos protocol in the lock-step round-by-round model. They achieved this by tightening the definition of a pivot in [PS17] to count all honest slots, including concurrent ones, not only uniquely successful ones. Like the original definition of pivots, however, this definition is tied to the specific protocol. The approach would not give non-trivial bounds for the Chia protocol, for example. Moreover, their result on the Praos protocol under the Δ-synchronous model is not tight (Figure 1(b)). We believe this is due to their analysis technique of mapping the Δ-synchronous model back to the lock-step round-by-round model. In contrast, our analysis is directly in the Δ-synchronous model and yields tight results in that model.

After the initial submission of this paper, we were made aware of independent work [GKR20], which obtained the same results for the PoW and the Ouroboros PoS protocols, but using totally a different set of techniques based on forkable strings.

1.4 Outline

In Section 2, we introduce a unified model for all three classes of protocols. In Section 3, we introduce the central concepts of this work: blocktree partitioning and Nakamoto blocks. These concepts are applicable to any longest chain protocol. In Section 4, we use these concepts in the security analysis of the three classes of protocol attaining the private attack security threshold of each. In Section 5 we explore the question of whether the private attack is worst case in a stronger sense for longest chain protocols.

2 MODELS

A key goal of this paper is to provide a common framework to analyze the security properties of various longest chain protocols. We focus here primarily on the graph theoretic and the stochastic aspects of the problem: some resource-dependent randomness is utilized by these protocols to select which node is eligible to create a block. The modality in which the randomness is generated leads to different stochastic processes describing the blocktree growth. Understanding these stochastic processes and the ability of the adversary to manipulate these processes to its advantage is the primary focus of the paper.

Different longest chain protocols use different cryptographic means to generate the randomness needed. We specifically exclude here the cryptographic aspects of the protocols, whose analysis is necessary to guarantee the full security of these protocols. In most of the protocols we consider (for example [GKL15, KRDO17]), the cryptographic aspects have already been carefully studied in the original papers and are not the primary bottleneck. In others, further work may be necessary to guarantee the full cryptographic security. In all of these protocols, we assume ideal sources of randomness to create a model that can then be analyzed independently.

We will adopt a continuous-time model, following the tradition set by Nakamoto [Nak08] and also used in several subsequent influential works (eg. [SZ15]) as well as more recent works (eg. [Ren19] and [LG20]). The continuous-time model affords analytical simplicity and allows us to focus on the essence of the problem without being cluttered by too many parameters. Our model corresponds roughly to the Δ−synchronous network model introduced in [PSS17] in the limit of a large number of lottery rounds over the duration of the network delay. This assumption seems quite reasonable. For example, the total hash rate in today's Bitcoin network is about 100 ExaHash/s, i.e. solving 10^{21} puzzles per second. Nevertheless, we believe our results can be extended to the discrete setting.

We first explain the model in the specific context of Nakamoto's Proof-of-Work longest chain protocol, and then generalize it to a unified model for all three classes of protocols we study in this paper.

2.1 Modeling proof-of-work longest chain

The blockchain is run on a network of n honest nodes and a set of malicious nodes. Each honest node mines blocks, adds them to the tip of the longest chain in its current view of the blocktree and broadcasts the blocks to other nodes. Malicious nodes also mine blocks, but they can be mined elsewhere on the blocktree, and they can also be made public at arbitrary times. Due to the memoryless nature of the puzzle solving and the fact that many attempts are tried per second, we model the block mining processes as Poisson with rates proportional to the hashing power of the miner.

Because of network delay, different nodes may have different views of the blockchain. Like the Δ-synchronous model in [PSS17], we assume there is a bounded communication delay Δ seconds between the n honest nodes. We assume malicious nodes have zero communication delay among each other, and they can always act in collusion, which in aggregate is referred as *the adversary*. Also the adversary can delay the delivery of all broadcast blocks by up to Δ time. Hence, the adversary has the ability to have one message delivered to honest nodes at different times, all of which has to be within Δ time of each other.

More formally, the evolution of the blockchain can be modeled as a process $\{(\mathcal{T}(t), C(t), \mathcal{T}^{(p)}(t), C^{(p)}(t)) : t \geq 0, 1 \leq p \leq n\}$, n being the number of honest miners, where:

- $\mathcal{T}(t)$ is a tree, and is interpreted as the *mother tree* consisting of all the blocks that are mined by both the adversary and the honest nodes up until time t, including blocks that are kept in private by the adversary and including blocks that are mined by the honest nodes but not yet heard by other honest nodes in the network.
- $\mathcal{T}^{(p)}(t)$ is an induced (public) sub-tree of the mother tree $\mathcal{T}(t)$ in the view of the p-th honest node at time t. It is the collection of all the blocks that are mined by node p or received from other nodes up to time t.

- $C^{(p)}(t)$ is a longest chain in the tree $\mathcal{T}^{(p)}(t)$, and is interpreted as the longest chain in the local view of the p-th honest node on which it is mining at time t. Let $L^{(p)}(t)$ denote the depth, i.e the number of blocks in $C^{(p)}(t)$ at time t.
- $C(t)$ is the common prefix of all the local chains $C^{(p)}(t)$ for $1 \le p \le n$.

The process evolution is as follows.

- **M0**: $\mathcal{T}(0) = \mathcal{T}^{(p)}(0) = C^{(p)}(0), 1 \le p \le n$ is a single root block, the genesis block.
- **M1**: Adversary blocks are mined following a Poisson process at rate λ_a. When a block is mined by the adversary, the mother tree $\mathcal{T}(t)$ is updated. The adversary can choose which block in $\mathcal{T}(t)$ to be the parent of the adversary block (i.e. the adversary can mine anywhere in the tree $\mathcal{T}(t)$.)
- **M2**: Honest blocks are mined at a total rate of λ_h across all the honest nodes, independent at each honest node and independent of the adversary mining process. When a block is mined by the honest node p, the sub-tree $\mathcal{T}^{(p)}(t)$ and the longest chain $C^{(p)}(t)$ is updated. According to the longest chain rule, this honest block is appended to the tip of $C^{(p)}(t)$. The mother tree $\mathcal{T}(t)$ is updated accordingly.
- **M3**: $\mathcal{T}^{(p)}(t)$ and $C^{(p)}(t)$ can also be updated by the adversary, in two ways: i) a block (whether is honest or adversary) must be added to $\mathcal{T}^{(p)}(t)$ within time Δ once it has appeared in $\mathcal{T}^{(q)}$ for some $q \neq p$, and the longest chain $C^{(p)}(t)$ is extended if the block is at its tip; ii) the adversary can replace $\mathcal{T}^{(p)}(t^-)$ by another sub-tree $\mathcal{T}^{(p)}(t)$ from $\mathcal{T}(t)$ as long as the new sub-tree $\mathcal{T}^{(p)}(t)$ is an induced sub-tree of the new tree $\mathcal{T}^{(p)}(t)$, and can update $C^{(p)}(t^-)$ to a longest chain in $T^{(p)}(t)$. [4]

We highlight the capabilities of the adversary in this model:

- **A1**: Can choose to mine on any one block of the tree $\mathcal{T}(t)$ at any time.
- **A2**: Can delay the communication of blocks between the honest nodes, but no more than Δ time.
- **A3**: Can broadcast privately mined blocks at times of its own choosing: when private blocks are made public at time t to node p, then these nodes are added to $\mathcal{T}^{(p)}(t^-)$ to obtain $\mathcal{T}^{(p)}(t)$. Note that by property M3(i), when private blocks appear in the view of some honest node p, they will also appear in the view of all other honest nodes by time $t + \Delta$.
- **A4**: Can switch the p-th honest node's mining from one longest chain to another of equal length at any time, even when its view of the tree does not change. In this case, $\mathcal{T}^{(p)}(t) = \mathcal{T}^{(p)}(t^-)$ but $C^{(p)}(t) \neq C^{(p)}(t^-)$.

The question is on what information can the adversary base in making these decisions? We will assume a causal adversary which has full knowledge of all past mining times of the honest blocks and the adversary blocks.

Proving the security (persistence and liveness) of the protocol boils down to providing a guarantee that the chain $C(t)$ converges fast as $t \to \infty$ and that honest blocks enter regularly into $C(t)$ regardless of the adversary's strategy.

[4] All jump processes are assumed to be right-continuous with left limits, so that $C(t)$, $\mathcal{T}(t)$ etc. include the new arrival if there is a new arrival at time t.

2.2 From PoW to a unified model

The model introduced in the last section can serve as a unified model for all three classes of protocols we study in this paper. The key difference between these classes of protocols is how the lottery in winning block proposal slots is conducted. This difference can be encapsulated by changing only one modeling assumption: **M1**, the assumption on the adversary mining process (Figure 4). In particular, the assumption on the honest behavior (**M2**) remains the same,

- **M1-PoW** (Proof-of-Work): The original assumption we already had: Adversary blocks are mined according to a Poisson process at rate λ_a, and the mined block can be appended to any parent block but only one, of the adversary's choosing, in the current mother tree $\mathcal{T}(t)$. This models the random attempts at solving the hash puzzle on one of the existing blocks.
- **M1-PS** (Praos/SnowWhite Proof-of-Stake model): The adversary blocks are mined[5] according to a Poisson process at rate λ_a (similar to PoW), but the adversary is allowed to append a version of each mined block simultaneously at *all* the blocks in the current tree $\mathcal{T}(t)$.
- **M1-Chia** (Chia Proof-of-Space model): The adversary blocks are mined according to multiple independent Poisson processes of rate λ_a, one at each block of the current tree $\mathcal{T}(t)$. A new block is appended to the tree at a certain block when a mining event happens.

Under **M1-PoW**, miners can only mine on one parent block at a time, a consequence of conservation of work. Hence, the mined block can only be appended to one of the parent blocks. In **M1-PS** and **M1-Chia**, the adversary is able to mine new blocks on *all* of the existing blocks of the blocktree. This is a consequence of the phenomenon of *Nothing-at-stake*: the same resource (stake in PoS, disk space in PoSpace) can be used by the nodes to participate in random lotteries at all parent blocks to propose new blocks. Hence, unlike under assumption **M1-PoW**, the overall mining rate of adversary blocks increases as the tree $\mathcal{T}(t)$ grows over time under both **M1-PS** and **M1-Chia**. However, the mining events across different blocks are fully *dependent* in **M1-PS** and completely *independent* in **M1-Chia**. This is a consequence of the difference of how randomness is used in running the lotteries at different blocks. In the case of Praos/SnowWhite, the same randomness is used. In the case of Chia, independent randomness is used.

We note that it may appear that the capability **A1** of the adversary (choosing where to mine), which is present in **M1-PoW**, is gone under **M1-PS** and **M1-Chia**. However, the reason is that the adversary does not have to choose because it can mine everywhere simultaneously. Thus the adversary is actually more powerful under the **M1-PS** and **M1-Chia** conditions because the adversary has at its disposal much larger number of adversary blocks to attack

[5] In these Proof-of-Stake protocols, block proposal slots are won by conducting lotteries using the keys of the stake holders rather than by solving difficult computational puzzles as in Proof-of-Work protocols. However,for convenience, we use the term "mining" to denote the winning of any type of lotteries.

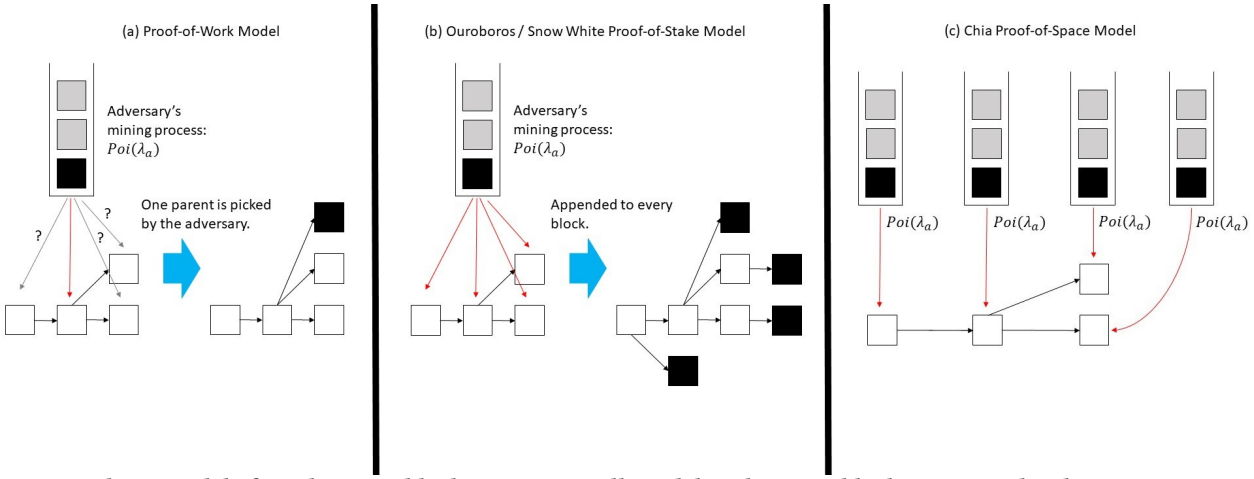

Figure 4: Three models for adversary block mining. In all models, adversary blocks are visualized as arriving via Poisson queues, and the focus is on how the block at the head of each queue is appended to the blocktree. In the PoW model, each adversary block can be appended to exactly one of the parent blocks of the existing blocktree. In the Paos/SnowWhte model, each adversary block can be appended to all possible parents blocks. In the Chia PoSpace model, the adversary blocks are mined independently on the parent blocks of the existing tree.

the protocol. Somewhat surprisingly, our security threshold results show that this extra power is not useful in Praos/SnowWhite but useful in Chia.

The modeling assumptions for these protocols will be justified in more details in the following two subsections. The reader who is comfortable with these assumptions can go directly to Section 3.

2.3 Ouroboros Praos and Snow White Proof-of-Stake model

This section shows how Ouroboros Praos [DGKR18] and Snow White [BPS16] Proof-of-Stake protocols can be modeled using assumption **M1-PS** as mentioned earlier. Both of these are Proof-of-Stake protocols, which means nodes get selected to create blocks in proportion to the number of coins (=stake) that they hold rather than the computation power held by the nodes. While the two protocols are similar at the level required for the analysis here, for concreteness, we will describe here the relation with Ouroboros Praos, which can handle adaptive corruption of nodes.

We consider here only the static stake scenario - the stake of various nodes is fixed during the genesis block and assume that there is a single epoch (the composition of epochs into a dynamic stake protocol can be done using the original approach in [DGKR18]). The common randomness as well as the stake of various users is fixed at genesis (more generally, these are fixed at the beginning of each epoch for the entire duration of the epoch). For this protocol, we will assume that all nodes have a common clock (synchronous execution). At each time t, every node computes a verifiable random function (VRF) of the current time, the common randomness and the secret key. If the output value of the VRF is less than a certain threshold, then that node is allowed to propose a block at that time, to which it appends its signature. The key property of the VRF is that any node with knowledge only of the public key can validate that it was obtained with a node possessing the corresponding secret key. An honest node will follow the prescribed protocol and thus only create one block which it will append to the longest chain in its view. However, a winning dishonest node can create many different blocks mining on top of distinct chains. Blocks which are well-embedded into the longest-chain are considered confirmed.

Now, we explain the connection of the protocol to our modeling in the earlier section. The first assumption is that time is quantized so finely that the continuous time modeling makes sense - this assumes that there is no simultaneous mining at any time point. However, if nodes mine blocks close to each other in time, they can be forked due to the delay Δ in the propagation time (thus we model concurrent mining through the effect of the propagation delay rather than through discrete time). Second, the honest action is to grow the longest chain through mining a new block at the tip - this justifies M2 (here λ_h is proportional to the total honest stake). The adversaries can mine blocks which can be appended to many different positions in the blockchain. We assume that in the worst case, every adversary arrival contributes to a block extending every single block in the tree. We note that furthermore, there is another action, which is that the adversary can create many different blocks at any given position of the blockchain. Since this action does not increase the length of any chain or increase future mining opportunities, we do not need to model this explicitly. However, we point out that, since we show that a certain prefix of the blockchain ending at a honest block remains fixed for all future, that statement continues to hold even under this expanded adversary action space.

2.4 Chia Proof-of-Space model

Chia consensus [CP19] incorporates a combination of Proof of Space (PoSpace) and Proof of time, and is another energy efficient alternative to Bitcoin. PoSpace [AAC$^+$17, DFKP15] is a cryptographic technique where provers can efficiently generate proofs to show that they allocate unused hard drive space for storage space. Proof of time is implemented by a Verifiable Delay Function (VDF) [BBBF18, Pie18] that requires a certain amount of sequential computations to execute, but can be verified far quicker: a VDF takes a challenge $c \in \{0,1\}^w$ and and a time parameter $t \in \mathbb{Z}^+$ as

input, and produces a output τ and a proof π in (not much more than) t sequential steps; the correctness of output τ can be verified with the proof π in much less than t steps. PoSpace enables Sybil resistance by restricting participation to nodes that have reserved enough hard disk space and VDF enables coordination without having synchronized clocks as well as preventing long-range attacks [PKF+18].

In Chia, each valid block B contains a PoSpace σ and a VDF output τ. A Chia full node mines a new block (B_i, with i denoting the depth of the block from Genesis) as follows:

(1) It first picks the block B_{i-1}, at the tip of the longest chain in its local view of the blocktree, as the parent block that the newly generated block B_i will be appended to.
(2) It draws a challenge c_1 deterministically from B_{i-1} and generates a valid PoSpace σ_i based on c_1 and a large file of size at least M bits it stores.
(3) It computes a valid VDF output τ_i based on a challenge c_2 and a time parameter t, where c_2 is also drawn deterministically from B_{i-1} and t is the hash of σ_i multiplied by a difficulty parameter T (i.e. $t = 0.\mathsf{H}(\sigma_i) \times T$ where H is a cryptographic hash function).
(4) A new block B_i comprised of σ_i, τ_i and some payload (example: transactions) is appended to B_{i-1} in the blocktree.

For each node, the "mining" time of a new block follows a uniform distribution in $(0, T)$: this is because the hash function H outputs a value that is uniformly distributed over its range. Suppose there are N full nodes in the Chia network, then the inter-arrival block time in Chia consensus would be $\min(U_1, U_2, \cdots, U_N)$, where $U_i \sim \mathsf{Unif}(0, T)$ for $1 \le i \le N$. Then the expected inter-arrival block time is

$$\mathbb{E}[\min(U_1, U_2, \cdots, U_N)] = \int_0^T (1 - t/T)^N dt = \frac{T}{N+1}.$$

So to maintain a fixed inter-arrival block time (example: 10 minutes in Bitcoin), the difficulty parameter T needs to be adjusted linearly as number of full nodes N grows. We also observe that the chance for a node storing two large files each of size at least M bits to find the first block is exactly doubled compared with a node storing one file, which provides Sybil resistance to Chia. Further we can model the mining process in Chia as a Poisson point process for large N. Fixing a parent block in the block tree, the number of new blocks mined in time t follows a binomial distribution $\mathsf{bin}(N, t/T)$, which approaches a Poisson distribution $\mathsf{Poi}(Nt/T)$ when $N \to \infty$ and $N/T \to C$ for some constant C.

Assume there are n honest nodes each controlling M bits of space, and the adversary has $a \cdot M$ bits of space, then the mining processes of honest blocks and adversary blocks are Poisson point processes with rate λ_h and λ_a respectively, where λ_h and λ_a are proportional to total size of disk space controlled by honest nodes ($n \cdot M$) and the adversary ($a \cdot M$) respectively. Also while the honest nodes are following the longest chain rule, the adversary can work on multiple blocks or even all blocks in the block tree as a valid PoSpace is easy to generate and the adversary can compute an unlimited amount of VDF outputs in parallel; a similar phenomenon occurs in Proof-of-Stake blockchains where it is termed as the Nothing-at-Stake (NaS) attack [BDK+19]. Hence, we can model the adversary blocks as generated according to multiple independent Poisson processes of rate λ_a, one at each block of the current tree $\mathcal{T}(t)$. A new block is appended to the tree at a certain block when a generation event happens. Like in the model for Ouroboros Praos and Snow White, the total rate of adversary block generation increases as the tree grows; however the generation events across different blocks are independent rather than fully dependent.

3 BLOCKTREE PARTITIONING AND NAKAMOTO BLOCKS

In this section, we will introduce the concept of *blocktree partitioning* to represent a general adversary attack as a collection of adversary trees racing against a fictitious honest chain. Using this representation, we define the key notion of *Nakamoto blocks* as honest blocks that are the winners of the race against all the past trees, and show that if a block is a Nakamoto block, then the block will forever remain in the longest chain. The results in this section apply to all three models. In fact, they are valid for any assumption on the adversary mining process in **M1** in the model in Section 2.1, because no statistical assumptions are made. In Section 4, we will perform security analysis in all three backbone models using the tool of Nakamoto blocks, by showing that they occur frequently with high probability whenever the adversary power is not sufficient to mount a successful private attack. This proves the liveness and persistency of the protocols.

First, we introduce the concept of blocktree partitioning and define Nakamoto blocks in the simpler case when $\Delta = 0$, and then we extend to general Δ. The unrealistic but pedagogically useful zero-delay case allows us to focus on the capability of the adversary to mine and publish blocks, while the general case brings in its capability to delay the delivery of blocks by the honest nodes as well.

3.1 Network delay $\Delta = 0$

3.1.1 Blocktree partitioning

Let τ_i^h and τ_i^a be the mining time of the i-th honest and adversary blocks respectively; $\tau_0^h = 0$ is the mining time of the genesis block, which we consider as the 0-th honest block.

Definition 3.1. **Blocktree partitioning** *Given the mother tree $\mathcal{T}(t)$, define for the i-th honest block b_i, the* adversary tree $\mathcal{T}_i(t)$ *to be the sub-tree of the mother tree $\mathcal{T}(t)$ rooted at b_i and consists of all the adversary blocks that can be reached from b_i without going through another honest block. The mother tree $\mathcal{T}(t)$ is partitioned into sub-trees $\mathcal{T}_0(t), \mathcal{T}_1(t), \ldots T_j(t)$, where the j-th honest block is the last honest block that was mined before time t.*

See Figure 2(b) for an example.

The sub-tree $\mathcal{T}_i(t)$ is born at time τ_i^h as a single block b_i and then grows each time an adversary block is appended to a chain of adversary blocks from b_i. Let $D_i(t)$ denote the depth of $\mathcal{T}_i(t)$; $D_i(\tau_i^h) = 0$.

3.1.2 Nakamoto blocks

Let $A_h(t)$ be the number of honest blocks mined from time 0 to t. $A_h(t)$ increases by 1 at each time τ_i^h. We make the following important definition.

Definition 3.2. *(**Nakamoto block for** $\Delta = 0$**)** Define*

$$E^0_{ij} = \textit{event that } D_i(t) < A_h(t) - A_h(\tau^h_i) \textit{ for all } t > \tau^h_j \quad (4)$$

for some $i < j$. The j-th honest block is called a Nakamoto block *if*

$$F^0_j = \bigcap_{i=0}^{j-1} E^0_{ij} \quad (5)$$

occurs.

We can interpret the definition of a Nakamoto block in terms of a *fictitious system*, having the same block mining times as the actual system, where there is a growing chain consisting of only honest blocks and the adversary trees are racing against this honest chain. (Figure 3). The event E^0_{ij} is the event that the adversary tree rooted at the i-th honest block does not catch with the fictitious honest chain *any* time after the mining of the j-th honest block. When the fictitious honest chain reaches a Nakamoto block, it has won the race against *all* adversary trees rooted at the past honest blocks.

Even though the events are about a fictitious system with a purely honest chain and the longest chain in the actual system may consist of a mixture of adversary and honest blocks, the actual chain can only grow faster than the fictitious honest chain, and so we have the following key lemma showing that a Nakamoto block will stabilize and remain in the actual chain forever.

Lemma 3.1. *(**Nakamoto blocks stabilize,** $\Delta = 0$**.)** If the j-th honest block is a Nakamoto block, then it will be in the longest chain $C(t)$ for all $t > \tau^h_j$. Equivalently, $C(\tau^h_j)$ will be a prefix of $C(t)$ for all $t > \tau^h_j$.*

Proof. Note that although honest nodes may have different views of the longest chain because of the adversary capability **A4**, $\mathcal{T}^{(p)}(t) = \mathcal{T}^{(q)}(t)$ and hence $L^{(p)}(t) = L^{(q)}(t)$ always hold for any $q \neq p$ at any time t when $\Delta = 0$. Let $L(t)$ be the length of the longest chain in the view of honest nodes. $L(0) = 0$. Note that since the length of the chain $C^{(p)}(t)$ increments by 1 immediately at every honest block mining event (this is a consequence of $\Delta = 0$), it follows that for all i and for all $t > \tau^h_i$,

$$L(t) - L(\tau^h_i) \geq A_h(t) - A_h(\tau^h_i). \quad (6)$$

We now proceed to the proof of the lemma.

We will argue by contradiction. Suppose F^0_j occurs and let $t^* > \tau^h_j$ be the smallest t such that $C(\tau^h_j)$ is not a prefix of $C^{(p)}(t)$ for some $1 \leq p \leq n$. Let b_i be the last honest block on $C^{(p)}(t^*)$ (which must exist, because the genesis block is by definition honest.) If b_i is generated at some time $t_1 > \tau^h_j$, then $C^{(p)}(t_1^-)$ is the prefix of $C^{(p)}(t^*)$ before block b_i, and does not contain $C(\tau^h_j)$ as a prefix, contradicting the minimality of t^*. So b_i must be generated before τ^h_j, and hence b_i is the i-th honest block for some $i < j$. The part of $C^{(p)}(t^*)$ after block b_i must lie entirely in the adversary tree $T_i(t^*)$ rooted at b_i. Hence,

$$L(t^*) \leq L(\tau^h_i) + D_i(t^*).$$

However we know that

$$D_i(t^*) < A_h(t^*) - A_h(\tau^h_i) \leq L(t^*) - L(\tau^h_i), \quad (7)$$

where the first inequality follows from the fact that F_j holds, and the second inequality follows from the longest chain policy (eqn. (6)). From this we obtain that

$$L(\tau^h_i) + D_i(t^*) < L(t^*), \quad (8)$$

which is a contradiction since $L(t^*) \leq L(\tau^h_i) + D_i(t^*)$. □

Lemma 3.1 justifies the name *Nakamoto block*: just like its namesake, a Nakamoto block has a godlike permanency. Also like its namesake, no one knows for sure whether a given block is a Nakamoto block: it is defined in terms of what happens in the indefinite future. However, the concept is useful because as long as a Nakamoto block appears in the last k blocks of the current longest chain, then the prefix before these k blocks will stabilize. Hence, the problem is reduced to showing under what conditions Nakamoto blocks exist and they enter the blockchain frequently.

Since Nakamoto blocks are defined in terms of a race between adversary trees and the honest chain, and the growth rate of each adversary tree is bounded by the growth rate of the private attack adversary chain no matter what the attack is, one can intuitively expect that if the private attack is not successful, i.e. the growth rate of the private adversary chain is less than that of the honest chain, then once in a while Nakamoto blocks will occur because the adversary trees cannot win all the time. This intuition is made precise in Section 4 for the three models of interest. The current task at hand is to extend the notion of Nakamoto blocks to the $\Delta > 0$ case.

3.2 General network delay Δ

Definition 3.2 of a Nakamoto block is tailored for the zero network delay case. When the network delay $\Delta > 0$, there is forking in the blockchain even without adversary blocks, and two complexities arise:

(1) Even when a honest block b has won the race against all the previous adversary trees, there can still be multiple honest blocks on the same level as b in the mother tree $\mathcal{T}(t)$ due to forking. Hence there is no guarantee that b will remain in the longest chain.
(2) Even when the honest block b is the only block in its level, the condition in Equation (4) is not sufficient to guarantee the stabilization of b: the number of honest blocks mined is an over-estimation of the amount of growth in the honest chain due to forking.

The first complexity is a consequence of the fact that when the network delay is non-zero, the adversary has the additional power to delay delivery of honest blocks to create split view among the honest nodes. In the context of the formal security analysis of Nakamoto's PoW protocol, the limit of this power is quantified by the notion of *uniquely successful* rounds in [GKL15] in the lock-step synchronous round-by-round model, and extended to the notion of *convergence opportunities* in [PSS17] in the Δ-synchronous model. The honest blocks encountering the convergence opportunities are called *loners* in [Ren19].

Definition 3.3. *The j-th honest block mined at time τ^h_j is called a* loner *if there are no other honest blocks mined in the time interval $[\tau^h_j - \Delta, \tau^h_j + \Delta]$.*

It is shown in [PSS17, Ren19] that a loner must be the only honest block in its depth in $\mathcal{T}(t)$ at any time t after the block is mined. Thus, to deal with the first complexity, we simply strengthen the definition of a Nakamoto block to restrict it to also be a loner block. Since loner blocks occur frequently, this is not an onerous restriction.

To deal with the second complexity, we define the race of the adversary trees not against a fictitious honest chain without forking as in definition 3.2, but against a fictitious honest tree with worst-case forking. This tree is defined as follows.

Definition 3.4. *Given honest block mining times τ_i^h's, define a honest fictitious tree $\mathcal{T}_h(t)$ as a tree which evolves as follows:*

(1) $\mathcal{T}_h(0)$ is the genesis block.
(2) The first mined honest block and all honest blocks within Δ are all appended to the genesis block at their respective mining times to form the first level.
(3) The next honest block mined and all honest blocks mined within time Δ of that are added to form the second level (which first level blocks are parents to which new blocks is immaterial) .
(4) The process repeats.

Let $D_h(t)$ be the depth of $\mathcal{T}_h(t)$.

We are now ready to put everything together to define Nakamoto blocks in general.

Definition 3.5. *(**Nakamoto block for general** Δ) Let us define:*

$$E_{ij} = \text{event that } D_i(t) < D_h(t-\Delta) - D_h(\tau_i^h + \Delta) \text{ for all } t > \tau_j^h + \Delta. \tag{9}$$

The j-th honest block is called a Nakamoto block *if it is a loner and*

$$F_j = \bigcap_{i=0}^{j-1} E_{ij} \tag{10}$$

occurs.

Note that when $\Delta = 0$, $D_h(t) = A_h(t)$, the number of honest blocks mined in $[0, t]$. Hence $E_{ij} = E_{ij}^0$. Also, every block is a loner. Here Definition 3.5 degenerates to Definition 3.2. Moreover, it is not difficult to see that

$$D_h(t-\Delta) - D_h(\tau_i^h + \Delta) \le A_h(t) - A_h(\tau_i^h)$$

so Definition 3.5 is indeed a strengthening of Definition 3.2. This strengthening allows us to show that Nakamoto blocks stabilize for all $\Delta > 0$.

Theorem 3.2. *(**Nakamoto blocks stabilize, general** Δ) If the j-th honest block is a Nakamoto block, then it will be in the chain $C(t)$ for all $t > \tau_j^h + \Delta$. This implies that the longest chain until the j-th honest block has stabilized.*

The proof of Theorem 3.2 is given in §B.

Nakamoto blocks are defined for general longest chain protocols. When applied to the Praos/SnowWhite protocols, the definition of Nakamoto blocks is a weakening of the definition of pivots in [PS17]. Although [PS17] did not define pivots explicitly in terms of races, one can re-interpret the definition as a race between the adversary and a fictitious honest chain consisting of *only* loner honest blocks. This fictitious chain can never occur in the actual system even when no adversary blocks are made public, because there are other honest blocks which are not loners but can make it into the main chain. On the other hand, Nakamoto blocks are defined directly as a race between the adversary and the fictitious honest chain which would arise if there were no public adversary blocks. This is why the definition of Nakamoto blocks leads to a tight characterization of the security threshold in the Praos/SnowWhite protocols, matching the private attack threshold, while the definition of pivots in [PS17] cannot. (Theorem 4.2). This tightening is similar to the tightening done in the recent work [KQR20] for the lock-step round-by-round model.

4 SECURITY ANALYSIS

The goal of this section is to show that the private attack is the worst attack for the three models defined in Section 2. More precisely, we want to show that *security threshold*, i.e. the maximum adversary power tolerable for any adversary strategy, is the same as that of Nakamoto's private attack. This is true for any total mining rate λ and for any Δ. (In fact, the threshold depends only on the product $\lambda\Delta$.) We will use the notion of Nakamoto blocks to establish these results.

4.1 Statement of results

Our goal is to generate a transaction ledger that satisfies *persistence* and *liveness* as defined in [GKL15]. Together, persistence and liveness guarantee robust transaction ledger; honest transactions will be adopted to the ledger and be immutable.

Definition 4.1 (from [GKL15]). A protocol Π maintains a robust public transaction ledger if it organizes the ledger as a blockchain of transactions and it satisfies the following two properties:

- (Persistence) Parameterized by $\tau \in \mathbb{R}$, if at a certain time a transaction tx appears in a block which is mined more than τ time away from the mining time of the tip of the main chain of an honest node (such transaction will be called confirmed), then tx will be confirmed by all honest nodes in the same position in the ledger.
- (Liveness) Parameterized by $u \in \mathbb{R}$, if a transaction tx is received by all honest nodes for more than time u, then all honest nodes will contain tx in the same place in the ledger forever.

As discussed in the introduction, the condition for the private attack on Nakamoto's Proof-of-Work protocol to be successful is

$$\lambda_a > \lambda_{\text{growth}}(\lambda_h, \Delta) = \frac{\lambda_h}{1 + \lambda_h \Delta} \tag{11}$$

in the fully decentralized regime. In terms of β, the fraction of adversary power, and λ, the total block mining rate:

$$\beta > \frac{1-\beta}{1 + (1-\beta)\lambda\Delta}. \tag{12}$$

The parameter $\lambda\Delta$ is the number of blocks generated per network delay, and determines the latency and throughput of the blockchain. If this condition is satisfied, then clearly the ledger does not have persistency or liveness. Hence, the above condition can be interpreted as a tradeoff between latency/throughput and the security (under private attack).

In the Praos/SnowWhite protocol, the honest growth rate is the same as in the PoW system. Consider now the adversary blocks. They are mined according to a Poisson process at rate λ_a. When a block is mined, the adversary gets to append that block to all the blocks in the current adversary chain (cf. Figure 4(b)). This leads to an exponential increase in the number of adversary blocks. However, the *depth* of that chain increases by exactly 1. Hence the growth of the adversary chain is exactly the same as the advversary chain under PoW. Hence, we get exactly the same private attack threshold (12) in both the PoW and the Praos/SnowWhite PoS protocols.

The theorem below shows that the the private attack threshold yields the true security threshold for both classes of protocols.

Theorem 4.2. *If*

$$\beta < \frac{1-\beta}{1+(1-\beta)\lambda\Delta}, \tag{13}$$

then the Nakamoto PoW and the Ouroboros/SnowWhite PoS protocols generate transaction ledgers such that each transaction tx[6] *satisfies* persistence *(parameterized by $\tau = \sigma$) and* liveness *(parameterized by $u = \sigma$) in Definition 4.1 with probability at least $1 - e^{-\Omega(\sigma^{1-\varepsilon})}$, for any $\varepsilon > 0$.*

For the Chia Proof-of-Space model, the private attack is analyzed in [CP19, FZ18]. The growth rate of the private adversary chain is $e\lambda_a$. (The magnification by a factor of e is due to the Nothing-at-Stake nature of the protocol; more on that in Section 4.4.). Hence the condition for success for the private attack is:

$$e\lambda_a > \frac{\lambda_h}{1+\lambda_h\Delta}, \tag{14}$$

in the fully decentralized setting. This implies the following condition on β, the adversary fraction of space resources:

$$e\beta > \frac{1-\beta}{1+(1-\beta)\lambda\Delta}. \tag{15}$$

For the Chia model, this threshold yields the true threshold as well.

Theorem 4.3. *If*

$$e\beta < \frac{1-\beta}{1+(1-\beta)\lambda\Delta}, \tag{16}$$

then the Chia Proof-of-Space protocol generate transaction ledgers satisfying persistence *(parameterized by $\tau = \sigma$) and* liveness *(parameterized by $u = \sigma$) in Definition 4.1 with probability at least $1 - e^{-\Omega(\sigma^{1-\varepsilon})}$, for any $\varepsilon > 0$.*

The security thresholds for the different models are plotted in Figure 1, comparing to existing lower bounds in the literature.

[6]In contrast to the theorems in [GKL15, PSS17], this theorem guarantees high probability persistence and liveness for *each* transaction rather than for the entire ledger. This is because our model has an infinite time-horizon while their model has a finite horizon, and guarantees for an infinite ledger is impossible. However, one can easily translate our results to high probability results for an entire finite ledger over a time horizon of duration polynomial in the security parameter σ using the union bound.

4.2 Approach

To prove Theorems 4.2 and 4.3, we use the technique of Nakamoto blocks developed in Section 3. Theorem 3.2 states that Nakamoto blocks remain in the longest chain forever. The question is whether they exist and appear frequently regardless of the adversary strategy. If they do, then the protocol has liveness and persistency: honest transactions can enter the ledger frequently through the Nakamoto blocks, and once they enter, they remain at a fixed location in the ledger. More formally, we have the following result.

Lemma 4.4. *Define $B_{s,s+t}$ as the event that there is no Nakamoto blocks in the time interval $[s, s+t]$. If*

$$P(B_{s,s+t}) < q_t < 1 \tag{17}$$

for some q_t independent of s and the adversary strategy, then the protocol generates transaction ledgers satisfying persistence *(parameterized by $\tau = \sigma$) and* liveness *(parameterized by $u = \sigma$) in Definition 4.1 with probability at least $1 - q_\sigma$.*

The proof of Lemma 4.4 can be found in §E. This reduces the problem to that of bounding the probability that there are no Nakamoto blocks in a long duration. Here we follow a similar style of reasoning as in the analysis of occurrence of pivots in the Sleepy Consensus protocol [PS17]:

(1) Show that the probability that the j-th honest block is a Nakamoto block is lower bounded by some $p > 0$ for all j and for all adversary strategy, in the parameter regime when the private attack growth rate is less than the honest chain growth rate.
(2) Bootstrap from (1) to bound the probability of the event $B_{s,t}$, an event of no occurrence of Nakamoto blocks for a long time.

Intuitively, if (1) holds, then one would expect that the chance that Nakamoto blocks do not occur over a long time is low, provided that a block being Nakaomoto is close to independent of another block being Nakamoto if the mining times of the two blocks are far apart. We perform the bootstrapping by exploiting this fact for the various models under consideration.

In [PS17], the bootstrapping yields a bound $\exp(-\Omega(\sqrt{t}))$ on $P(B_{s,s+t})$. By recursively applying the bootstrapping procedure, we are able to get a bound $\exp(-\Omega(t^{1-\varepsilon}))$ on $P(B_{s,s+t})$, for any $\varepsilon > 0$. We apply this general analysis strategy to the three models in the next two subsections.

4.3 Nakamoto PoW and Praos/SnowWhite PoS Models

This subsection is dedicated to proving Theorem 4.2. We will show that if

$$\lambda_a < \frac{\lambda_h}{1+\lambda_h\Delta}, \tag{18}$$

then Nakamoto blocks occur frequently and regularly under both the PoW and the Praos/SnowWhite PoS models. Since the adversary in the Praos/SnowWhite PoS model is stronger, it suffices for us to prove the statement in that model.

As outlined in the section above, to prove Theorem 4.2, we need to show that there exists constants $A_\varepsilon, a_\varepsilon > 0$ such that

$$P(B_{s,s+t}) < A_\varepsilon \exp(-a_\varepsilon t^{1-\varepsilon})$$

for all $s, t > 0$. In this context, we first establish that the probability of occurrence of a Nakamoto block is bounded away from 0.

Lemma 4.5. *If*

$$\lambda_a < \frac{\lambda_h}{1 + \Delta\lambda_h},$$

there exists a constant $p > 0$ such that the probability that the j-th honest block is a Nakamoto block is at least p for all j.

The proof of Lemma 4.5 is given in §C.1 of the Appendix. It is based on connecting the event of being a Nakamoto block to the event of a random walk never returning to the starting point. An alternative proof is presented in §C.2 of [DKT+20].

We next obtain a bound on $P(B_{s,s+t})$.

Lemma 4.6. *If*

$$\lambda_a < \frac{\lambda_h}{1 + \Delta\lambda_h},$$

then for any $\varepsilon > 0$ there exist constants $a_\varepsilon, A_\varepsilon$ so that for all $s, t \geq 0$,

$$P(B_{s,s+t}) < A_\varepsilon \exp(-a_\varepsilon t^{1-\varepsilon}).$$

Proof of Lemma 4.6 is given in §C.2 of the Appendix. Then combining Lemma 4.6 with Lemma 4.4 implies Theorem 4.2.

4.4 Chia Proof-of-Space Model

This subsection is dedicated to proving Theorem 4.3. We will show that if

$$e\lambda_a < \frac{\lambda_h}{1 + \lambda_h\Delta}, \tag{19}$$

then Nakamoto blocks occur frequently and regularly under the Chia Proof-of-Space model.

Since the occurrence of a Nakamoto block depends on whether the adversary trees from the previous honest blocks can catch up with the (fictitious) honest tree, we next turn to an analysis of the growth rate of an adversary tree. Note that under assumption **M1 − Chia**, adversary blocks are mined at rate λ_a independently at each block of the mother tree $\mathcal{T}(t)$. Hence, each adversary tree $\mathcal{T}_i(t)$ grows statistically in the same way (and independent of each other). Without loss of generality, let us focus on the adversary tree $\mathcal{T}_0(t)$, rooted at genesis, of the tree $\mathcal{T}(t)$. The depth of the tree $\mathcal{T}_0(t)$ is $D_0(t)$ and defined as the maximum depth of its blocks. The genesis block is always at depth 0 and hence $\mathcal{T}_0(0)$ has depth zero.

With the machinery of branching random walks, we can show that the growth rate of depth of $\mathcal{T}_0(t)$ is $e\lambda_a$ while the total number of adversary blocks in $\mathcal{T}_0(t)$ grows exponentially with time t. Hence, compared to the Praos/SnowWhite model we just analyzed, the growth rate of each adversary tree is magnified by a factor of e. Thus, the Nothing-at-Stake phenomenon is more significant in the Chia model compared to the Praos/SnowWhite model, due to the independence of mining opportunities at different blocks.

We will also need a tail bound on $D_0(t)$. While such estimates can be read from [Shi15], we bring instead a quantitative statement suited for our needs.

Lemma 4.7. *For $m \geq 1$,*

$$P(D_0(t) \geq m) \leq \left(\frac{e\lambda_a t}{m}\right)^m. \tag{20}$$

Details on the analysis of $\mathcal{T}_0(t)$ and the proof of Lemma 4.7 are in §D.1 in the Appendix.

With Lemma 4.7, we show below that in the regime $e\lambda_a < \frac{\lambda_h}{1+\lambda_h\Delta}$, Nakamoto blocks has a non-zero probability of occurrence.

Lemma 4.8. *If*

$$e\lambda_a < \frac{\lambda_h}{1 + \lambda_h\Delta},$$

then there is a $p > 0$ such that that probability the j-th honest block is a Nakamoto block is greater than p for all j.

The proof of this result can be found in §D.2 of the Appendix.

Having established the fact that Nakamoto blocks occurs with non-zero frequency, we can bootstrap on Lemma 4.8 to get a bound on the probability that in a time interval $[s, s + t]$, there are no Nakamoto blocks, i.e. a bound on $P(B_{s,s+t})$.

Lemma 4.9. *If*

$$e\lambda_a < \frac{\lambda_h}{1 + \lambda_h\Delta},$$

then for any $\varepsilon > 0$ there exist constants $\bar{a}_\varepsilon, \bar{A}_\varepsilon$ so that for all $s, t \geq 0$,

$$P(B_{s,s+t}) \leq \bar{A}_\varepsilon \exp(-\bar{a}_\varepsilon t^{1-\varepsilon}). \tag{21}$$

The proof of this result is almost verbatim identical with Lemma 4.6, and will not be repeated here. The complete proof can be found in §D.3 of [DKT+20]. Then combining Lemma 4.9 with Lemma 4.4 implies Theorem 4.3.

5 DOES NAKAMOTO REALLY ALWAYS WIN?

We have shown that the threshold for the adversary power beyond which the private attack succeeds is in fact the tight threshold for the security of the three models **M1-PoW**, **M1-PS** and **M1-Chia**. However, security threshold is a statistical concept. Can we say that the private attack is the worst attack in a stronger, deterministic, sense?

Indeed, it turns out that one can, with a slight strengthening of the private attack, in a special case: the PoW model with network delay $\Delta = 0$. In this setting, we can indeed make a stronger statement.

In the PoW model, any attack strategy π consists of two components: where to place each new adversary arrival and when to release the adversary blocks. Consider a specific attack π_{SZ}: the Sompolinsky and Zohar's strategy of private attack with pre-mining [SZ16]. This attack focuses on a block b: it builds up a private chain with the maximum lead over the public chain when block b is mined, and then starts a private attack from that lead. We have the following result.

Theorem 5.1. *Let $\tau_1^h, \tau_2^h, \ldots$ and $\tau_1^a, \tau_2^a, \ldots$ be a given sequence of mining times of the honest and adversary blocks. Let b be a specific block. (i) Suppose π violates the persistence of b with parameter k, i.e. b leaves the longest chain after becoming k-deep. Then the π_{SZ} attack on b also forces b to leave the longest chain after becoming k-deep, under the* same *mining times. (ii) Suppose b is an honest block and π violates liveness for the k consecutive honest blocks starting with b, i.e. none of the k consecutive honest blocks starting with b stay in the longest chain indefinitely. Then the π_{SZ} attack on b also forces these k consecutive honest blocks to leave the longest chain indefinitely under the* same *mining times.*

The full proof of this theorem, together with a counter-example in the case of $\Delta > 0$, can be found in §F of [DKT+20]. To demonstrate the main ideas used in the full proof, we focus here on a special case of where the adversary attacks the first honest block, b_1, mined after the genesis block. Note that in this special case, the Sompolinsky and Zohar's attack strategyπ_{SZ} against b_1 is simply Nakamoto's private attack starting at the genesis block. In this context, we prove that if the persistence of b_1 with parameter k is violated by an adversary following some arbitrary attack strategy π, then, it is also violated by an adversary following the private attack under the same sequence of mining times for the honest and adversary blocks. The proof will be built on the observation that at any depth, there can be at most one honest block when $\Delta = 0$. This observation is a direct result of the Chain Growth Lemma in [GKL15], and is a consequence of the fact that there is no forking among the honest blocks when delay $\Delta = 0$.

Proof. Let $L(.)$ and $L^*(.)$ denote the lengths of the public longest chains, denoted by C and C^* under π and the private attack respectively. Let τ_1 be the mining time of block b_1, and, define $t > \tau_1$ as the first time block b disappears from C after it becomes k deep within C, under π. Let H and A denote the number of honest and adversary blocks mined by time t under the given sequence of mining times.

We first focus on π. Since π removes b_1 from C at time t, there is another chain building on the genesis block that is parallel to C and at least as long as C at time t. (See top of Figure 5.) Since there can be at most one honest block at every depth and there cannot be any honest block deeper than $L(t)$ (by virtue of the fact that $L(t)$ is the length of the public longest chain), $A \geq L(t) \geq H$. Also, since b_1 is at least k deep at time t, $L(t) \geq k$. Hence, $A \geq \max\{H, k\}$.

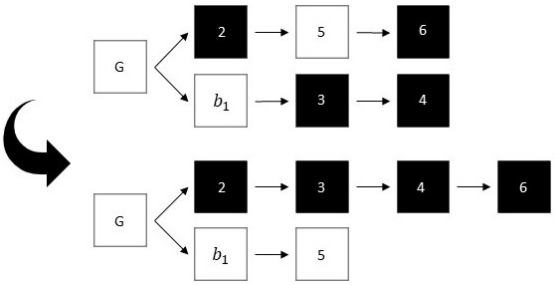

Figure 5: Blocktrees built under an arbitrary attack π and the private attack by time t are given at the top and bottom respectively. Colors black and white represent the adversary and the honest blocks, and, the blocks are labeled by the mining order. Here, $k = 3, H = 2, A = 4, L(t) = 3, L^*(t) = 2$. Under π, the adversary is successful in attacking b_1 at time t. Under the same mining times, the private attack has 4 blocks in private and the honest chain has 2 blocks. By the time the honest chain grows to 3 blocks, the adversary can kick out b_1 by releasing the private chain.

Now consider the blocktree under the private attack π^* at time t. (Bottom of figure 5.) Since no adversary block is mined on C^* under the private attack, $L^*(t) = H$. The length of the private chain starting at the genesis is exactly $A \geq \max\{H, k\} = \max\{L^*(t), k\}$. If $L^*(t) \geq k$, the block b_1 can be kicked out now as the adversary can release the private chain at this time. On the other hand, if $L^*(t) < k$, the adversary can wait until the public chain grows to length k and then release the private chain, which will be at least of length k. In either case, the private attack is successful in the violation of persistence for b_1.

□

In contrast to the PoW setting, a beautiful example from [Shi19] indicates that private attack is no longer the worst attack for every sequence of arrival times under the Praos/SnowWhite model, even for $\Delta = 0$. Figure 6 explains this example, and exhibits the blocktree partitioning for this example. With only 1/3 as many mining times opportunities to 2/3 for the honest players, the protocol can lose persistence. A private attack would not be able to accomplish the same, because the adversary has less mining opportunities than the honest nodes. This is somewhat surprising, given that the security threshold is 1/2 for this model (at $\Delta = 0$). This also suggests that although the two settings, PoW and Praos/SnowWhite have identical security thresholds, their "true" worst case behaviors, taken over all mining time sequences, are different. The larger number of blocks available to the adversary in the Praos/SnowWhite protocol does have some effect in the true worst-case sense, and this allows the mounting of a more serious attack than a private attack. However, these are very atypical mining time sequences, and this difference does not show up in the security threshold.

So perhaps Nakamoto almost always wins.

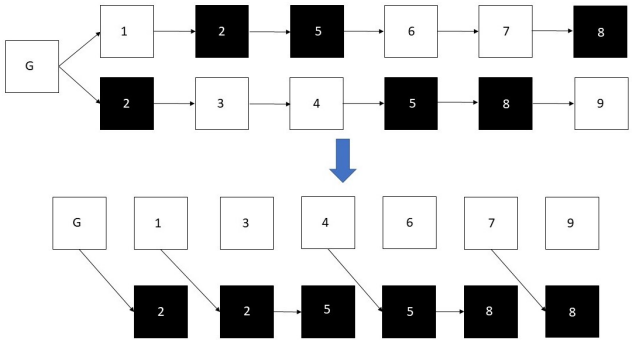

Figure 6: On the top is the blocktree for the example above. Colors black and white represent adversary and honest blocks respectively. The mining time of each block is stated on it. On the bottom is the partition of the blocktree into honest blocks and adversary chains, verifying that indeed there are no Nakamoto blocks. The adversary mines two blocks every third mining time and gets two copies of it. By publishing the shallower block and keeping the deeper block in private and having the honest nodes mine on the shallower block, it can continue the balance attack indefinitely. This attacks relies on a periodic arrival pattern of the blocks. In a random environment, this pattern cannot hold indefinitely and the attack is not sustainable. So randomness saves Praos/SnowWhite.

6 ACKNOWLEDGMENTS

Amir Dembo and Ofer Zeitouni were partially supported by a US-Israel BSF grant. Ertem Nusret Tas was supported in part by the Stanford Center for Blockchain Research. This research is also

supported in part by NSF under grants CCF-1705007, DMS-1954337, 1651236 and Army Research Office under grant W911NF-14-1-0220. We thank the reviewers for the helpful comments.

REFERENCES

[AAC+17] Hamza Abusalah, Joël Alwen, Bram Cohen, Danylo Khilko, Krzysztof Pietrzak, and Leonid Reyzin. Beyond hellman's time-memory trade-offs with applications to proofs of space. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 357–379. Springer, 2017.

[Aïd13] Elie Aïdékon. Convergence in law of the minimum of a branching random walk. *The Annals of Probability*, 41(3A):1362–1426, 2013.

[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *Annual international cryptology conference*, pages 757–788. Springer, 2018.

[BDK+19] Vivek Bagaria, Amir Dembo, Sreeram Kannan, Sewoong Oh, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Proof-of-stake longest chain protocols: Security vs predictability. *arXiv preprint arXiv:1910.02218*, 2019.

[BGK+18] Christian Badertscher, Peter Gaži, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 913–930. ACM, 2018.

[BPS16] Iddo Bentov, Rafael Pass, and Elaine Shi. Snow white: Provably secure proofs of stake. *IACR Cryptology ePrint Archive*, 2016:919, 2016.

[CP19] Bram Cohen and Krzysztof Pietrzak. The chia network blockchain. *https://www.chia.net/assets/ChiaGreenPaper.pdf*, 2019.

[DFKP15] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of space. In *Annual Cryptology Conference*, pages 585–605. Springer, 2015.

[DGKR18] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 66–98. Springer, 2018.

[DKT+20] Amir Dembo, Sreeram Kannan, Ertem Nusret Tas, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Everything is a race and nakamoto always wins. *arXiv preprint arXiv:2005.10484*, 2020.

[Drm09] Michael Drmota. The height of increasing trees. *Annals of Combinatorics*, 12(4):373–402, 2009.

[FZ18] Lei Fan and Hong-Sheng Zhou. A scalable proof-of-stake blockchain in the open setting (or, how to mimic nakamoto's design via proof-of-stake), 2018. Cryptology ePrint Archive, Report 2017/656, Version 20180425:201821.

[GKL15] 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.

[GKR20] Peter Gaži, Aggelos Kiayias, and Alexander Russell. Tight consistency bounds for bitcoin. Cryptology ePrint Archive, Report 2020/661, 2020. https://eprint.iacr.org/2020/661.

[HS09] Yueyun Hu and Zhan Shi. Minimal position and critical martingale convergence in branching random walks, and directed polymers on disordered trees. *The Annals of Probability*, 37(2):742–789, 2009.

[KQR20] Aggelos Kiayias, Saad Quader, and Alexander Russell. Consistency of proof-of-stake blockchains with concurrent honest slot leaders. *arXiv preprint arXiv:2001.06403*, 2020.

[KRDO17] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

[LG20] Jing Li and Dongning Guo. Continuous-time analysis of the bitcoin and prism backbone protocols. *arXiv preprint arXiv:2001.05644*, 2020.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[Pie18] Krzysztof Pietrzak. Simple verifiable delay functions. In *10th innovations in theoretical computer science conference (itcs 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[Pit94] Boris Pittel. Notes on the heights of random recursive trees and random m-ary search trees. *Random Structures Alg.*, 5:337–347, 1994.

[PKF+18] Sunoo Park, Albert Kwon, Georg Fuchsbauer, Peter Gaži, Joël Alwen, and Krzysztof Pietrzak. Spacemint: A cryptocurrency based on proofs of space. In *International Conference on Financial Cryptography and Data Security*, pages 480–499. Springer, 2018.

[PS17] Rafael Pass and Elaine Shi. The sleepy model of consensus. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 380–409. Springer, 2017.

[PSS17] R Pass, L Seeman, and A Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2017.

[Ren19] Ling Ren. Analysis of nakamoto consensus. Technical report, Cryptology ePrint Archive, Report 2019/943.(2019). https://eprint. iacr. org …, 2019.

[Shi15] Zhan Shi. *Branching Random Walks*, volume 2151 of *Lecture Notes in Mathematics*. Springer Verlag, New York NY, 2015.

[Shi19] Elaine Shi. Analysis of deterministic longest-chain protocols. In *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*, pages 122–12213. IEEE, 2019.

[SZ15] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 507–527. Springer, 2015.

[SZ16] Yonatan Sompolinsky and Aviv Zohar. Bitcoin's security model revisited. *arXiv preprint arXiv:1605.09193*, 2016.

APPENDIX

A DEFINITIONS AND PRELIMINARY LEMMAS FOR THE PROOFS

In this section, we define some important events which will appear frequently in the analysis and provide some useful lemmas.

Let $\delta_i^h = \tau_i^h - \tau_{i-1}^h$ and $\delta_i^a = \tau_i^a - \tau_{i-1}^a$ denote the time intervals for subsequent honest and adversary arrival events. Let d_i^h denote the depth of the i-th honest block within $D_h(t)$. Define X_d, $d > 0$, as the time it takes for D_h to reach depth d after reaching depth $d-1$. In other words, X_d is the difference between the times $t_1 > t_2$, where t_1 is the minimum time t such that $D_h(t) = d$, and, t_2 is the minimum time t such that $D_h(t) = d-1$.

Let U_j be the event that the j-th honest block b_j is a loner, i.e.,

$$U_j = \{\tau_{j-1}^h < \tau_j^h - \Delta\} \cap \{\tau_{j+1}^h > \tau_j^h + \Delta\}.$$

Let $\hat{F}_j = U_j \cap F_j$ be the event that b_j is a Nakamoto block. Then we can define the following catch up event:

$$\hat{B}_{ik} = \text{event that } D_i(\tau_k^h + \Delta) \geq D_h(\tau_{k-1}^h) - D_h(\tau_i^h + \Delta), \quad (22)$$

which is the event that the adversary launches a private attack starting from b_i and catches up the fictitious honest chain right before b_k is mined. The following lemma shows that event $\hat{F}_j$ can be represented with $\hat{B}_{ik}$'s.

Lemma A.1. *For each j,*

$$\hat{F}_j^c = F_j^c \cup U_j^c = \left(\bigcup_{(i,k):0\leq i<j<k} \hat{B}_{ik} \right) \cup U_j^c. \quad (23)$$

Proof.

$$\begin{aligned}
& U_j \cap E_{ij} \\
= \; & U_j \cap \{D_i(t) < D_h(t-\Delta) - D_h(\tau_i^h + \Delta) \text{ for all } t > \tau_j^h + \Delta\} \\
= \; & U_j \cap \{D_i(t+\Delta) < D_h(t) - D_h(\tau_i^h + \Delta) \text{ for all } t > \tau_j^h\} \\
= \; & U_j \cap \{D_i(\tau_k^{h^-} + \Delta) < D_h(\tau_k^{h^-}) - D_h(\tau_i^h + \Delta) \text{ for all } k > j\} \\
= \; & U_j \cap \{D_i(\tau_k^h + \Delta) < D_h(\tau_{k-1}^h) - D_h(\tau_i^h + \Delta) \text{ for all } k > j\}
\end{aligned}$$

Since $\hat{F}_j = F_j \cap U_j = \bigcap_{0\leq i<j} E_{ij} \cap U_j$, by the definition of $\hat{B}_{ik}$ we have $\hat{F}_j = \left(\bigcap_{(i,k):0\leq i<j<k} \hat{B}_{ik}^c \right) \cap U_j$. Taking complement on both side, we can conclude the proof.

Finally, define the parameter r as follows:

$$r := \frac{\lambda_a}{\lambda_h}(1 + \Delta\lambda_h),$$

for which $r < 1$ holds whenever

$$\lambda_a < \frac{\lambda_h}{1 + \Delta\lambda_h}.$$

□

B PROOF OF THEOREM 3.2

Notation used in this section is defined in section 3.

For the proof of the stabilization property of a Nakamoto block, it is crucial to show that $D_h(t)$ gives a conservative bound on the growth of the chains $C^{(p)}$ from time s to t. For this purpose, we prove the following proposition:

Proposition B.1. *For any given s, t such that $s + \Delta < t - \Delta$;*

$$D_h(t - \Delta) - D_h(s + \Delta) \leq L^{(p)}(t) - L^{(p)}(s)$$

for any honest miner p.

Proof. Assume that the increase in $L^{(p)}$ within the interval $[s, t]$ is solely due to the arrival of honest blocks to some miner in the interval $[s-\Delta, t]$. Then, we first show that delaying every block that arrives within this interval by Δ minimizes the increase in $L^{(p)}$ from s to t for any $t > s + \Delta$. To prove this, first observe that minimizing the increase in $L^{(p)}$ is equivalent to maximizing the time it takes for $C^{(p)}$ to reach any depth d. Now, let h_i be the block at the tip of $C^{(p)}$ when it reaches depth d, and, assume that it took $\delta_i \leq \Delta$ time for p to learn about h_i after it was mined. Then, $C^{(p)}$ reaches depth d at time $\tau_i^h + \delta_i$. However, if the message for h_i was delayed for $\delta_i' > \delta_i$ time, then, either $C^{(p)}$ would have reached depth d at time $\tau_i^h + \delta_i' \geq \tau_i^h + \delta_i$ with block h_i at its tip, or, another block h_j, with index $j \neq i$ would have brought $C^{(p)}$ to depth d at some time t, $\tau_i^h + \delta_i' > t > \tau_i^h + \delta_i$. Hence, delaying the transmission of h_i increases the time it takes for $C^{(p)}$ to reach depth d. This implies that h_i should be delayed as long as possible, which is Δ. Since this argument also applies to any other block h_j that might also bring $C^{(p)}$ to depth d when h_i is delayed, every block should be delayed by Δ to maximize the time for $C^{(p)}$ to reach any depth d. This, in turn, minimizes the increase in $L^{(p)}$ by any time $t > s$.

Next, define the following random variable:

$$L_{max}(t) = \max_{p=1,..,n} (L^{(p)}(t)).$$

Then, we can assert that;

$$L_{max}(t - \Delta) \leq L^{(p)}(t) \leq L_{max}(t)$$

for any honest miner p. Then,

$$L^{(p)}(t) - L^{(p)}(s) \geq L_{max}(t - \Delta) - L_{max}(s).$$

From the paragraph above, we know that delaying every honest block by Δ minimizes $L^{(p)}(t)$ for any t. Hence, this action also minimizes $L^{(p)}(t) - L^{(p)}(s)$ for any $t > s + 2\Delta$. Now, assume that no honest miner hears about any adversary block in the interval $[s, t]$ and every honest block is delayed by Δ. Then, the difference $L_{max}(t - \Delta) - L_{max}(s)$ will be solely due the honest blocks that arrive within the period $[s, t - \Delta]$. However, in this case, depth of L_{max} changes via the same process as D_h (when each miner has infinitesimal power), which implies the following inequality:

$$L_{max}(t - \Delta) - L_{max}(s) \geq D_h(t - \Delta) - D_h(s + \Delta).$$

Hence, we see that when every block is delayed by Δ and there are no adversary blocks heard by p in the time interval $[s, t]$;

$$L^{(p)}(t) - L^{(p)}(s) \geq D_h(t - \Delta) - D_h(s + \Delta).$$

However, delaying honest blocks less than Δ time or the arrival of adversary blocks to p in the period $[s, t]$ only increases the difference $L^{(p)}(t) - L^{(p)}(s)$. Consequently;

$$D_h(t - \Delta) - D_h(s + \Delta) \leq L^{(p)}(t) - L^{(p)}(s)$$

for any honest miner p. □

Now we are ready to prove Theorem 3.2.

Proof. We prove that the j-th honest block will be included in any future chain $C(t)$ for $t > \tau_i^h + \Delta$, by contradiction. Suppose $\hat{F}_j$ occurs and let $t^* > \tau_j^h + \Delta$ be the smallest t such that the j-th honest block is not contained in $C^{(p)}(t)$ for some $1 \leq p \leq n$. Let h_i be the last honest block on $C^{(p)}(t^*)$, which must exist, because the genesis block is by definition honest. If $\tau_i^h > \tau_j^h + \Delta$ for h_i, then, $C^{(p)}(\tau_i^{h-})$ is the prefix of $C^{(p)}(t^*)$ before block h_i, and, does not contain the j-th honest block, contradicting the minimality of t^*. Therefore, h_i must be mined before time $\tau_j^h + \Delta$. Since the j-th honest block is a loner, we further know that h_i must be mined before time τ_j^h, implying that h_i is the i-th honest block for some $i < j$. In this case, part of $C^{(p)}(t^*)$ after block h_i must lie entirely in the tree $\mathcal{T}_i(t^*)$ rooted at h_i. Hence,

$$L^{(p)}(t^*) \leq L^{(p)}(\tau_i^h) + D_i(t^*). \tag{24}$$

However, we know that;

$$D_i(t^*) < D_h(t^* - \Delta) - D_h(\tau_i^h + \Delta) \leq L^{(p)}(t^*) - L^{(p)}(\tau_i^h) \tag{25}$$

where the first inequality follows from the fact that $\hat{F}_j$ holds and the second inequality follows from Proposition B.1. From this we obtain that

$$L^{(p)}(\tau_i^h) + D_i(t^*) < L^{(p)}(t^*) \tag{26}$$

which is a contradiction since $L^{(p)}(t^*) \leq L^{(p)}(\tau_i^h) + D_i(t^*)$. This concludes the proof. □

C PROOFS FOR SECTION 4.3

Notations used in this section are defined in §A.

Subsequent propositions are used in future proofs.

Proposition C.1. *Let Y_d, $d \geq 1$, be i.i.d random variables, exponentially distributed with rate λ_h. Then, each random variable X_d can be expressed as $\Delta + Y_d$.*

The proof of Proposition C.1 is given in §C of [DKT+20].

Proposition C.2. *For any constant a,*

$$P(\sum_{d=a}^{n+a} X_d > n(\Delta + \frac{1}{\lambda_h})(1 + \delta)) \leq e^{-n\Omega(\delta^2(1+\Delta\lambda_h)^2)}.$$

Proposition C.2 is proven using a Chernoff bound analysis and Proposition C.1.

Proposition C.3. *Probability that there are less than $n\frac{\lambda_a(1-\delta)}{\lambda_h}$ adversary arrival events from time τ_0^h to τ_{n+1}^h is upper bounded by $e^{-n\Omega(\delta^2 \frac{\lambda_a}{\lambda_h})}$.*

Proposition C.3 is proven using the Poisson tail bounds.

Proposition C.4. *Define B_n as the event that there are at least n adversary arrivals while D_h grows from depth 0 to n:*

$$B_n = \{\sum_{i=1}^{n} X_i \geq \sum_{i=0}^{n} \delta_i^a\}$$

If

$$\lambda_a < \frac{\lambda_h}{1+\lambda_h\Delta},$$

then,

$$P(B_n) \leq e^{-A_0 n},$$

where,

$$A_0 = s\Delta + \ln(\frac{\lambda_a\lambda_h}{(\lambda_h - s)(\lambda_a + s)}) > 0$$

and,

$$s = \frac{\lambda_h - \lambda_a}{2} + \frac{2 - \sqrt{4 + \Delta^2(\lambda_a + \lambda_h)^2}}{2\Delta}.$$

Proof is by using Chernoff bound, and, optimizing for the value of s. It also uses Proposition C.1.

C.1 Proof of Lemma 4.5

The proof is based on random walk theory.

Proof. We would like to lower bound the probability that the j-th honest block is a loner and F_j happens. Since the j-th honest block is a loner with probability $e^{-2\lambda_h\Delta} > 0$ for all j, the probability that it is a Nakamoto block can be expressed as

$$P(F_j \mid \text{j-th honest block is a loner}) \cdot e^{-2\lambda_h\Delta}$$

Then, the proof is reduced to obtaining a lower bound on

$$P(F_j \mid \text{j-th honest block is a loner}).$$

For this purpose, we assume that the j-th honest block is a loner, and, proceed to obtain a lower bound on the probability of the event F_j:

For any adversary tree $\mathcal{T}_i$, $i < j$;

$$D_i(t) < D_h(t-\Delta) - D_h(\tau_i^h + \Delta)$$

for all times $t > \tau_j^h + \Delta$, which is equivalent to

$$D_i(t+\Delta) < D_h(t) - D_h(\tau_i^h + \Delta)$$

for all times $t > \tau_j^h$.

Let U_j be the event that the j-th honest block is a loner. Let G_j be the event that no adversary block is mined within the time period $[\tau_j^h, \tau_j^h + \Delta]$. Then, $P(G_j) = e^{-\lambda_a\Delta}$, and, we can lower bound $P(F_j|U_j)$ in the following way:

$$P(F_j|U_j) \geq P(F_j \cap G_j|U_j) = e^{-\lambda_a\Delta}P(F_j|U_j, G_j)$$

Since the events G_j, $j = 1, 2, ..$ are shift invariant, the probability $P(F_j|U_j, G_j)$ is equal to the probability of the following event $\hat{F}_j$:

For any adversary tree $\mathcal{T}_i$, $i < j$;

$$D_i(t) < D_h(t) - D_h(\tau_i^h + \Delta)$$

for all times $t > \tau_j^h$. Now, define $D^*(t)$ as the depth of the deepest adversary tree at time t for $t \geq \tau_j^h$:

$$D^*(t) := \max_{0 \leq i < j} D_i(t) + D_h(\tau_i^h + \Delta)$$

Then, $\hat{F}_j$ basically represents the event that D^* is behind D_h for all times $t \geq \tau_j^h$.

We next express $\hat{F}_j$ in terms of the following events:

$$E_1 := \{D^*(\tau_j^h) < D_h(\tau_j^h)\}$$

E_1 is the event that the tip of the deepest adversary tree, D^*, is behind the tip of the honest tree, D_h at the arrival time of the j-th honest block.

E_2 is the event that $D_h(t) - D_h(\tau_j^h)$ is greater than the number of adversary arrivals during the time period $[\tau_j^h, t]$ for all t, $t > \tau_j^h$.

E_3 is the event that $D_h(\tau_j^h) - D_h(\tau_i^h + \Delta)$ is greater than the number of adversary arrivals during the time period $[\tau_i^h, \tau_j^h]$ for all i, $0 \leq i < j$.

We can now express $\hat{F}_j$ in terms of E_1 and E_2:

$$E_1 \cap E_2 \subseteq \hat{F}_j$$

Moreover, when a new adversary block is mined, depth of any of the trees $\mathcal{T}_i$, $i < j$, increases by at most 1. Hence, E_3 implies that none of the trees $\mathcal{T}_i$, $i < j$, has depth greater than or equal to $D_h(\tau_j^h)$ at time τ_j^h. Consequently,

$$E_3 \subseteq E_1,$$

which further implies

$$E_3 \cap E_2 \subseteq \hat{F}_j$$

Observing that E_3 and E_2 are independent events, we can express the probability of $\hat{F}_j$ as;

$$P(\hat{F}_j) \geq P(E_3)P(E_2)$$

Now, define E_2' as the event that $D_h(t-\Delta) - D_h(\tau_j^h)$ is greater than the number of adversary arrivals during the time period $[\tau_j^h, t]$ for all t, $t > \tau_j^h + \Delta$. Let G_j' be the event that there is no adversary arrival during the time interval $[\tau_j^h, \tau_j^h + \Delta]$. Observe that again, $P(G_j') = e^{-\lambda_a\Delta}$, and, the events G_j', $j = 1, 2, ..$ are shift invariant. Hence, we can do a similar trick as was done for the probabilities of F_j and G_j to obtain

$$P(E_2') \geq e^{-\lambda_a\Delta}P(E_2).$$

Since the increase times of D_h and the inter-arrival times of adversary arrivals are i.i.d, the growth processes of D_h and the number of adversary blocks are time reversible. Hence, probability of E_3 approaches that of E_2' from above as $j \to \infty$. Then, for all j, we can write

$$P(\hat{F}_j) \geq P(E_3)P(E_2) \geq P(E_2')P(E_2) \geq e^{-\lambda_a\Delta}P(E_2)^2$$

We now calculate the probability of the event E_2. To aid us in the calculation of $P(E_2)$, we construct a random walk $S[n]$. Here, the random walk is parametrized by the total number of adversary arrivals and increases in D_h since time τ_j^h. $S[n]$ stands for the difference between the increase in D_h and the number of adversary

arrivals when there has been, in total, n number of increases in D_h or adversary arrivals since time τ_j^h. Notice that when $\Delta = 0$, D_h increases by one whenever there is an honest arrival. Hence, $S[n]$ simply counts the difference between the number of honest and adversary arrivals when there are n arrivals in total. In this case, $S[n]$ jumps up by 1 when there is an honest arrival, and, goes down by 1 when there is an adversary arrival. Since the event that whether the next arrival is honest or adversary is independent of the past arrivals, $S[n]$ is a random walk when $\Delta = 0$.

On the other hand, when $\Delta > 0$, we have to construct a slight different random walk $S[n]$ for the difference between the increase in D_h and the number of adversary arrivals due to the Δ dependence. Although this random walk has non-intuitive distributions for the jumps, we observe that

(1) Expectation of these jumps is positive as long as

$$\lambda_a < \frac{\lambda_h}{1 + \lambda_h \Delta}$$

(2) Expectation of the absolute value of the jumps is finite.

Then, due to the Strong Law of Large Numbers, every state of this random walk is transient, and, the random walk has a positive drift. This implies that starting at $S[0] = 1$, the probability of $S[n]$ hitting or falling below 0 is equal to some number $1 - c$, where $1 \geq c > 0$.

Finally, observe that the probability of $S[n]$ hitting or falling below 0 is exactly the probability of the event E_2^c. Hence, $P(E_2) = c > 0$. Combining this observation with previous findings yields the following lower bound for $P(F_j|U_j)$:

$$P(F_j|U_j) \geq e^{-\lambda_a \Delta} P(F_j|U_j, G_j) \geq e^{-2\lambda_a \Delta} P(E_2)^2 = e^{-2\lambda_a \Delta} c^2 = p > 0$$

where $p > 0$ does not depend on j. This concludes the proof. □

C.2 Proof of Lemma 4.6

We first state the following lemma which will be used in the proof of Lemma 4.6. Recall that we have defined event $\hat{B}_{ik}$ in §A as:

$$\hat{B}_{ik} = \text{event that } D_i(\tau_k^h + \Delta) \geq D_h(\tau_{k-1}^h) - D_h(\tau_i^h + \Delta). \quad (27)$$

Lemma C.5. *There exists a constant $c > 0$ such that*

$$P(\hat{B}_{ik}) \leq e^{-c(k-i-1)}$$

Proof. We know from Proposition C.3 that there are more than $(1-\delta)(k-i)\lambda_a/\lambda_h$ adversary arrival events in the time period $[\tau_i^h, \tau_k^h + \Delta]$ except with probability $e^{-\Omega((k-i)\delta^2\lambda_a/\lambda_h)}$. Moreover, Proposition C.4 states that

$$P(\sum_{i=1}^{n} X_i \geq \sum_{i=0}^{n} \delta_i^a) \leq e^{-A_0 n}$$

for large n. Then, using the union bound, we observe that for any fixed δ, probability of $\hat{B}_{ik}$ when there are more than $(1-\delta)(k-i)\lambda_a/\lambda_h$ adversary arrival events in the time period $[\tau_i^h, \tau_k^h + \Delta]$ is upper bounded by the following expression:

$$\frac{1}{1 - e^{-C_1}} e^{-C_1(k-i)}$$

where

$$C_1 = \frac{A_0(1-\delta)\lambda_a}{\lambda_h}.$$

Hence,

$$P(\hat{B}_{ik}) < \frac{1}{1 - e^{-C_1}} e^{-C_1(k-i)} + e^{-\Omega((k-i)\delta^2 \frac{\lambda_a}{\lambda_h})} \leq C_2 e^{-C_3(k-i)}$$

for any $k, i, k > i+1$, and appropriately chosen constants $C_2, C_3 > 0$ as functions of the fixed δ. Finally, since $P(\hat{B}_{ik})$ decreases as $k - i$ grows and is smaller than 1 for all $k > i + 1$, we obtain the desired inequality for a sufficiently small $c \leq C_3$. □

We can now proceed with the main proof.

We divide the proof in to two steps. In the first step, we prove for $\varepsilon = 1/2$. By Lemma A.1, we have

$$\hat{F}_j^c = F_j^c \cup U_j^c = \left(\bigcup_{(i,k): i<j<k} \hat{B}_{ik} \right) \cup U_j^c. \quad (28)$$

Divide $[s, s+t]$ into $\sqrt{t}$ sub-intervals of length $\sqrt{t}$, so that the r th sub-interval is:

$$\mathcal{J}_r := [s + (r-1)\sqrt{t}, s + r\sqrt{t}].$$

Now look at the first, fourth, seventh, etc sub-intervals, i.e. all the $r = 1 \mod 3$ sub-intervals. Introduce the event that in the ℓ-th $1 \mod 3$th sub-interval, an adversary tree that is rooted at a honest block arriving in that sub-interval or in the previous $(0 \mod 3)$ sub-interval catches up with a honest block in that sub-interval or in the next $(2 \mod 3)$ sub-interval. Formally,

$$C_\ell = \bigcap_{j: \tau_j^h \in \mathcal{J}_{3\ell+1}} U_j^c \cup \left(\bigcup_{(i,k): \tau_j^h - \sqrt{t} < \tau_i^h < \tau_j^h, \tau_j^h < \tau_k^h + \Delta < \tau_j^h + \sqrt{t}} \hat{B}_{ik} \right).$$

Note that for distinct ℓ, the events C_ℓ's are independent. Also, we have

$$P(C_\ell) \leq P(\text{no arrival in } \mathcal{J}_{3\ell+1}) + 1 - p < 1 \quad (29)$$

for large enough t, where p is a uniform lower bound such that $P(\hat{F}_j) \geq p$ for all j provided by Lemma 4.5.

Introduce the atypical events:

$$B = \bigcup_{(i,k): \tau_i^h \in [s,s+t] \text{ or } \tau_k^h + \Delta \in [s,s+t], i<k, \tau_k^h + \Delta - \tau_i^h > \sqrt{t}} \hat{B}_{ik},$$

and

$$\tilde{B} = \bigcup_{(i,k): \tau_i^h < s, s+t < \tau_k^h + \Delta} \hat{B}_{ik}.$$

The events B and $\tilde{B}$ are the events that an adversary tree catches up with an honest block far ahead. Then we have

$$\begin{aligned} P(B_{s,s+t}) &\leq P\Big(\bigcap_{j: \tau_j^h \in [s,s+t]} U_j^c \Big) + P(B) + P(\tilde{B}) + P\Big(\bigcap_{\ell=0}^{\sqrt{t}/3} C_\ell \Big) \\ &= P\Big(\bigcap_{j: \tau_j^h \in [s,s+t]} U_j^c \Big) + P(B) + P(\tilde{B}) + (P(C_\ell))^{\sqrt{t}/3} \\ &\leq e^{-c_2 t} + P(B) + P(\tilde{B}) + (P(C_\ell))^{\frac{\sqrt{t}}{3}} \end{aligned} \quad (30)$$

for some positive constant c_2 when t is large, where the equality is due to independence. Next we will bound the atypical events B and

$\tilde{B}$. Consider the following events

$$D_1 = \{\#\{i : \tau_i^h \in (s-\sqrt{t}-\Delta, s+t+\sqrt{t}+\Delta)\} > 2\lambda_h t\}$$

$$D_2 = \{\exists i,k : \tau_i^h \in (s,s+t), (k-i) < \frac{\sqrt{t}}{2\lambda_h}, \tau_k^h - \tau_i^h + \Delta > \sqrt{t}\}$$

$$D_3 = \{\exists i,k : \tau_k^h + \Delta \in (s,s+t), (k-i) < \frac{\sqrt{t}}{2\lambda_h}, \tau_k^h - \tau_i^h + \Delta > \sqrt{t}\}$$

In words, D_1 is the event of atypically many honest arrivals in $(s-\sqrt{t}-\Delta, s+t+\sqrt{t}+\Delta)$ while D_2 and D_3 are the events that there exists an interval of length $\sqrt{t}$ with at least one endpoint inside $(s,s+t)$ with atypically small number of arrivals. Since the number of honest arrivals in $(s,s+t)$ is Poisson with parameter $\lambda_h t$, we have from the memoryless property of the Poisson process that $P(D_1) \le e^{-c_0 t}$ for some constant $c_0 = c_0(\lambda_a, \lambda_h) > 0$ when t is large. On the other hand, using the memoryless property and a union bound, and decreasing c_0 if needed, we have that $P(D_2) \le e^{-c_0\sqrt{t}}$. Similarly, using time reversal, $P(D_3) \le e^{-c_0\sqrt{t}}$. Therefore, again using the memoryless property of the Poisson process,

$$\begin{aligned} P(B) &\le P(D_1 \cup D_2 \cup D_3) + P(B \cap D_1^c \cap D_2^c \cap D_3^c) \\ &\le e^{-c_0 t} + 2e^{-c_0\sqrt{t}} + \sum_{i=1}^{2\lambda_h t} \sum_{k:k-i>\sqrt{t}/2\lambda_h} P(\hat{B}_{ik}) \quad (31) \\ &\le e^{-c_3\sqrt{t}}, \quad (32) \end{aligned}$$

for large t, where $c_3 > 0$ are constants that may depend on λ_a, λ_h and the last inequality is due to Lemma C.5 . We next claim that there exists a constant $\alpha > 0$ such that, for all t large,

$$P(\tilde{B}) \le e^{-\alpha t}. \quad (33)$$

Indeed, we have that

$$\begin{aligned} &P(\tilde{B}) \\ &= \sum_{i<k} \int_0^s P(\tau_i^h \in d\theta) P(\hat{B}_{ik}, \tau_k^h - \tau_i^h + \Delta > s+t-\theta) \\ &\le \sum_i \int_0^s P(\tau_i^h \in d\theta) \sum_{k:k>i} P(\hat{B}_{ik})^{1/2} P(\tau_k^h - \tau_i^h + \Delta > s+t-\theta)^{1/2}. \end{aligned} \quad (34)$$

The tails of the Poisson distribution yield the existence of constants $c, c' > 0$ so that

$$\begin{aligned} &P(\tau_k^h - \tau_i^h + \Delta > s+t-\theta) \quad (35) \\ &\le \begin{cases} 1, & (k-i) > c(s+t-\theta-\Delta) \\ e^{-c'(s+t-\theta-\Delta)}, & (k-i) \le c(s+t-\theta-\Delta). \end{cases} \quad (36) \end{aligned}$$

Lemma C.5 and (35) yield that there exists a constant $\alpha > 0$ so that

$$\sum_{k:k>i} P(\hat{B}_{i,k})^{1/2} P(\tau_k^h - \tau_i^h > s+t-\theta-\Delta)^{1/2} \le e^{-2\alpha(s+t-\theta-\Delta)}. \quad (37)$$

Substituting this bound in (34) and using that $\sum_i P(\tau_i^h \in d\theta) = d\theta$ gives

$$\begin{aligned} P(\tilde{B}) &\le \sum_i \int_0^s P(\tau_i^h \in d\theta) e^{-2\alpha(s+t-\theta-\Delta)} \\ &\le \int_0^s e^{-2\alpha(s+t-\theta-\Delta)} d\theta \le \frac{1}{2\alpha} e^{-2\alpha(t-\Delta)} \le e^{-\alpha t}, \quad (38) \end{aligned}$$

for t large, proving (33).

Combining (32), (38) and (30) concludes the proof of step 1.

In step two, we prove for any $\varepsilon > 0$ by recursively applying the bootstrapping procedure in step 1. Assume the following statement is true: for any $\theta \ge m$ there exist constants $\bar{a}_\theta, \bar{A}_\theta$ so that for all $s, t \ge 0$,

$$\tilde{q}[s, s+t] \le \bar{A}_\theta \exp(-\bar{a}_\theta t^{1/\theta}). \quad (39)$$

By step 1, it holds for $m = 2$.

Divide $[s, s+t]$ into $t^{\frac{m-1}{2m-1}}$ sub-intervals of length $t^{\frac{m}{2m-1}}$, so that the r th sub-interval is:

$$\mathcal{J}_r := [s + (r-1)t^{\frac{m}{2m-1}}, s + rt^{\frac{m}{2m-1}}].$$

Now look at the first, fourth, seventh, etc sub-intervals, i.e. all the $r = 1 \mod 3$ sub-intervals. Introduce the event that in the ℓ-th $1 \mod 3$th sub-interval, an adversary tree that is rooted at a honest block arriving in that sub-interval or in the previous $(0 \mod 3)$ sub-interval catches up with a honest block in that sub-interval or in the next $(2 \mod 3)$ sub-interval. Formally,

$$C_\ell = \bigcap_{j:\tau_j^h \in \mathcal{J}_{3\ell+1}} U_j^c \cup \left(\bigcup_{(i,k):\tau_j^h - t^{\frac{m}{2m-1}} < \tau_i^h < \tau_j^h, \tau_j^h < \tau_k^h + \Delta < \tau_j^h + t^{\frac{m}{2m-1}}} \hat{B}_{ik} \right).$$

Note that for distinct ℓ, the events C_ℓ's are independent. Also by (39), we have

$$P(C_\ell) \le A_m \exp(-\bar{a}_m t^{1/(2m-1)}). \quad (40)$$

Introduce the atypical events:

$$B = \bigcup_{(i,k):\tau_i^h \in [s,s+t] \text{ or } \tau_k^h + \Delta \in [s,s+t], i<k, \tau_k^h + \Delta - \tau_i^h > t^{\frac{m}{2m-1}}} \hat{B}_{ik},$$

and

$$\tilde{B} = \bigcup_{(i,k):\tau_i^h < s, s+t < \tau_k^h + \Delta} \hat{B}_{ik}.$$

The events B and $\tilde{B}$ are the events that an adversary tree catches up with an honest block far ahead. Following the calculations in step 1, we have

$$\begin{aligned} P(B) &\le e^{-c_1 t^{\frac{m}{2m-1}}} \quad (41) \\ P(\tilde{B}) &\le e^{-\alpha t}, \quad (42) \end{aligned}$$

for large t, where c_1 and α are some positive constant.

Then we have

$$\begin{aligned} \tilde{q}[s, s+t] &\le P\Big(\bigcap_{j:\tau_j^h \in [s,s+t]} U_j^c\Big) + P(B) + P(\tilde{B}) + P\Big(\bigcap_{\ell=0}^{t^{\frac{m-1}{2m-1}}/3} C_\ell\Big) \\ &= P\Big(\bigcap_{j:\tau_j^h \in [s,s+t]} U_j^c\Big) + P(B) + P(\tilde{B}) + (P(C_\ell))^{t^{\frac{m-1}{2m-1}}/3} \\ &\le e^{-c_2 t} + e^{-ct^{\frac{m}{2m-1}}} + e^{-\alpha t} \\ &\quad + (A_m \exp(-\bar{a}_m t^{1/(2m-1)}))^{t^{\frac{m-1}{2m-1}}/3} \\ &\le \bar{A}'_m \exp(-\bar{a}'_m t^{\frac{m}{2m-1}}) \quad (43) \end{aligned}$$

for large t, where A'_m and a'_m are some positive constant.

So we know the statement in (39) holds for all $\theta \geq \frac{2m-1}{m}$. Start with $m_1 = 2$, we have a recursion equation $m_k = \frac{2m_{k-1}-1}{m_{k-1}}$ and we know (39) holds for all $\theta \geq m_k$. It is not hard to see that $m_k = \frac{k+1}{k}$ and thus $\lim_{k\to\infty} m_k = 1$, which concludes the lemma.

D PROOFS FOR SECTION 4.4

Notations used in this section are defined in §A.

D.1 The adversary tree via branching random walks

We first give a description of the (dual of the) adversary tree in terms of a Branching Random Walk (BRW). Such a representation appears already in [Pit94, Drm09], but we use here the standard language from, e.g., [Aïd13, Shi15].

Consider the collection of k tuples of positive integers, $\mathcal{I}_k = \{(i_1, \ldots, i_k)\}$, and set $\mathcal{I} = \cup_{k>0}\mathcal{I}_k$. We consider elements of $\mathcal{I}$ as labelling the vertices of a rooted infinite tree, with $\mathcal{I}_k$ labelling the vertices at generation k as follows: the vertex $v = (i_1, \ldots, i_k) \in \mathcal{I}_k$ is the i_k-th child of vertex $(i_1, \ldots, i_{k-1})$ at level $k-1$. An example of labelling is given in Figure 7. For such v we also let $v^j = (i_1, \ldots, i_j)$, $j = 1, \ldots, k$, denote the ancestor of v at level j, with $v^k = v$. For notation convenience, we set $v^0 = 0$ as the root of the tree.

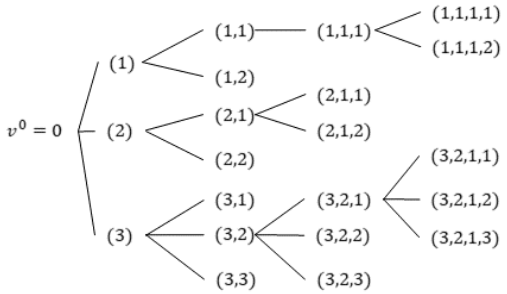

Figure 7: Labelling the vertices of a rooted infinite tree.

Next, let $\{\mathcal{E}_v\}_{v\in\mathcal{I}}$ be an i.i.d. family of exponential random variables of parameter λ_a. For $v = (i_1, \ldots, i_k) \in \mathcal{I}_k$, let $\mathcal{W}_v = \sum_{j\leq i_k} \mathcal{E}_{(i_1,\ldots,i_{k-1},j)}$ and let $S_v = \sum_{j\leq k} \mathcal{W}_{v^j}$. This creates a labelled tree, with the following interpretation: for $v = (i_1, \ldots, i_j)$, the W_{v^j} are the waiting for v^j to appear, measured from the appearance of v^{j-1}, and S_v is the appearance time of v. A moments thought ought to convince the reader that the tree S_v is a description of the adversary tree, sorted by depth.

Let $S_k^* = \min_{v\in\mathcal{I}_k} S_v$. Note that S_k^* is the time of appearance of a block at level k and therefore we have

$$\{D_0(t) \leq k\} = \{S_k^* \geq t\}. \tag{44}$$

S_k^* is the minimum of a standard BRW. Introduce, for $\theta < 0$, the moment generating function

$$\begin{aligned}\Lambda(\theta) &= \log \sum_{v\in\mathcal{I}_1} E(e^{\theta S_v}) = \log \sum_{j=1}^{\infty} E(e^{\sum_{i=1}^{j}\theta\mathcal{E}_i}) \\ &= \log \sum_{j=1}^{\infty} (E(e^{\theta\mathcal{E}_1}))^j = \log \frac{E(e^{\theta\mathcal{E}_1})}{1 - E(e^{\theta\mathcal{E}_1})}.\end{aligned}$$

Due to the exponential law of $\mathcal{E}_1$, $E(e^{\theta\mathcal{E}_1}) = \frac{\lambda_a}{\lambda_a - \theta}$ and therefore $\Lambda(\theta) = \log(-\lambda_a/\theta)$.

An important role is played by $\theta^* = -e\lambda_a$, for which $\Lambda(\theta^*) = -1$ and

$$\sup_{\theta<0}\left(\frac{\Lambda(\theta)}{\theta}\right) = \frac{\Lambda(\theta^*)}{\theta^*} = \frac{1}{\lambda_a e} = \frac{1}{|\theta^*|}.$$

Indeed, see e.g [Shi15, Theorem 1.3], we have the following.

Lemma D.1.

$$\lim_{k\to\infty} \frac{S_k^*}{k} = \sup_{\theta<0}\left(\frac{\Lambda(\theta)}{\theta}\right) = \frac{1}{|\theta^*|}, \quad a.s.$$

In fact, much more is known, see e.g. [HS09].

Lemma D.2. *There exist explicit constants $c_1 > c_2 > 0$ so that the sequence $S_k^* - k/\lambda_a e - c_1 \log k$ is tight, and*

$$\liminf_{k\to\infty} S_k^* - k/\lambda_a e - c_2 \log k = \infty, a.s.$$

Note that Lemmas D.1, D.2 and (44) imply in particular that $D_0(t) \leq e\lambda_a t$ for all large t, a.s., and also that

$$\text{if } e\lambda_a > \lambda_h \text{ then } D_0(t) > \lambda_h t \text{ for all large } t, \text{ a.s.}. \tag{45}$$

With all these preparations, we can give a simple proof for Lemma 4.7.

Proof. We use a simple upper bound. Note that by (44),

$$P(D_0(t) \geq m) = P(S_m^* \leq t) \leq \sum_{v\in\mathcal{I}_m} P(S_v \leq t). \tag{46}$$

For $v = (i_1, \ldots, i_k)$, set $|v| = i_1 + \cdots + i_k$. Then, we have that S_v has the same law as $\sum_{j=1}^{|v|} \mathcal{E}_j$. Thus, by Chebycheff's inequality, for $v \in \mathcal{I}_m$,

$$P(S_v \leq t) \leq Ee^{\theta S_v} e^{-\theta t} = \left(\frac{\lambda_a}{\lambda_a - \theta}\right)^{|v|} e^{-\theta t}. \tag{47}$$

But

$$\sum_{v\in\mathcal{I}_m} \left(\frac{\lambda_a}{\lambda_a - \theta}\right)^{|v|} = \sum_{i_1\geq 1,\ldots,i_m\geq 1} \left(\frac{\lambda_a}{\lambda_a - \theta}\right)^{\sum_{j=1}^{m} i_j} \tag{48}$$

$$= \left(\sum_{i\geq 1}\left(\frac{\lambda_a}{\lambda_a - \theta}\right)^{i}\right)^{m} = \left(-\frac{\theta}{\lambda_a}\right)^{-m}. \tag{49}$$

Combining (47), (48), we have

$$P(D_0(t) \geq m) \leq \left(-\frac{\theta}{\lambda_a}\right)^{-m} e^{-\theta t},$$

and optimizing over θ we have when $\theta = -m/t$,

$$P(D_0(t) \geq m) \leq \left(\frac{e\lambda_a t}{m}\right)^{m}.$$

□

D.2 Proof of Lemma 4.8

In this proof, let $r_h := \frac{\lambda_h}{1+\lambda_h\Delta}$.

The random processes of interest start from time 0. To look at the system in stationarity, let us extend them to $-\infty < t < \infty$. More specifically, define $\tau_{-1}^h, \tau_{-2}^h, \ldots$ such that together with $\tau_0^h, \tau_1^h, \ldots$ we have a double-sided infinite Poisson process of rate λ_h. Also, for each $i < 0$, we define an independent copy of a random adversary tree $\mathcal{T}_i$ with the same distribution as $\mathcal{T}_0$. And we extend the definition of $\mathcal{T}_h(t)$ and $D_h(t)$ to $t < 0$: the last honest block

mined at $\tau^h_{-1} < 0$ and all honest blocks mined within $(\tau^h_{-1} - \Delta, \tau^h_{-1})$ appear in $\mathcal{T}_h(t)$ at their respective mining times to form the level -1, and the process repeats for level less than -1; let $D_h(t)$ be the level of the last honest arrival before t in $\mathcal{T}_h(t)$, i.e., $D_h(t) = \ell$ if $\tau^h_i \le t < \tau^h_{i+1}$ and the i-th honest block appears at level ℓ of $\mathcal{T}_h(t)$.

These extensions allow us to extend the definition of E_{ij} to all $i, j, -\infty < i < j < \infty$, and define E_j and $\hat{E}_j$ to be:

$$E_j = \bigcap_{i<j} E_{ij}$$

and

$$\hat{E}_j = E_j \cap U_j.$$

Note that $\hat{E}_j \subset \hat{F}_j$, so to prove that $\hat{F}_j$ has a probability bounded away from 0 for all j, all we need is to prove that $\hat{E}_j$ has a non-zero probability.

Recall that we have defined the event $\hat{B}_{ik}$ in §A as:

$$\hat{B}_{ik} = \text{event that } D_i(\textstyle\sum_{m=i}^{k-1} R_m + \Delta + \tau^h_i) \ge D_h(\tau^h_{k-1}) - D_h(\tau^h_i + \Delta). \tag{50}$$

Following the idea in Lemma A.1, we have

$$E_j \cap U_j = \bigcap_{i<j} E_{ij} \cap U_j = \left(\bigcap_{i<j<k} \hat{B}^c_{ik} \right) \cap U_j.$$

Hence $E_j \cap U_j$ has a time-invariant dependence on $\{\mathcal{Z}_i\}$, which means that $p = P(\hat{E}_j)$ does not depend on j. Then we can just focus on $P(\hat{E}_0)$. This is the last step to prove.

$$\begin{aligned} P(\hat{E}_0) &= P(E_0|U_0)P(U_0) \\ &= P(E_0|U_0)P(R_0 > \Delta)P(R_{-1} > \Delta) \\ &= e^{-2\lambda_h \Delta} P(E_0|U_0). \end{aligned}$$

It remains to show that $P(E_0|U_0) > 0$. We have

$$\begin{aligned} E_0 = \text{event that } & D_i(\sum_{m=i}^{k-1} R_m + \Delta + \tau^h_i) < D_h(\tau^h_{k-1}) - D_h(\tau^h_i + \Delta) \\ & \text{for all } k > 0 \text{ and } i < 0, \end{aligned}$$

then

$$E_0^c = \bigcup_{k>0, i<0} \hat{B}_{ik}. \tag{51}$$

Let us fix a particular $n > 2\lambda_h \Delta > 0$, and define:

$$\begin{aligned} G_n = \text{event that} & D_m(3n/\lambda_h + \tau^h_m) = 0 \\ & \text{for } m = -n, -n+1, \ldots, -1, 0, +1, \ldots, n-1, n \end{aligned}$$

Then

$$\begin{aligned} P(E_0|U_0) &\ge P(E_0|U_0, G_n)P(G_n|U_0) \\ &= \left(1 - P(\cup_{k>0, i<0} \hat{B}_{ik}|U_0, G_n)\right) P(G_n|U_0) \\ &\ge \left(1 - \sum_{k>0, i<0} P(\hat{B}_{ik}|U_0, G_n)\right) P(G_n|U_0) \\ &\ge (1 - a_n - b_n) P(G_n|U_0) \end{aligned} \tag{52}$$

where

$$a_n := \sum_{(i,k): -n \le i < 0 < k \le n} P(\hat{B}_{ik}|U_0, G_n) \tag{53}$$

$$b_n := \sum_{(i,k): i < -n \text{ or } k > n} P(\hat{B}_{ik}|U_0, G_n). \tag{54}$$

Using (20), we can bound $P(\hat{B}_{ik}|U_0, G_n)$. Consider two cases:

Case 1: $-n \le i < 0 < k \le n$:

$$\begin{aligned} P(\hat{B}_{ik}|U_0, G_n) &= P(\hat{B}_{ik}|U_0, G_n, \sum_{m=i}^{k-1} R_m + \Delta \le 3n/\lambda_h) \\ &\quad + P(\sum_{m=i}^{k-1} R_m + \Delta > 3n/\lambda_h|U_0, G_n) \\ &\le P(\sum_{m=i}^{k-1} R_m + \Delta > 3n/\lambda_h|U_0, G_n) \\ &\le P(\sum_{m=i}^{k-1} R_m > 5n/(2\lambda_h)|U_0) \\ &\le P(\sum_{m=i}^{k-1} R_m > 5n/(2\lambda_h))/P(U_0) \\ &\le A_1 e^{-\alpha_1 n} \end{aligned}$$

for some positive constants A_1, α_1 independent of n, k, i. The last inequality follows from the fact that R_i's are iid exponential random variables of mean $1/\lambda_h$. Summing these terms, we have:

$$\begin{aligned} a_n &= \sum_{(i,k): -n \le i < 0 < k \le n} P(B_{ik}|U_0, G_n) \\ &\le \sum_{(i,k): -n \le i < 0 < k \le n} A_1 e^{-\alpha_1 n} := \bar{a}_n, \end{aligned}$$

which is bounded and moreover $\bar{a}_n \to 0$ as $n \to \infty$.

Case 2: $k > n$ or $i < -n$:

For $0 < \varepsilon < 1$, let us define event W^ε_{ik} to be:

$$W^\varepsilon_{ik} = \text{event that } D_h(\tau^h_{k-1}) - D_h(\tau^h_i + \Delta) \ge (1-\varepsilon)\tfrac{r_h}{\lambda_h}(k - i - 1). \tag{55}$$

Then we have

$$P(\hat{B}_{ik}|U_0, G_n) \le P(\hat{B}_{ik}|U_0, G_n, W^\varepsilon_{ik}) + P(W^{\varepsilon\,c}_{ik}|U_0, G_n).$$

We first bound $P(W^{\varepsilon\,c}_{ik}|U_0, G_n)$:

$$\begin{aligned} P(W^{\varepsilon\,c}_{ik}|U_0, G_n) &\le P(W^{\varepsilon\,c}_{ik}|\tau^h_{k-1} - \tau^h_i - \Delta > \frac{k-i-1}{(1+\varepsilon)\lambda_h}) \\ &\quad + P(\tau^h_{k-1} - \tau^h_i - \Delta \le \frac{k-i-1}{(1+\varepsilon)\lambda_h}) \\ &\le P(W^{\varepsilon\,c}_{ik}|\tau^h_{k-1} - \tau^h_i - \Delta > \frac{k-i-1}{(1+\varepsilon)\lambda_h}) \\ &\quad + e^{-\Omega(\varepsilon^2(k-i-1))} \\ &\le e^{-\Omega(\varepsilon^4(k-i-1))} + e^{-\Omega(\varepsilon^2(k-i-1))} \\ &\le A_2 e^{-\alpha_2(k-i-1)} \end{aligned} \tag{56}$$

for some positive constants A_2, α_2 independent of n, k, i, where the second inequality follows from the Erlang tail bound and the third inequality follows from Proposition C.2 .

Meanwhile, we have

$$\begin{aligned}
& P(\hat{B}_{ik}|U_0, G_n, W_{ik}^{\varepsilon}) \\
\leq\ & P(D_i(\sum_{m=i}^{k-1} R_m + \Delta + \tau_i^h) \geq (1-\varepsilon)\frac{r_h}{\lambda_h}(k-i-1)|U_0, G_n, W_{ik}^{\varepsilon}) \\
\leq\ & P(D_i(\sum_{m=i}^{k-1} R_m + \Delta + \tau_i^h) \geq (1-\varepsilon)\frac{r_h}{\lambda_h}(k-i-1) \\
& \qquad |U_0, G_n, W_{ik}^{\varepsilon}, \sum_{m=i}^{k-1} R_m + \Delta \leq (k-i-1)\frac{r_h+\lambda_a e}{2\lambda_a e}\frac{1}{\lambda_h}) \\
& + P(\sum_{m=i}^{k-1} R_m + \Delta > (k-i-1)\frac{r_h+\lambda_a e}{2\lambda_a e}\frac{1}{\lambda_h}|U_0, G_n, W_{ik}^{\varepsilon}) \\
\leq\ & P(\sum_{m=i}^{k-1} R_m + \Delta > (k-i-1)\frac{r_h+\lambda_a e}{2\lambda_a e}\frac{1}{\lambda_h}|U_0, G_n, W_{ik}^{\varepsilon}) \\
& + \left(\frac{r_h+\lambda_a e}{2(1-\varepsilon)r_h}\right)^{(1-\varepsilon)\frac{r_h}{\lambda_h}(k-i-1)}
\end{aligned}$$

where the first term in the last inequality follows from (20), and the second term can also be bounded:

$$\begin{aligned}
& P(\sum_{m=i}^{k-1} R_m + \Delta > (k-i-1)\frac{r_h+\lambda_a e}{2\lambda_a e}\frac{1}{\lambda_h}|U_0, G_n, W_{ik}^{\varepsilon}) \\
=\ & P(\sum_{m=i}^{k-1} R_m + \Delta > (k-i-1)\frac{r_h+\lambda_a e}{2\lambda_a e}\frac{1}{\lambda_h}|U_0, W_{ik}^{\varepsilon}) \\
\leq\ & P(\sum_{m=i}^{k-1} R_m + \Delta > (k-i-1)\frac{r_h+\lambda_a e}{2\lambda_a e}\frac{1}{\lambda_h})/P(U_0, W_{ik}^{\varepsilon}) \\
\leq\ & A_3 e^{-\alpha_3(k-i-1)}
\end{aligned}$$

for some positive constants A_3, α_3 independent of n, k, i. The last inequality follows from the fact that $(r_h + \lambda_a e)/(2\lambda_a e) > 1$ and the R_i's have mean $1/\lambda_h$, while $P(U_0, W_{ik}^{\varepsilon})$ is a event with high probability as we showed in (56).

Then we have

$$\begin{aligned}
P(\hat{B}_{ik}|U_0, G_n) \leq\ & A_2 e^{-\alpha_2(k-i-1)} + \left(\frac{r_h+\lambda_a e}{2(1-\varepsilon)r_h}\right)^{(1-\varepsilon)\frac{r_h}{\lambda_h}(k-i-1)} \\
& + A_3 e^{-\alpha_3(k-i-1)}. \qquad (57)
\end{aligned}$$

Summing these terms, we have:

$$\begin{aligned}
b_n =\ & \sum_{(i,k):i<-n \text{ or } k>n} P(\hat{B}_{ik}|U_0, G_n) \\
\leq\ & \sum_{(i,k):i<-n \text{ or } k>n} [A_2 e^{-\alpha_2(k-i-1)} \\
& + \left(\frac{r_h+\lambda_a e}{2(1-\varepsilon)r_h}\right)^{(1-\varepsilon)\frac{r_h}{\lambda_h}(k-i-1)} + A_3 e^{-\alpha_3(k-i-1)}] \\
:=\ & \bar{b}_n
\end{aligned}$$

which is bounded and moreover $\bar{b}_n \to 0$ as $n \to \infty$ when we set ε to be small enough such that $\frac{r_h+\lambda_a e}{2(1-\varepsilon)r_h} < 1$.

Substituting these bounds in (52) we finally get:

$$P(E_0|U_0) > [1 - (\bar{a}_n + \bar{b}_n)]P(G_n|U_0) \qquad (58)$$

By setting n sufficiently large such that $\bar{a}_n$ and $\bar{b}_n$ are sufficiently small, we conclude that $P(\hat{E}_0) > 0$.

E PROOF OF PERSISTENCE AND LIVENESS

In this section, we will prove Lemma 4.4. Our goal is to generate a transaction ledger that satisfies persistence and liveness as defined in section 4.1. Together, persistence and liveness guarantees *robust transaction ledger* [GKL15]; honest transactions will be adopted to the ledger and be immutable.

Proof. We first prove persistence by contradiction. For a chain C_t with the last block mined at time t, let $C_t^{\lceil\sigma}$ be the chain resulting from pruning a chain C_t up to σ, by removing the last blocks at the end of the chain that were mined after time $t-\sigma$. Note that $C^{\lceil\sigma}$ is a prefix of C, which we denote by $C^{\lceil\sigma} \preceq C$.

Let C_t denote the longest chain adopted by an honest node with the last block mined at time t. Suppose there exists a longest chain C_t' adopted by some honest node with the last block mined at time $t' > t$ and $C_t^{\lceil\sigma} \not\preceq C_{t'}$. There are a number of honest blocks mined in the time interval $[t-\sigma, t]$, each of which can be in $C_t, C_{t'}$, or neither. We partition the set of honest blocks generated in that interval with three sets: $\{\mathcal{H}_t \triangleq \{H_j \in C_t : \tau_j \in [t-\sigma, t]\}, \mathcal{H}_{t'} \triangleq \{H_j \in C_{t'} : \tau_j \in [t-\sigma, t]\}$, and $\mathcal{H}_{\text{rest}} \triangleq \{H_j \notin C_t \cup C_{t'} : \tau_j \in [t-\sigma, t]\}$, depending on which chain they belong to.

Then we claim that $C_t^{\lceil\sigma} \not\preceq C_{t'}$ implies that $\hat{F}_j^c$ holds for all j such that $\tau_j \in [t-\sigma, t]$. This in turn implies that $P(C_t^{\lceil\sigma} \not\preceq C_{t'}) \leq P(\cap_{j:\tau_j \in [t-\sigma,t]} \hat{F}_j^c)$. However, we know that the probability of this happening is as low as q_σ. This follows from the following facts. (i) the honest blocks in C_t does not make it to the longest chain at time t': $H_j \notin C_{t'}$ for all $H_j \in \mathcal{H}_t$, which follows from $C_t^{\lceil\sigma} \not\preceq C_{t'}$. (ii) the honest blocks in $C_{t'}$ does not make it to the longest chain C_t at time t: $H_j \notin C_t$ for all $H_j \in \mathcal{H}_{t'}$, which also follows from $C_t^{\lceil\sigma} \not\preceq C_{t'}$. (iii) the rest of the honest blocks did not make it to either of the above: $H_j \notin C_t \cup C_{t'}$ for all $H_j \in \mathcal{H}_{\text{rest}}$.

We next prove liveness. Assume a transaction tx is received by all honest nodes at time t, then we know that with probability at least $1 - q_\sigma$, there exists one honest block b_j mined at time τ_j^h with $\tau_j^h \in [t, t+\sigma]$ and event $\hat{F}_j$ occurs, i.e., the block b_j and its ancestor blocks will be contained in any future longest chain. Therefore, tx must be contained in block b_j or one ancestor block of b_j since tx is seen by all honest nodes at time $t < \tau_j$. In either way, tx is stabilized forever. Thus, liveness holds. □