

Blockchain with Varying Number of Players

T-H. Hubert Chan* Naomi Ephraim[†] Antonio Marcedone[†] Andrew Morgan[†]
Rafael Pass[‡] Elaine Shi[†]

Abstract

Nakamoto’s famous blockchain protocol enables achieving consensus in a so-called permissionless setting—anyone can join (or leave) the protocol execution, and the protocol instructions do not depend on the identities of the players. His ingenious protocol prevents “sybil attacks” (where an adversary spawns any number of new players) by relying on computational puzzles (a.k.a. “moderately hard functions”) introduced by Dwork and Naor (Crypto’92). Recent work by Garay et al (EuroCrypt’15) and Pass et al. (EuroCrypt’17) demonstrate that this protocol provably achieves consistency and liveness assuming a) honest players control a majority of the computational power in the network, b) the puzzle-difficulty is appropriately set as a function of the maximum network message delay and the total computational power of the network, and c) the computational puzzle is modeled as a random oracle.

These works, however, leave open the question of how to set the puzzle difficulty in a setting where the computational power in the network is changing. Nakamoto’s protocol indeed also includes a description of a difficulty update procedure. A recent work by Garay et al. (Crypto’17) indeed shows a variant of this difficulty adjustment procedure can be used to get a sound protocol as long as the computational power does not change too fast — however, under two restrictions: 1) their analysis assumes that the attacker cannot delay network messages, and 2) the changes in computational power in the network changes are statically set (i.e., cannot be adaptively selected by the adversary). In this work, we show the same result but without these two restrictions, demonstrating the soundness of a (slightly different) difficulty update procedure, assuming only that the computational power in the network does not change too fast (as a function of the maximum network message delays); as an additional contribution, our analysis yields a tight bound on the “chain quality” of the protocol.

*The University of Hong Kong. hubert@cs.hku.hk

[†]Cornell University. nhe22@cornell.edu, {a.marcedone, asmmathematics, runting}@gmail.com

[‡]Cornell Tech. rafael@cs.cornell.edu

1 Introduction

Distributed systems have been historically analyzed in a *closed* setting—a.k.a. the *permissioned setting*—in which the number of participants in the system, as well as their identities, are common knowledge. In 2008, Nakamoto [Nak08] proposed his celebrated “blockchain protocol” which attempts to achieve consensus in a *permissionless* setting: anyone can join (or leave) the protocol execution (without getting permission from a centralized or distributed authority), and the protocol instructions do not depend on the identities of the players. The core blockchain protocol (a.k.a. “Nakamoto consensus”, or the “Bare-bones blockchain protocol”), roughly speaking, is a method for maintaining a *public, immutable* and *ordered* ledger of records (for instance, in the Bitcoin application, these records are simply transactions); that is, records can be added to the *end* of the ledger at any time (but only to the end of it); additionally, we are guaranteed that records previously added cannot be removed or reordered and that all honest users have a *consistent view* of the ledger—we refer to this as *consistency*. Additionally, the protocol should satisfy a *liveness* property: transactions submitted by an honest user get incorporated into the ledger sufficiently fast.

The key challenge with the permissionless setting is that an attacker can trivially mount a so-called “sybil attack”—it simply spawns lots of players (that it controls) and can thus easily ensure that it controls a majority of all the players. Indeed, Barak et al [BCL⁺05] proved that this is a fundamental problem with the permissionless model. Nakamoto blockchain protocol overcomes this issue by relying on “computational puzzles”—a.k.a. *moderately hard functions* or *proofs of work*—put forth by Dwork and Naor [DN92]: roughly speaking, the participants are required to solve the computational puzzle of some well-defined difficulty in order to confirm “blocks” of transactions—this is referred to as *mining*; More precisely, each participant (i.e., miner) maintains its own local “chain” of “blocks” of records/messages, called the *blockchain*, and attempts to extend it with new blocks by trying to solve a computational puzzle which is a function of the current chain and the new block of transactions.

Next, rather than attempting to provide robustness whenever the majority of the participants are honest (since participants can be easily spawned in the permissionless setting), Nakamoto’s goal was to provide robustness of the protocol under the assumption that a *majority of the computing power* is held by honest participants. Indeed, recent works by Garay et al. [GKL15] and Pass et al. [PSS17] formally proved that Nakamoto’s blockchain protocol satisfies the above-mentioned consistency and liveness under different network assumptions, as long as the puzzle difficulty—referred to as the *mining hardness*—is appropriately set as a function of the maximum delay in the network, and the total computing power in the network; additionally, it is assumed that the total computing power in the network remains *unchanged*.

Setting the hardness mining: But how do we ensure that the mining difficulty is appropriately set? As shown in [PS17b], unless there is an upper-bound on the network delay, we cannot hope to get security in the permissionless setting, so the first assumption (of known network delay) in the result of [PSS17] is needed. The knowledge of computing power assumption, however, may not be. Indeed, Nakamoto’s blockchain protocol also provides an estimation and hardness update mechanism based on how fast the chain grows in order to deal with a variable number of participating player (or, more specifically, a varying amount of computing power in the network). This leaves open the question of whether Nakamoto’s protocol, or any other protocol remains secure

in a setting with a varying number of players:

Does Nakamoto’s protocol satisfy consistency and liveness when the total computing power in the network can vary, assuming just an upper-bound on the maximum network delay?

An elegant recent work by Garay et al [GKL17] provides a first step towards addressing this question; they prove security assuming a) the computing power changes are sufficiently “smooth” (i.e., the number of miners are not growing too fast or too slow), b) the changes in computing power are independent of the execution (and in particular, are not adversarially selected), and c) assuming no network delays (i.e., in the fully synchronous setting). Additionally, as we discuss in the related work section, the concrete parameters obtained, and in particular the “chain quality” obtained in their analysis are not optimal.

We first observe that condition a) is necessary for Nakamoto’s blockchain: if changes in computing power are too fast, the mining hardness is not appropriately calibrated and the attack from [PSS17] directly applies. Conditions b) and c), however, are quite severe restrictions from a practical point of view.

Our main result shows how to overcome both these issues, proving security of Nakamoto’s protocol assuming only a) the initial mining hardness is not too small relative to the maximum network delay and the initial number of players; and b) the change in computing power change is not too fast (as a function of the maximum network delay). Our parameters match the parameters of earlier blockchain analyses in the fixed mining power setting [GKL15, PSS17]:

Theorem 1.1 (Informal). *Assume a proof-of-work random oracle. There exists a permissionless blockchain protocol that retains security as long as 1) at any time, the adversary controls only a minority coalition; 2) the number of players in the system does not increase too abruptly; and 3) the protocol is initialized with a mining hardness that is not too easy with respect to the maximum network delay and the initial mining power.*

In summary, our results further our understanding of the feasibility of consensus in a truly permissionless environment with a varying number of players.

1.1 Related Work

Analysis of Nakamoto’s blockchain under a fixed number of players. Several earlier works analyzed the security of Nakamoto’s blockchain protocol under the assumption that the adversary controls only minority of the total mining power. An elegant work by Sompolinsky and Zohar shows that Nakamoto’s blockchain retains consistency against certain restricted attacks [SZ15]. Garay, Kiayas and Leonardos [GKL15] were the first to show that Nakamoto’s blockchain retains consistency against an arbitrarily malicious adversary that controls, as long as the adversary must deliver messages immediately. Pass et al. were the first to prove the security of Nakamoto’s blockchain when the adversary can reorder and delay messages arbitrarily, as long as any message sent by honest nodes is delivered with a maximum of Δ rounds. Subsequently, Garay et al. also extended their earlier analysis the same bounded delay model [GKL15].

To the best of our knowledge, all these works assume that the number of nodes n is a-priori known and fixed over the entire duration of the protocol.

Most closely related work. In a recent elegant work, Garay, Kiayias, and Leonardos [GKL17] were the first to analyze the security of the Nakamoto blockchain under a varying number of players. They proved that a variant of Nakamoto’s protocol with difficulty adjustment satisfies consistency, chain quality, and chain growth. Their analysis employed several elegant ideas: for example, they employ a Martingale analysis to bound the stochastic process induced by mining. Further, their idea of expressing chain growth (and other random variables) in terms of work rather than the absolute number of blocks also inspired some techniques adopted in our work.

The analysis by Garay et al. [GKL17] is a first step towards understanding the feasibility of reaching consensus in a truly permissionless setting where the number of players can vary over time. They give a *partial* answer to this question, leaving several important questions open: 1) Garay et al. [GKL17] works only when the adversary is forced to deliver messages immediately, and they phrase it as an open question how to prove security when the adversary is allowed to delay messages; 2) Garay et al. [GKL17]’s analysis is applicable only when the adversary cannot adaptively choose the number of nodes in each round — in other words, the adversary must commit to how many nodes per round upfront prior to the protocol execution, thus leaving it open how to reason about security when the adversary can adaptively choose the number of players; and 3) Garay et al. [GKL17]’s chain quality analysis is not tight: for example, they prove roughly $\frac{1}{4}$ -chain quality when the adversary controls up to $\frac{1}{3}$ of the mining power. Our chain quality proof is tight and matches the chain quality of earlier fixed- n analyses [PSS17, GKL15]. For example, for the aforementioned case of $\frac{1}{3}$ corruption, we prove $\frac{1}{2}$ -chain quality.

2 Technical Roadmap

2.1 Nakamoto’s Varying Difficulty Blockchain

At this moment, we informally describe a simplified version of Nakamoto’s blockchain and explain their difficulty adjustment scheme [Nak08]. Later in Section 4, we will formally specify a variant of a Nakamoto-like blockchain protocol (with difficulty adjustment), and our proofs apply to this formally specified Nakamoto variant.

Nakamoto’s blockchain relies on a proof-of-work random oracle henceforth denoted H . Without loss of generality, we assume that any node can query H at most once. In Nakamoto’s blockchain, each node maintains an internal blockchain *chain* at any point of time. Each $chain[i]$ is referred to as a (mined) block and is of the format

$$chain[i] := (h_{-1}, \eta, txs, p, t, h)$$

containing the hash of the previous block denoted h_{-1} , a nonce η , a record txs , a difficulty parameter p , a timestamp t , and a hash h ¹. Let $chain := \text{extract}(chain)$ be the sequence of records contained in the sequence of blocks *chain*. \mathbf{chain} is the version that honest nodes output to the environment.

Blockchain validity. Let $chain[i] = (h_{-1}, \eta, txs, p, t, h)$ and let $chain[:i-1] = (h'_{-1}, \eta', txs', p', t', h')$. The block $chain[i]$ is only *valid* with respect to the predecessor chain $chain[:i-1]$ if the following conditions hold:

¹In reality (as well as in the description in the introduction), h is not included in the block (as it can be easily determined from the remaining elements); we include it to ensure that we can verify validity of a block using only $H.ver$.

1. $h_{-1} = h'$;
2. $h = \mathbf{H}(h_{-1}, \eta, \text{txs}, p, t)$, and $h < D_p$; and
3. the timestamp t and the difficulty parameter p respect certain constraints w.r.t. to the prefix $\text{chain}[i - 1]$, and we shall explain these constraints later.

Note that the second condition requires that η be a “difficult enough” solution for a proof-of-work puzzle solution for the puzzle payload $(h_{-1}, \text{txs}, p, t)$, where “difficult enough” is specified by the condition that the random oracle query $\mathbf{H}(h_{-1}, \eta, \text{txs}, p, t)$ outputs a “hash” value $h < D_p$, and D_p is chosen such that the probability that any fresh output from the random oracle is less than D_p with probability only p .

Finally, a blockchain is valid if each block refers to the previous block’s hash and moreover if each block is valid with respect to its predecessor chain.

Chain preference: most-work chain. In the Nakamoto protocol, all nodes are initialized with a canonical genesis block. Then, in every round, every node receives all valid blockchains from the network — and if any of the blockchains received has more work than the node’s local *chain*, the node changes its local *chain* to the *most-work chain*. Suppose that in a valid blockchain *chain*, each block $\text{chain}[i]$ is denoted as $\text{chain}[i] = (-, -, -, p_i, -, -)$, then the “work-length” of this blockchain, henceforth denoted $\|\text{chain}[i]\|$, is defined as

$$\|\text{chain}[i]\| = \sum_{i=1}^{|\text{chain}|} \frac{1}{p_i}$$

Throughout the paper, we use $\|\text{chain}[i]\|$ to denote the work length, i.e., the total work contained in the blockchain, and we use $|\text{chain}|$ to denote the length of *chain* in terms of the number of blocks.

Mining. In every round, let *chain* be a node’s local chain, and let t be the current time (i.e., round counter). Now the node selects a random puzzle solution η , a set of transactions it wishes to confirm denoted txs , and queries the proof-of-work oracle with the puzzle payload $(\text{chain}[-1].h, \eta, \text{txs}, p, t)$ where t is the current round counter and p is a difficulty parameter to be specified later. If the outcome is less than D_p , the node extends the blockchain with the next block, and it will then announce the new blockchain to the rest of the network.

Difficulty adjustment. Nakamoto’s original blockchain [Nak08] hardcodes an initial difficulty parameter henceforth denoted p_0 . Every L_{epoch} number of blocks, the difficulty parameter is recalculated based on *chain* whose length is assumed to be a multiple of L_{epoch} .

Roughly speaking, the difficulty calculation function *chain* inspects the recent L_{epoch} blocks in *chain*, and estimates how long it took for the chain to grow by L_{epoch} number of blocks in the most recent past. The goal of Nakamoto’s blockchain is to maintain an expected block interval of roughly 10 minutes, even when the number of players can change over time. Thus, if the past L_{epoch} blocks took more than $10L_{\text{epoch}}$ minutes, then the difficulty should reduce (i.e., p becomes larger); otherwise, the difficulty should increase (i.e., p becomes smaller).

Although this idea appears simple, there are two important things to note that are critical to security.

- **Obtaining somewhat accurate timestamps.** Purported timestamps in blocks are not guaranteed to be accurate. While honest nodes always report time truthfully, corrupt nodes can put in arbitrary timestamps. Nakamoto’s blockchain relies on the following ideas to obtain somewhat accurate timestamps.

1. Honest nodes reject blockchains that carry timestamps 2 hours or more in the future;
2. In a valid blockchain, a block’s timestamp cannot be smaller than the median of the past 11 blocks.

The idea here is that as long as the adversary controls only minority of the mining power, every honest node’s blockchain has positive chain quality [PSS17], and thus every now and then there will be an honest block. Since honest blocks have truthful timestamps, the above rules in effect allow the adversary to skew the timestamps by a maximum of 2 hours or so. Our provably secure variant later (Section 4) is inspired by these ideas but we use a variant of these rules.

- **Bounded difficulty change.** Another important defense mechanism is that the difficulty change must be bounded. In the Nakamoto blockchain, the maximum amount of change in the difficulty parameter is bounded from both sides by a factor of 4. As we will explain later, this bounded change condition is important to maintaining the protocol’s security. Without it, there will be an attack that breaks consistency with inverse polynomial probability.

2.2 Challenges of Analyzing a Varying Difficulty Blockchain

With the exception of the recent work by Garay et al. [GKL17], all previous analyses of the blockchain protocol [PSS17, GKL15] assume a fixed difficulty parameter p and a fixed number of players n in every round. In a permissionless setting, anyone can join and leave the protocol at any time. Thus understanding a blockchain with difficulty adjustment is an important next step for us to understand the feasibilities and infeasibilities of reaching consensus in a permissionless setting, where the number of players may not be known a-priori and can change over time.

As it turns out, extending previous analyses that assume fixed n and p [PSS17, GKL15] to the case of varying n and p is highly non-trivial! For example, Garay et al. [GKL17] also attempted to do this but their analysis and results has several limitations as we explained in Section 1.1.

Core random variables in previous blockchain analyses. To understand the challenges, let us turn to the core of earlier analyses [PSS17, GKL15]. Both Garay et al. and Pass et al.’s analysis eventually boil down to bounding two important random variables associated with the stochastic process induced by the mining protocol:

1. *Total number of adversarial blocks within a time window.* We would like to prove that since an adversary does not have too much mining power, it cannot inject too many blocks into a blockchain during a given time window.
2. *Number of convergence opportunities within a time window.* A convergence opportunity is a good pattern where in some round, a single honest node mines a block and in the Δ adjacent rounds before and after, no honest node mines a block. Convergence opportunities are “good” in two senses: *i*) obviously by definition every convergence opportunity allows honest chains to grow; and *ii*) the adjacent Δ rounds of silence on both sides also help with consistency as shown in earlier works [GKL15, PSS17].

If one can prove sharp concentration bounds for the above two core random variables, then proving chain quality and consistency is not too difficult [PSS17, GKL15] (we focus on chain quality and consistency here since chain growth is easier to prove in comparison): essentially, both chain quality and consistency proofs boil down to showing that in every $\Theta(\kappa)$ blocks of time, there must be more convergence opportunities than adversarial blocks except with $\text{negl}(\kappa)$ probability.

Challenge of varying difficulty: adaptive choice of p and n creates dependence. Take the easier task of upper bounding adversarial blocks as an example (bounding convergence opportunities is more complicated). In the case of fixed p , upper bounding adversarial blocks is relatively easy: one can imagine that honest or corrupt nodes make queries to a proof-of-work oracle. For each query, the oracle will flip a random coin of a fixed probability p , and if the outcome is heads, a block is mined. Thus given any fixed time window of length t , the oracle can receive at most ρnt queries from corrupt nodes where ρ denotes the maximum corrupt fraction. Therefore, upper bounding adversarial total blocks during this window is simply a matter of applying the standard Chernoff bound.

Unfortunately this approach fails in the case of varying p and varying n . More specifically, the adversary has influence over the choice of p and n for every round, and such choices can be made adaptively based on having observed the entire previous of the execution. The ability for an adversary to adaptively choose n and p over time creates a dependence in the underlying stochastic process, making standard Chernoff bound approach fail. The earlier work by Garay et al. [GKL17] adopted a Martingale analysis but even their approach only *partially* deals with such dependence — they allow the adversary to adaptively choose p over time but not n . In other words, their analysis is applicable in a model where the adversary must commit to the number of nodes in each round upfront prior to the protocol execution.

2.3 Our Approach

We now give an overview of our approach. Before proceeding, we note that our approach departs from earlier analysis in three respects:

1. We view the block-mining stochastic process “coin by coin” rather than “round by round” where the latter was taken by all previous analyses [PSS17, GKL15, GKL17] — this is a core reason why our analysis can deal with adaptive choice of n over time while the earlier approach by Garay et al. [GKL17] could not. In other words, our ideal-world analysis is oblivious to how many rounds have elapsed, and only cares about how many coin flips have taken place.
2. Like Garay et al. [GKL17], in an ideal-world analysis, we bound random variables in terms of “total work” rather than the absolute number of blocks, since now coins can have varying difficulty parameters, and the work received upon each successful coin flip is weighted by the difficulty parameter.

We then show that since the real-world protocol only changes the difficulty every $\Theta(\kappa)$ blocks, statements in the ideal world that bounds total work can be converted back to bounds on absolute number of blocks in the real-world.

3. Finally, we employ the method of moment generating functions [MR95] for proving measure concentration bounds. We carefully deal with the dependence in the “world of moment generating functions”, and effectively show that as long as nodes adopt difficulty parameters p that

have bounded difference in any medium sized duration, then the adversary is limited in its ability to blow up the moment generating functions of core random variables, despite its ability to adaptively select p and n .

We now present an intuitive roadmap for our analysis and this should help the reader navigate our subsequent formal proofs.

2.3.1 A Core Randomized Experiment

First, we capture the core stochastic process of a varying difficulty blockchain protocol in a simple randomized experiment (Section 6). This randomized experiment captures what happens in any medium sized duration in the blockchain protocol (where “medium sized” is a technical condition to be specified later in Definition 2.1).

In this simple randomized experiment, an arbitrary, possibly computationally unbounded adversary interacts with an oracle. The adversary adaptively specifies a sequence of difficulty parameters p_1, p_2, \dots, p_T , and each time the oracle flips a coin that comes up heads with probability p_i specified by the adversary where $i \in [T]$. If the i -th coin flip comes up heads, the adversary receives work $1/p_i$. We stress that the adversary’s choice of p_i can depend on the outcomes of the first $i - 1$ coin flips.

We specify an important *bounded change* condition on this randomized experiment: all probabilities submitted by the adversary must be at most a multiplicative factor γ apart from each other. The implication of this constraint is that in the blockchain protocol, we require that over any medium sized duration, all nodes must be mining at difficulties not too far apart from each other — as we explain later, this bounded change condition is necessary for a Nakamoto-like varying difficulty blockchain to be secure.

Bounding adversarial total work. It is not hard to see that no matter how the adversary adaptively chooses the p values, in expectation the adversary receives T amount of total work. We wish to prove that no matter how the adversary manipulates the choice of p , it cannot increase its work received by too much. Let $\mathbf{W}[1 : i]$ be a random variable denoting the total work received by the first i coin flips. We employ the method of moment generating functions, and let $\mathbf{E}[t \exp(\mathbf{W}[1 : i])]$ denote the moment generating function of $\mathbf{W}[1 : i]$. To cause deviation from the mean, the adversary wants to blow up the moment generating function $\mathbf{E}[t \exp(\mathbf{W}[1 : T])]$. The core of our proof boils down to showing that to maximize $\mathbf{E}[t \exp(\mathbf{W}[1 : T])]$, the adversary’s best strategy is to always stick to the smallest p value. Informally, no matter what the choice of p is, the expected contribution to the total work of every coin flip is 1. However, the smaller each p is, the higher the variance which helps to create deviation. To prove this formally, though, requires a somewhat involved induction proof that considers the moment generating function of the random variable $Z_{i+1} := \mathbf{W}[1 : i + 1] - \mathbf{W}[1 : i]$ (i.e., the contribution of the $(i + 1)$ -st coin to total work). Effectively, we need to show that the moment generating function of the Z_{i+1} conditioning on any prefix of the execution is maximized when the adversary just chooses the smallest p possible for the $(i + 1)$ -st coin. Finally, had the adversary stuck to the same p value throughout, then we can reduce the task of proving measure concentration to a standard Chernoff bound.

Bounding convergence opportunities. Next, we wish to lower bound the *work* received by “convergence opportunities” in the aforementioned randomized experiment. A coin flip i is con-

sidered a convergence opportunity if it is the only successful coin in the adjacent neighborhood of $2m + 1$ coin flips, i.e., coins in the range $[i - m, i + m]$. By definition, the outcome of each coin flip will now affect whether coins in its immediate neighborhood are convergence opportunities. Earlier in Section 2.2, we briefly explained why convergence opportunity is a good pattern that helps with chain quality and consistency.

Bounding the work received by convergence opportunities turns out to be much trickier than the task of bounding adversarial total work. When bounding adversarial total work, there was a single source of dependence: the adversary’s ability to choose p . Now we have two different sources of dependence: the adversary’s ability to choose p , as well as each coin flip’s influence over its immediate neighborhood.

To cope with this issue, we divide the coin flips into three *portions* (whose technical definition will be presented in Section 6). Each “portion” is crafted such that within the portion, we only have to deal with dependence resulting from adversarial choice of p — every other dependence within a portion acts in our favor (i.e., the correlation would be negative had the adversary stuck to the same p). Now we bound the moment generating function of each portion using a similar method we adopted for bounding adversarial work — but now with a more involved argument, because it is in the adversary’s interest to use a small p value to cause variance in the work received, but on the other hand, it is in the adversary’s interest to use a large p to “deny” more coins from becoming convergence opportunities.

Finally, although arbitrary dependence is possible among the 3 portions, since the number of portions is small, it is not difficult to sharply upper bound the moment generating function of all 3 portions even when the dependence among the portions can act in a fashion that maximally hurts measure concentration (formally this relies on the convexity of the moment generating function).

2.3.2 Ideal-World Blockchain with Varying Difficulty

Having proved sharp measure concentration bounds on core random variables, we now turn our attention to an ideal-world blockchain protocol denoted Π_{ideal} (see Section 7 for a formal description), and we would like to understand quantities such as work growth, work quality, and consistency in this ideal-world protocol. We first briefly describe the ideal-world protocol and then explain the quantities we care about.

In our ideal-world blockchain protocol, all nodes mine blocks by interacting with an ideal functionality $\mathcal{F}_{\text{tree}}$ that keeps track of all valid blockchains seen so far. To conduct a mining attempt, each node specifies to $\mathcal{F}_{\text{tree}}$ the chain to extend from, a difficulty parameter p , and additional block payload which we do not care about in the ideal-world protocol. $\mathcal{F}_{\text{tree}}$ now flips a coin of corresponding probability to decide if the next block is mined. The difficulty parameters of all (honest and corrupt) queries are determined by the adversary in an arbitrary fashion, as long as an important bounded change condition is satisfied.

Definition 2.1 (Bounded difficulty change in medium sized duration). *Among any window of $\Theta(\kappa/p)$ consecutive $\mathcal{F}_{\text{tree}}$ -coin flips where p is the probability of the initial coin in this window, the difficulty parameters of all $\mathcal{F}_{\text{tree}}$ -queries must be bounded apart by a constant factor γ .*

This bounded change condition turns out to be necessary for a Nakamoto-like varying difficulty blockchain to be secure: otherwise, if adjacent blocks are allowed to have difficulty parameters that are polynomially apart, there is a non-negligible probability that the adversary can mine a

particularly difficult block containing a large amount of work — such a block can overwrite more than $\Theta(\kappa)$ trailing blocks and thus break consistency.

Besides bounded change in difficulty, we require that the *total mining rate be bounded*. More specifically, we require that at any point of time, the “expected” block interval is larger than the maximum network delay Δ by an appropriate constant factor (that is dependent on the corruption threshold) — this technical condition was necessary in earlier analysis [PSS17] of the blockchain with fixed n and p . As Pass et al. argue [PSS17], even in the case of fixed n and p , violating this condition can lead to breaking consistency.

We wish to prove the following statements about the ideal-world protocol Π_{ideal} . Henceforth, let *view* be an execution trace of the ideal protocol Π_{ideal} . If some honest node’s output is *chain* in some round in *view*, we say that *chain* is an honest chain in *view*. Note also that for ease-of-understanding, our explanations below are slightly informal; and we defer the formal definitions of work quality, consistency, and work growth to Section 7.

- *Work growth*. Work growth lower bound says that in almost all views (i.e., all but a negligible fraction), if during some time-frame $[t_0, t_1]$ honest nodes have made $T = \Theta(\kappa)$ queries to $\mathcal{F}_{\text{tree}}$, then any honest chain at t_1 must contain at least $(1 - \epsilon)T$ more work than any honest chain at t_0 where ϵ is an arbitrarily small constant; In other words, honest chains do not grow too slowly in terms of work-length.
- *Work quality*. Work quality says that in almost all views, in any honest chain, any sequence of roughly $\Theta(\kappa)$ blocks must contain a sufficient fraction of honest work. In particular, a block counts towards honest work if it is mined by an honest node.
- *Consistency*. Informally, consistency requires that in almost all views, all honest chains are prefixes of each other except for the trailing $\Theta(\kappa)$ blocks.
- *Work growth upper bound*. Work growth upper bound is similarly defined as work growth lower bound, but for the other direction, i.e., honest chains do not grow too quickly.

Here we take consistency as an example and explain at a very high-level how we prove consistency. We defer the discussions of how to prove the remainder of the properties to Section 7. Consistency can be proven by making the following observation: for each convergence opportunity in *view*, suppose the block mined by this convergence opportunity (henceforth denoted \mathbf{B}^*) is at work-length $[w, w + W]$, then the adversary must have work covering the range $[w, w + W]$, or else the block in every honest chain at work-length $[w, w + W]$ must be the block \mathbf{B}^* . The consistency proof then follows, roughly, by showing that there must be more work earned by convergence opportunities than total adversarial work in every $\Theta(\kappa)$ blocks of time. To bound the work earned by convergence opportunities and total adversarial work in this time window, we would turn to the analysis of our earlier core randomized experiment, and observe that this core randomized experiment captures the nature of the stochastic process of our ideal protocol. The actual proof for consistency contains additional technicalities which we defer to Section 7.

2.3.3 Analyzing a Nakamoto-Like Varying Difficulty Blockchain

Finally, we present a provably secure version of a Nakamoto-like varying difficulty blockchain (referred to as the real-world protocol and denoted Π_{real}). In comparison with the original Nakamoto’s

protocol [Nak08], our provably secure variant uses a slightly different strategy to ensure the relative accuracy of purported timestamps in blockchains. We require that 1) all timestamps in a valid blockchain must strictly increase; and 2) honest nodes reject chains carrying timestamps of the future (*c.f.* earlier in Section 2.1, we described the corresponding rules for Nakamoto).

Our variant has the following advantage which can easily be formalized: assuming chain quality, for every arbitrarily small constant ϵ , there must be an honest block every $\epsilon\kappa$ blocks. Thus our blockchain timestamp constraints effectively stipulate that *any adversarial block's timestamp is sandwiched between two close-by honest blocks* (or genesis/end-of-chain). Thus even adversarial blocks' timestamps cannot deviate too much from the truthful time.

Like Nakamoto, our real-world protocol also sticks to the same difficulty parameter every $L_{\text{epoch}} = \Theta(\kappa)$ blocks — a constraint that any valid blockchain must respect. Further, just like in Nakamoto, difficulty change is performed by measuring how long it took to mine roughly L_{epoch} blocks in the recent past. Importantly, the difficulty change between adjacent epochs are bounded above and below by a constant factor denoted γ . As explained earlier, this bounded change condition is indeed necessary for a Nakamoto-like blockchain to be secure.

Our analysis of Π_{real} proves the following informal statements where technical conditions such as “safe”, “calibrated”, “bounded change in n ” are to be stated precisely in Section 4 — in all these statements we implicitly assume that the adversary controls a minority coalition at any time:

- We prove that the real-world protocol Π_{real} retains consistency and chain quality, as long as 1) the protocol starts in a safe parameter regime, and 2) the number of players n does not increase too suddenly.
- Should the number of players drop suddenly, however, for a while the chain growth can be slow, but consistency and chain quality are nonetheless retained. Similarly, if the initial difficulty parameter is set too pessimistically, consistency and chain quality are retained but initial chain growth can be slow for a while.
- Provided that the protocol starts off with a calibrated difficulty parameter that somewhat accurately reflects the initial number of nodes, and further, provided that the number of players does not increase or decrease abruptly, then not only do we obtain consistency and chain quality, we also guarantee that at any time during the execution, the chain growth rate is a constant factor times the maximum network delay Δ (time per block).

Since the real-world protocol sticks to the same difficulty parameter every epoch, we can easily translate “work growth” and “work quality” to the more standard notions of “chain growth” and “chain quality”. Since the notion of work no longer appears in our final theorem statements, effectively our varying difficulty blockchain exposes (almost) identical abstraction as a blockchain with fixed difficulty [PSS17, GKL15], and applications building on top need not be aware of the difficulty parameter.

Reasoning about the real-world protocol through an inductive argument. Proving the real-world protocol secure involves reasoning that the real-world protocol emulates the ideal world. The bulk of the proof is an inductive argument of the following nature:

- *A safe epoch leads to a next safe epoch.* If in the present epoch, the difficulty parameter is in a safe region that ensures consistency, then the next epoch will also end up with a difficulty parameter that falls within a safe region that ensures consistency. Note that having a more

difficult puzzle is always good for consistency, but can potentially cause the chain to grow very slowly. On the other hand, having a puzzle that is too easy can break consistency.

- *A calibrated epoch leads to a next calibrated epoch.* Additionally, if the present epoch’s difficulty parameter is in a calibrated region such that the actual block interval is off by only a constant factor relative to a targeted block interval Δ_{tgt} , then the next epoch will have a difficulty parameter such that the actual block interval also approximates the targeted block interval Δ_{tgt} up to a constant factor. For consistency to hold, the targeted block interval must be an appropriate constant factor larger than the maximum network delay Δ .

3 Protocol Execution in a Permissionless Model

3.1 Execution in a Permissionless Model

We will adopt a permissionless execution model that was used in several recent works [PSS17, GKL15, PS17a]. Notably, in this model, there is no a-priori common knowledge of the set of players who will participate in a protocol; nodes can join and leave at any time; and the network does not authenticate the sender’s identity.

Interactive Turing Machines and round-based execution. A protocol refers to an algorithm for a set of interactive Turing Machines (also called nodes) to interact with each other. The execution of a protocol Π is directed by an environment $\mathcal{Z}(1^\kappa)$ (where κ is a security parameter), which activates a number of nodes that are either honest or corrupt. Honest nodes faithfully follow the protocol’s prescription, whereas corrupt nodes are controlled by an adversary \mathcal{A} which reads all their inputs/messages and sets their outputs/messages to be sent.

The environment \mathcal{Z} is a catch-all term that encompasses everything that lives outside the “box” defined by the protocol. For example, as mentioned later, part of the environment \mathcal{Z} ’s job is to provide inputs to honest nodes and receive outputs from them. This models the fact that the inputs to the protocol may originate from external applications and the protocol’s outputs can be consumed by external applications where any external application or other protocols running in the system are viewed as part of \mathcal{Z} .

A protocol’s execution proceeds in *rounds* that model atomic time steps. At the beginning of every round, honest nodes receive inputs from an environment \mathcal{Z} ; at the end of every round, honest nodes send outputs to the environment \mathcal{Z} .

Nodes joining, leaving, and adaptive corruption. We consider the following model for spawning and corrupting nodes:

- The environment \mathcal{Z} can spawn new nodes at the beginning of any round; newly spawned nodes are either honest or corrupt.
- At any point, \mathcal{Z} can *corrupt* an honest party j which means that \mathcal{A} gets access to its local state and subsequently, \mathcal{A} controls party j .
- At any point, \mathcal{Z} can *uncorrupt* a corrupted player j , which means that \mathcal{A} no longer controls j . A player that becomes uncorrupt is treated in the same way as a newly spawning player, i.e., the player’s internal state is re-initialized and then the player starts executing the honest protocol

no longer controlled by \mathcal{A} . A node that has become uncorrupted would run whatever initialization procedure the honest protocol stipulates.

Importantly, the number of nodes can vary over the duration of protocol execution, and \mathcal{Z} can decide the number of players in a round adaptively based on the prefix of the execution.

Communication model. \mathcal{A} is responsible for delivering all messages sent by parties (honest or corrupted) to *all* other parties. \mathcal{A} cannot modify the content of messages broadcast by honest players, *but it may delay or reorder the delivery of messages* as long as it respects a Δ -bounded delivery constraint stated below:

If an honest node sends a message in round r , then, in any round $t \geq r + \Delta$, any node that is honest in round t will have received the message at the beginning of the round, including nodes that may have just spawned or become uncorrupt in round t .

The identity of the sender is not known to the recipient.²

Henceforth in our paper, we assume that the blockchain protocol has a-priori knowledge of the maximum network delay Δ . In a recent work, Pass and Shi [PS17b] have shown that without knowledge of Δ , consensus is impossible in a permissionless network where the exact number of players is not known in advance (and the lower bound holds even when assuming a proof-of-work oracle).

3.2 Notations

Notations for randomized execution. A protocol’s execution is randomized, where the randomness comes from honest players as well as the adversary denoted \mathcal{A} that controls all corrupt nodes, and the environment \mathcal{Z} that sends inputs to honest nodes during the protocol execution. We use the notation $\text{view} \leftarrow_{\S} \text{EXEC}^{\Pi}(\kappa, \mathcal{A}, \mathcal{Z})$ to denote a randomly sampled execution trace, and $|\text{view}|$ denotes the number of rounds in the execution trace view . More specifically, view is a random variable denoting the joint view of all parties (i.e., all their inputs, random coins and messages received, including those from the random oracle) in the above execution; note that this joint view fully determines the execution.

Variable conventions. Unless otherwise noted, we assume that all variables are polynomially-bounded function of the security parameter κ . For two variables that by default are functions of κ , we say that $\text{var}_1 < \text{var}_2$ iff for every $\kappa \in \mathbb{N}$, $\text{var}_1(\kappa) < \text{var}_2(\kappa)$. If any variable is not a function of κ , we shall explicitly note that the variable is a *constant*. Variables may also be functions of each other as defined later by relations that $(\mathcal{A}, \mathcal{Z})$ must additionally satisfy for our blockchain protocol to be secure.

Negligible functions. In this paper, whenever we use the term “negligible function”, we always exclusively refer to a *strongly* negligible function. A function $\text{negl}(\kappa)$ is said to be strongly negligible iff there exists some constant $c > 0$, and some $\kappa_0 \in \mathbb{N}$, such that $\text{negl}(\kappa) < \exp(-c\kappa)$ for every $\kappa \geq \kappa_0$.

²We could also consider a seemingly weaker model where messages sent by corrupted parties need not be delivered to all honest players. We can easily convert the weaker model to the stronger model by having honest parties “gossip” all messages they receive.

3.3 Syntax of a Blockchain Protocol

Inputs and outputs. In each time step, the environment \mathcal{Z} inputs payload (e.g., a set of transactions) denoted txs to a node. A node outputs an abstract blockchain denoted chain to the environment \mathcal{Z} , where chain is an ordered sequence of blocks each containing a block payload denoted txs_i :

$$\text{chain} := \{\text{txs}_i\}_{i \in [|\text{chain}|]}$$

Blockchains with a proof-of-work random oracle. A proof-of-work blockchain as represented by Nakamoto’s original proposal [Nak08, PSS17, GKL15] relies on a proof-of-work random oracle. Let $H : \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$ denote a random function. Nodes are allowed to query two functions H and H.ver . $\text{H}(x)$ simply outputs the outcome of the random function $H(x)$, and $\text{H.ver}(x, y)$ outputs 1 iff $H(x) = y$, else it outputs 0. In any round, any node is allowed to make an arbitrary number of queries to H.ver but *at most one query* to H . If the adversary \mathcal{A} controls q corrupt nodes, we allow \mathcal{A} to make q sequential queries to H . We emphasize that the environment \mathcal{Z} cannot access the random oracle.

3.4 Security Definitions for a Blockchain Protocol

Terminology. Whenever we say that

“Except with negligible probability over the choice of view , some event $\text{ev}(\text{view})$ is satisfied”,

we formally mean that for any p.p.t. $(\mathcal{A}, \mathcal{Z})$ compliant w.r.t. to protocol Π , there exists a negligible function $\text{negl}(\cdot)$ such that for all $\kappa \in \mathbb{N}$,

$$\Pr[\text{view} \leftarrow \text{EXEC}^\Pi(\kappa, \mathcal{A}, \mathcal{Z}) : \text{ev}(\text{view}) \text{ is violated}] \leq \text{negl}(\kappa)$$

When the context is clear, we often omit writing the protocol explicitly.

Below we define the security properties satisfied by a blockchain protocol, including chain growth, chain quality, and consistency. Henceforth, whenever we say “*an honest chain in view*”, we mean some honest node’s output chain to the environment \mathcal{Z} in some round in view . We use the notation $\text{chain}_i^t(\text{view})$ to denote node i ’s chain in round t in view — since the context is clear, we often omit writing the view explicitly in the above notation.

Chain growth. Intuitively, chain growth stipulates that honest nodes’ chains grow at a relatively steady pace, neither too fast nor too slow. More formally, we say that a blockchain protocol $\Pi_{\text{blockchain}}$ satisfies (K, g_0, g_1) -chain growth, iff except with negligible probability over the choice of view , the following properties hold:

- *Consistent length.* For any t and $r \geq t + \Delta$, any node i honest in round r , and any node j honest in round r , it holds that $|\text{chain}_j^r| \geq |\text{chain}_i^t|$; here, the notation chain_i^t means the chain of node i in round t .
- *Growth lower bound.* For any honest chain chain^t in round t , and any honest chain chain^r in round r such that $g_0(r - t) \geq K$, it must hold that

$$|\text{ch}^r| - |\text{ch}^t| > g_0 \cdot (r - t)$$

- *Growth upper bound.* For any honest chain chain^t in round t , and any honest chain chain^r in round $r \geq t$, it must hold that

$$|\text{ch}^r| - |\text{ch}^t| < \max(g_1 \cdot (r - t), K)$$

Chain quality. Intuitively, chain quality stipulates that any sufficiently long window of consecutive blocks in any honest chain must have sufficiently many blocks that were “mined” by honest nodes. More formally, we say that a blockchain protocol $\Pi_{\text{blockchain}}$ satisfies (K, μ_0) -chain quality, iff except with negligible probability over the choice of *view*, the following properties hold:

- For any honest chain chain in *view*, any $K' \geq K$ consecutive blocks denoted $\text{chain}[\ell + 1 : \ell + K']$ must have more than μ_0 fraction mined by honest nodes.

In the above, we say that a block $\text{chain}[\ell]$ is mined by honest nodes in *view* iff the environment \mathcal{Z} input the contents contained in $\text{chain}[\ell]$ to some honest node when its output chain to \mathcal{Z} is $\text{chain}[: \ell - 1]$. If a block is not mined by honest nodes, we often say that the block is corrupt.

Consistency. Intuitively, consistency requires that all honest chains in *view* agree with each other except for the trailing few blocks. More formally, we say that a blockchain protocol $\Pi_{\text{blockchain}}$ satisfies K -consistency, iff except with negligible probability over the choice of *view*, the following holds:

- For any node i honest in round t , and any node j honest in round $r \geq t$ where j may be the same as or different from i , it holds that

$$\text{chain}_i^t[: -K] \prec \text{chain}_j^r$$

where \prec denotes “is a prefix of”. By convention, we say that for any list x , $x \prec x$, i.e., any list is considered a prefix of itself.

Note that when $i = j$, the above definition implies consistency with one’s future self.

4 A Nakamoto-Like Varying Difficulty Blockchain

In this section, we formally specify a variant of Nakamoto’s varying difficulty blockchain for which we can prove security. The main difference from Nakamoto’s varying difficulty blockchain is in the way we constrain timestamps in blocks which we will elaborate in this section.

Regarding our assumptions. Our provably secure variant adopts the same clock synchrony assumptions as the original Nakamoto with varying difficulty. We stress that although we describe the protocol assuming that nodes have perfectly synchronized clocks, our results also extend to a model where nodes have clocks that are weakly synchronized — as Pass and Shi [PS17c] point out, clock offsets can be charged to the network delay.

4.1 Valid Blocks and Blockchain

We first state basic validity rules for blocks and blockchains.

Valid blocks. Each block B is of the format

$$B := (h_{-1}, \text{txs}, \eta, \text{time}, p, h)$$

where $h_{-1} \in \{0, 1\}^\kappa$ is the hash of the chain that the block extends from, $\text{txs} \in \{0, 1\}^*$ denotes a set of transactions to be confirmed (or any payload), $\eta \in \{0, 1\}^\kappa$ denotes a puzzle solution, $\text{time} \in \{0, 1\}^*$ denotes the purported time at which this block was mined, p denotes the difficulty parameter of this block, and $h \in \{0, 1\}^\kappa$ denotes the hash of the present block. A block is said to be *valid* iff the following holds:

- $\text{H.ver}((h_{-1}, \text{txs}, \eta, \text{time}, p), h) = 1$; and
- $h < D_p$, where D_p is chosen such that a random string from $\{0, 1\}^\kappa$ is less than D_p with probability p .

Valid blockchain. A blockchain in the real-world protocol, henceforth denoted chain , is a chain of blocks $\text{chain} := (\text{genesis}, B_1, B_2, \dots, B_\ell)$ for some $\ell \in \mathbb{N}$, where

$$\text{genesis} := (\perp, \perp, \perp, t = 0, p = \infty, \perp)$$

is a special block said to be the genesis block.

We say that a blockchain $\text{chain} := (\text{genesis}, B_1, B_2, \dots, B_\ell)$ is *valid* iff the following hold:

- All of B_1, \dots, B_ℓ are valid blocks;
- For each $i \in [\ell]$, it holds that $B_i.h_{-1} = B_{i-1}.h$ — by convention we let $B_0 := \text{genesis}$. In other words, each block in the chain must extend the previous block;
- For each $i \in [\ell]$, it holds that $B_i.\text{time} > B_{i-1}.\text{time}$, i.e., all timestamps in the blockchain must strictly increase;
- For each $i \in [\ell]$, it holds that $B_i.p = \text{getdiff}(\text{chain}[: i - 1])$ where getdiff is a subroutine defined in Figure 1.

Let chain denote a blockchain, for convenience we define $\text{chain.time} := \text{chain}[-1].\text{time}$, i.e., a chain 's timestamp is its last block's timestamp.

4.2 A Provable Variant of Nakamoto's Varying Difficulty Blockchain

We now present the entire protocol (henceforth referred to as the real-world protocol, whereas our proof will begin by analyzing a simpler, ideal-world protocol that captures the essence of the real-world protocol).

We begin by defining the chain preference rule, i.e., what it means for a blockchain to have the most work.

Longest chain in terms of total work. We use the notation $|\text{chain}|$ to denote the actual length of chain (i.e., number of blocks). We use the notation $\|\text{chain}\|$ to denote the “total work” accumulated in chain — more specifically,

$$\|\text{chain}\| := \sum_{i=1}^{|\text{chain}|} \frac{1}{\text{chain}[i].p}$$

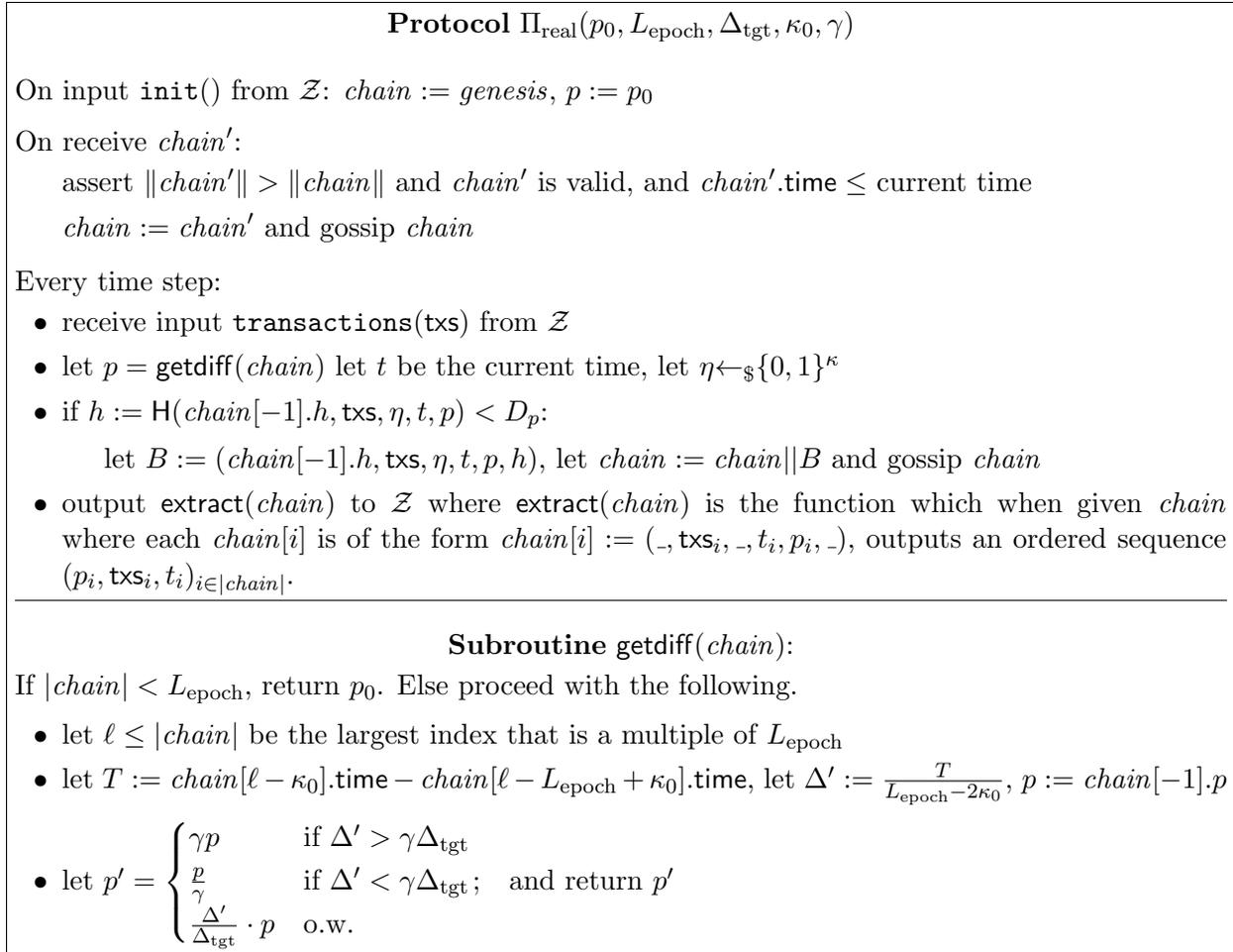


Figure 1: Real-world blockchain protocol with difficulty adjustment.

Real-world protocol. Figure 1 describes the protocol with difficulty adjustment. At a high-level, the protocol measures roughly how many nodes there have been in the recent past, and adjusts the difficulty parameter such that the expected block interval approximates a parameter Δ_{tgt} , which, as noted later, should be set to be an appropriate constant fraction larger than the maximum network delay Δ . In Nakamoto’s blockchain, the difficulty adjustment is performed every epoch containing $L_{\text{epoch}} = \Theta(\kappa)$ blocks. Recall that our ideal-world analysis requires that for roughly $\Theta(\kappa)$ blocks of time, all difficulty parameters must be bounded apart by a constant γ . Thus, in our protocol, we use the constant γ to restrict how much the difficulty parameter can change in each epoch.

More concretely, to estimate how many nodes there have been in the recent past, the idea is to measure how long it took for the blockchain to grow roughly $L_{\text{epoch}} = \Theta(\kappa)$ blocks. Achieving this requires that nodes place timestamps in blocks. While honest nodes always faithfully report timestamps in blocks, corrupt nodes can put in arbitrary timestamps. To ensure the safety of the protocol, we need to enforce the relative accuracy of timestamps in blocks, since otherwise these timestamps cannot be used to accurately measure the recent number of nodes. Thus, we require that the timestamps contained in blocks be strictly increasing, and that honest nodes always reject

timestamps that are in the future. Due to chain quality, every corrupt block must be sandwiched between two honest blocks that are close by (or genesis/end-of-chain). Therefore our blockchain timestamp rule ensures that adversarial blocks contain timestamps that are at most an ϵ factor off from the truthful time, where ϵ is an arbitrarily small constant.

Remark 4.1 (Regarding the timestamping rule). *We note that Nakamoto’s blockchain [Nak08] has a similar blockchain timestamping rule as ours (and as explained above, a rule of this kind is necessary to deal with adversarially injected timestamps). Further, Garay et al [GKL17]’s earlier varying difficulty blockchain analysis also adopted a similar timestamping rule as ours (but with some small modifications).*

Notations. We define the following notations.

- Let $m^t(\text{view})$ and $n^t(\text{view})$ denote the number of honest and corrupt nodes at time t in view respectively;
- Let p_0 denote the initial difficulty parameter hardcoded in the blockchain protocol,
- Let Δ_{tgt} be the targeted block interval (which must be set appropriately based on the maximum network delay Δ as explained later).
- Let γ be the bounded difficulty change parameter determined by Π_{real} ;
- Let χ be a constant related to how fast the mining power can change;
- Let ϕ be a constant related to the margin of honest fraction over corrupt fraction;
- Let ν be a constant related to how easy the puzzle can be relative to the total mining power and the maximum network delay Δ .

4.3 Compliant Executions

To prove the aforementioned Nakamoto variant secure, we will need to impose some mild constraints on the execution environment. At a very high level, we require that 1) the number of players do not change too abruptly; and 2) the initial difficulty is not too easy relative to the maximum network delay and the initial mining power. Further, as all earlier analyses for the fixed-mining-power setting, we assume that the maximum network delay Δ is known to the protocol, and further, in any round, the adversary controls only a minority of the total mining power. We now state these intuitive constraints precisely.

Let Δ_{tgt} be a function in κ , let ϕ, ν, χ, γ denote any positive constants. We say that a p.p.t. pair $(\mathcal{A}, \mathcal{Z})$ is Π_{real} -compliant w.r.t. parameters $(\phi, \nu, \chi, p_0, L_{\text{epoch}}, \Delta_{\text{tgt}}, \kappa_0, \gamma)$ iff for any κ , any view in the support of $\text{EXEC}^{\Pi_{\text{real}}}(\kappa, \mathcal{A}, \mathcal{Z})$, the following hold:

- *Bounded change in mining power.* For any t_0 and $t_1 := t_0 + W$ where $W := 4\chi^2\gamma L_{\text{epoch}}\Delta_{\text{tgt}}$, it holds that³

$$\frac{\max_{t=t_0}^{t_1} m^t(\text{view})}{\min_{t=t_0}^{t_1} m^t(\text{view})} \leq \chi$$

³In comparison, the original blockchain analysis works by Garay et al. [GKL15] and Pass et al. [PSS17] assumed fixed computing power throughout.

- *Majority honest.* The number of honest nodes must exceed corrupt ones by a constant margin in every round of view:

$$\text{for any round } t \leq |\text{view}|, \quad \frac{m^t(\text{view})}{n^t(\text{view})} > \frac{1 + \phi}{1 - \nu}$$

- *Safe start.* At the start of the execution, the honest mining rate must not be too high. More formally, let $m^1(\text{view})$ denote the number of honest nodes in the first round of view, it must hold that

$$p_0 m^1(\text{view}) < \frac{6\chi}{\Delta_{\text{tgt}}}$$

Remark 4.2 (Regarding bounded change on number of nodes). *In the above, we require that the mining power does not increase or decrease abruptly. However, we note that the proofs in this paper actually show that consistency and chain quality hold as long as the mining power does not increase too abruptly (and other relevant conditions are respected). Our proof later shows that should the mining power decrease suddenly, consistency and chain quality are nonetheless retained — only that the chain growth can be slow for a while should mining power drop suddenly. In the above, we stated a stronger condition where the mining power change must be bounded from both sides — since if so, we can state a stronger bound on chain growth, i.e., the chain grows at a rate such that the expected block interval at any time is a constant factor larger than the maximum network delay.*

Admissible parameters. Henceforth we say that $\Gamma_{\text{real}}(\phi, \nu, \chi, p_0, L_{\text{epoch}}, \Delta_{\text{tgt}}, \kappa_0, \gamma) = 1$ iff the following hold:

- ϕ, ν, χ, γ are positive constants independent of κ and the remaining parameters are polynomially bounded functions in κ ;
- $L_{\text{epoch}} > 8\kappa_0 \geq \kappa$
- $\chi < \gamma$
- $\nu < \frac{1}{4}$
- $\Delta < \frac{\Delta_{\text{tgt}}\nu}{12\chi}$, i.e., the targeted block interval must be an appropriate constant factor larger than the maximum network delay Δ .

Intuitive explanations for admissibility. We now explain intuitively why we impose the aforementioned constraints on the parameters.

- $L_{\text{epoch}} > 8\kappa_0 \geq \kappa$: First, epoch length being sufficiently long is needed for the ideal-world (and hybrid-world) analysis to work. The adversary can manipulate epoch boundaries in various ways to increase its advantage, e.g., choose the easier difficulty level to mine. The requirement for each epoch to be long enough bounds such adversarial advantage to an arbitrarily small constant amount.

Second, κ_0 needs to be $\Omega(\kappa)$ long but relatively short in comparison to the epoch length. When recalculating difficulty, we chop off κ_0 trailing blocks of the previous epoch for consistency — this explains why κ_0 must be sufficiently long. On the other hand, obviously κ_0 needs to be

smaller than the epoch’s length, and further, we wish to have sufficiently many blocks left for recalculating difficulty, even after removing the κ_0 trailing blocks of an epoch. Our protocol additionally removes the beginning κ_0 blocks of an epoch during difficulty recalculation — this avoids the need to deal with epoch boundary in the analysis of difficulty recalculation.

- $\chi < \gamma$: χ bounds the rate of change in mining power over any fixed window, while γ is the maximum difficulty change per epoch defined by Π_{real} . This requirement is needed for the protocol’s difficulty adjustment to be fast enough to track the change in mining power.
- $\nu < \frac{1}{4}$: recall that $\nu := 2\alpha_{\max}(\Delta + 1)$. Therefore, roughly speaking, this constraint on the constant ν determines how much larger the block interval must be in relation to the network delay. As Pass et al. [PSS17], the blockchain protocol is unsafe if the expected block interval is too small (i.e., the difficulty of mining is too small in relation to the network delay).

5 Main Theorems

In practice, when we run blockchain protocols, we choose an initial difficulty parameter p_0 based on our estimate of how many nodes there will be initially. If we happen to over-estimate, then the protocol will have a “safe start”. If our estimate happens to be somewhat accurate, then the protocol will have a “calibrated start”. We can prove the following statements about Π_{real} under either a safe or calibrated start:

1. Under a safe start and appropriate choice of parameters, the protocol Π_{real} achieves consistency and chain quality, and further, after a polynomially bounded warmup time, the protocol’s chain growth approximates the targeted block interval Δ_{tgt} up to a constant factor.
2. Further, if the starting mining power is not just upper bounded by a safe threshold, but in fact is “calibrated” — more specifically, suppose that $(\mathcal{A}, \mathcal{Z})$ respects the following for every view of non-zero support:

$$[\text{Calibrated start:}] \quad \frac{1}{2\chi\gamma\Delta_{\text{tgt}}} < p_0 m^1(\text{view}) < \frac{6\chi}{\Delta_{\text{tgt}}}$$

Then, the warmup time can be 0, i.e., start from the very beginning and throughout the execution, the chain growth always approximates Δ_{tgt} up to a constant factor.

The above intuition is summarized in the following theorem statement.

Theorem 5.1 (Π_{real} realizes a blockchain). *For any admissible parameters such that*

$$\Gamma_{\text{real}}(\phi, \nu, \chi, p_0, L_{\text{epoch}}, \Delta_{\text{tgt}}, \kappa_0, \gamma) = 1$$

for any constants $\epsilon, \epsilon' > 0$, and any $T_0 > \epsilon'\kappa$, it holds that $\Pi_{\text{real}}(p_0, L_{\text{epoch}}, \Delta_{\text{tgt}}, \kappa_0, \gamma)$ satisfies the following properties against any p.p.t. $(\mathcal{A}, \mathcal{Z})$ that is Π_{real} -compliant w.r.t. these parameters:

- $(T_0, g_0, g_1, t_{\text{warm}})$ -chain growth, where $g_0 := \frac{1}{3\chi^2\Delta_{\text{tgt}}}$, $g_1 := \frac{7\chi^2}{\Delta_{\text{tgt}}}$, and t_{warm} is some polynomially bounded function in κ .
- (T_0, μ) -chain quality where $\mu := 1 - \frac{1+\epsilon}{1+\phi}$;

- T_0 -consistency.

Further, if $(\mathcal{A}, \mathcal{Z})$ additionally respects the following “calibrated start” condition $\frac{1}{2\chi\gamma\Delta_{\text{tgt}}} < p_0 m^1(\text{view}) < \frac{6\chi}{\Delta_{\text{tgt}}}$, then $t_{\text{warm}} = 0$.

The remainder of the paper will mostly focus on proving the above theorem.

6 Analysis of a Core Randomized Experiment

Proving the above main theorem will involve multiple technical steps as we explained earlier in the technical roadmap.

The most mathematical part of the argument centers around a core randomized experiment that captures the essence of the nature of the stochastic process induced by the Nakamoto-like blockchain. In comparison with the full blockchain protocol, the core randomized experiment has a very clean definition as we present later in this section. However, proving sharp measure concentration bounds for this core randomized experiment requires somewhat sophisticated techniques relying on reasoning about moment generating functions of important random variables, and often conditioned on prefixes of the execution.

6.1 Core Randomized Experiment and Intuition

Randomized experiment. Consider the following simple randomized experiment. There is a mining oracle. An adversary \mathcal{A} adaptively specifies m queries p_1, p_2, \dots, p_T to the oracle — in particular, each query can depend on the outcomes of all previous queries. Upon receiving a query with the parameter p , the oracle flips a random coin that comes up with heads with probability p . If the coin comes up heads, \mathcal{A} earns $\frac{1}{p}$ amount of work.

We say that \mathcal{A} is (p_1, γ) -admissible iff 1) the first query submitted by \mathcal{A} is p_1 , and 2) in every possible execution trace of non-zero support, it holds that $\frac{1}{\gamma} \leq \frac{p_i}{p_j} \leq \gamma$.

Intuition: relation between this randomized experiment and our blockchain protocol.

In this section, we will present sharp measure concentration bounds for certain random variables defined over the above randomized experiment. In particular, the above randomized experiment captures the core stochastic nature of our blockchain protocol over every “ $\Theta(\kappa)$ blocks of time”. At 30,000 feet, the reader may think of each coin as an honest or corrupt node’s mining attempt in a blockchain protocol, and each mining attempt can be parametrized by a difficulty parameter p_i chosen by the adversary \mathcal{A} — note that in our later blockchain protocol, even honest nodes’ mining difficulty parameters can be influenced by the adversary. The more difficult a puzzle is, the proportionally more reward (i.e., work) is given out should the mining attempt be successful. In our later blockchain protocol, we require that over every “ $\Theta(\kappa)$ blocks of time”, all nodes’ difficulty parameters are at most a constant factor apart from each other — this explains why we require that the probability parameters in the above experiment have bounded difference. It turns out that *this bounded difference condition is necessary for the blockchain protocol to be secure*: if difficulty parameters can be an arbitrary polynomial factor apart, then there will can be an attack in which the adversary is lucky and mines a very difficult block with inverse-polynomial probability — and this very difficult block can allow the adversary to reverse $\Theta(\kappa)$ recent blocks in the blockchain, thus breaking consistency.

We will analyze two random variables related to the above randomized experiment:

- *Total work received (Section 6.2)*. Later in the blockchain protocol, honest node always favors the most-work chain. Thus the total work received by \mathcal{A} in the above experiment will later be used in our blockchain analysis to bound important variables such as adversarial successful work, i.e., how much work corrupt nodes can contribute to the blockchain over a duration of time — and adversarial successful work is a crucial variable in proving both chain quality and consistency.
- *Convergence opportunities (Section 6.3)*. We will define a notion of “convergence opportunities” for our simple randomized experiment. This notion directly corresponds to the notion of a convergence opportunity for our blockchain protocol, analogous to the way Pass et al. [PSS17] defined a convergence opportunity. At a very high level, a convergence opportunity is a good pattern where in Δ rounds, no honest nodes mine a block; then in one round, a single honest node mines a block; followed by another Δ rounds of silence in which no honest nodes mine a block. Later in our blockchain analysis, we will show that whenever this pattern happens, the adversary is forced to expense a certain amount of work to prevent convergence, and thus we can prove consistency by showing that the adversary cannot earn too much successful work (i.e., adversarial successful work upper bound as mentioned above).

As Garay et al. [GKL15] and Pass et al. [PSS17], these above random variables are core to reasoning about the security of a blockchain protocol. In a fixed-mining-power setting as in earlier works [GKL15, PSS17], bounding the total work equates to bounding the number of blocks, and the analyses were much easier since the adversary could not adaptively choose the difficulty parameter p . In our analyses below, we focus on bounding total work rather than the absolute number of blocks, and we address the challenges that arise due to the adversary’s ability to adaptively choose p .

6.2 Total Work Received

Let $\mathbf{W}^{\mathcal{A}}$ be a random variable representing the total amount of work received by \mathcal{A} in this randomized experiment; let $\mathbf{W}^{(p)}$ denote the total work received if the same parameter p is chosen for all queries.

Claim 6.1. *For any real t , the function $p \mapsto p(e^{\frac{t}{p}} - 1)$ is monotonically decreasing in $p \in [0, 1]$.*

Lemma 6.2 (Dominating Moment Generating Functions). *Suppose \mathcal{A} is (p_1, γ) -admissible. Then, for any real t , we have $\mathbf{E}[\exp(t\mathbf{W}^{\mathcal{A}})] \leq \mathbf{E}[\exp(t\mathbf{W}^{(\frac{p_1}{\gamma})})]$.*

Proof. Fixing real t , we write $\varphi(p) := p(e^{\frac{t}{p}} - 1) + 1$, which is monotonically decreasing in $p \in [0, 1]$ by Claim 6.1.

Consider the filtration $\{\mathcal{F}_i : i \in [T]\}$ of sigma-algebras, where \mathcal{F}_i corresponds to all the randomness generated up to (and including) the i th query.

For $i \in [T]$, we use $[1 : i]$ to denote the prefix of the first i queries. Then, it suffices to prove that for all $i \in [T]$, $\mathbf{E}[\exp(t\mathbf{W}^{\mathcal{A}}[1 : i])] \leq \mathbf{E}[\exp(t\mathbf{W}^{(\frac{p_1}{\gamma})}[1 : i])]$.

We prove by induction on i . For the base case $i = 1$, we have $\mathbf{E}[\exp(t\mathbf{W}^{\mathcal{A}}[1 : 1])] = p_1 e^{\frac{t}{p_1}} + (1 - p_1)e^0 = \varphi(p_1) \leq \varphi(\frac{p_1}{\gamma}) = \mathbf{E}[\exp(t\mathbf{W}^{(\frac{p_1}{\gamma})}[1 : 1])]$, where the inequality follows from Claim 6.1.

We next consider the inductive step, and assume that the inequality holds for some $1 \leq i < T$. Conditioning on \mathcal{F}_i , the contribution to the total work from queries up to step i is determined. Let Z_{i+1} denote the work received by \mathcal{A} in the $i+1$ st query. Hence, we have the following inequality on the random variable $\mathbf{E}[\exp(t\mathbf{W}^{\mathcal{A}}[1:i+1])|\mathcal{F}_i] = \exp(t\mathbf{W}^{\mathcal{A}}[1:i]) \cdot \mathbf{E}[\exp(tZ_{i+1})|\mathcal{F}_i] = \exp(t\mathbf{W}^{\mathcal{A}}[1:i]) \cdot \mathbf{E}[\varphi(p_{i+1})|\mathcal{F}_i] \leq \exp(t\mathbf{W}^{\mathcal{A}}[1:i]) \cdot \varphi(\frac{p_1}{\gamma})$, where the inequality comes from $p_{i+1} \geq \frac{p_1}{\gamma}$ and Claim 6.1. Taking expectation on the above inequality, we have

$$\begin{aligned} \mathbf{E}[\exp(t\mathbf{W}^{\mathcal{A}}[1:i+1])] &\leq \mathbf{E}[\exp(t\mathbf{W}^{\mathcal{A}}[1:i])] \cdot \varphi(\frac{p_1}{\gamma}) \\ &\leq \mathbf{E}[\exp(t\mathbf{W}^{(\frac{p_1}{\gamma})}[1:i])] \cdot \varphi(\frac{p_1}{\gamma}) = \mathbf{E}[\exp(t\mathbf{W}^{(\frac{p_1}{\gamma})}[1:i+1])] \end{aligned}$$

where the second inequality comes from the induction hypothesis. This finishes the inductive step and the proof. \square

Lemma 6.3 (Total work credited is concentrated around mean). *Suppose \mathcal{A} is (p_1, γ) -admissible for some $0 < p_1 < 1$ and $\gamma \geq 1$. Then, we have the following.*

- (a) For any $\epsilon > 0$, $\Pr[\mathbf{W}^{\mathcal{A}} \geq (1 + \epsilon)T] \leq \exp(-\frac{\epsilon^2}{2+\epsilon} \cdot \frac{p_1}{\gamma} \cdot T)$.
- (b) For any $0 < \epsilon \leq 1$, $\Pr[\mathbf{W}^{\mathcal{A}} \leq (1 - \epsilon)T] \leq \exp(-\frac{\epsilon^2}{2} \cdot \frac{p_1}{\gamma} \cdot T)$.

Proof. Lemma 6.2 states that the moment generating function of $\mathbf{W}^{\mathcal{A}}$ is dominated by that of $\mathbf{W}^{(\frac{p_1}{\gamma})}$. Observing that $\mathbf{E}[\mathbf{W}^{(\frac{p_1}{\gamma})}] = T$, the standard proof of the Chernoff bound using moment generating function gives the required result. \square

6.3 Convergence Opportunities

Henceforth assume that $T > 2m + 1$. We say that a sequence of consecutive $2m + 1$ coins centered around $m < i \leq T - m$ forms an m -convergence opportunity iff

- coin flips $i - m$ to $i - 1$ all come up tails;
- i -th coin flip comes up heads; and
- coin flips $i + 1$ to $i + m$ all come up tails.

In this case, sometimes we also say that i is a convergence opportunity for short. The *work* received by a convergence opportunity i is a random variable defined as $\frac{1}{p_i}$.

Lemma 6.4 (Moment Generating Function for Convergence Opportunity). *Suppose \mathcal{A} is (p_1, γ) -admissible for some $0 < p_1 < 1$ and $\gamma \geq 1$. Suppose further that $m < i \leq T - m$, and Z_i is the contribution of query i towards $\mathbf{C}^{\mathcal{A}}$. Let \mathcal{F} be the sub-sigma-algebra corresponding to the randomness up to (and including) some query $i' \leq i - m - 1$.*

Then, for $t < 0$, $\Pr[\exp(tZ_i)|\mathcal{F}] \leq \frac{\rho p_1}{\gamma} (e^{\frac{t\gamma}{p_1}} - 1) + 1$, where $\rho := (1 - \gamma p_1)^{2m}$.

Proof. Let E_i to be the event that i is a convergence opportunity. Let $\mathbf{p} := (p_{i-m}, \dots, p_i, \dots, p_{i+m})$ be the parameters chosen from query $i - m$ to $i + m$.

Equivalent experiment. We describe an equivalent experiment to determine whether each query in $[i - m : i + m]$ will return heads. For each $j \in [i - m : i + m]$, let X_j be sampled uniformly at random from $[0, 1]$ independently. Then, for each query j starting from $i - m$ to $i + m$, the outcome is sampled as follows:

1. The parameter p_j is chosen according to the outcomes in steps prior to query j ; in particular, this can depend on \mathcal{F} , p_ℓ 's and X_ℓ 's for $\ell < j$. The assumption on (p_1, γ) -admissibility guarantees that $\frac{p_1}{\gamma} \leq p_j \leq \gamma p_1$ holds with probability 1.
2. If $X_j \leq p_j$, the j th query will return heads; otherwise, it will return tails.

Define A to be the event that all queries in $[i - m : i - 1]$ return tails, and B to be the event that all queries in $[i + 1 : i + m]$ returns tails. Let \widehat{A} be the event that for all $j \in [i - m : i - 1]$, $X_j > \gamma p_1$, and \widehat{B} be the event that for all $j \in [i + 1 : i + m]$, $X_j > \gamma p_1$.

Define $\alpha := \Pr[A|\mathcal{F}]$. Observe that since the event \widehat{A} is independent of \mathcal{F} , $\alpha \geq \Pr[\widehat{A}|\mathcal{F}] = \Pr[\widehat{A}] = (1 - \gamma p_1)^m$. Define $\beta := \Pr[B|\mathcal{F}, A, \text{query } i \text{ is heads}]$. Similarly, we have $\beta \geq (1 - \gamma p_1)^m$. Hence, for $t < 0$, we have

$$\mathbf{E}[\exp(tZ_i)|\mathcal{F}] = \alpha \cdot \mathbf{E}[\exp(tZ_i)|\mathcal{F}, A] + (1 - \alpha) \quad (1)$$

$$= \alpha \cdot \mathbf{E}[p_i \beta \cdot e^{\frac{t}{p_i}} + 1 - p_i \beta | \mathcal{F}, A] + (1 - \alpha) \quad (2)$$

$$= \alpha \cdot \mathbf{E}[\beta p_i \cdot (e^{\frac{t}{p_i}} - 1) | \mathcal{F}, A] + 1 \quad (3)$$

$$\leq \alpha \cdot \mathbf{E}\left[\frac{\beta p_1}{\gamma} \cdot (e^{\frac{t\gamma}{p_1}} - 1) | \mathcal{F}, A\right] + 1 \quad (4)$$

$$\leq (1 - \gamma p_1)^{2m} \cdot \frac{p_1}{\gamma} \cdot (e^{\frac{t\gamma}{p_1}} - 1) + 1, \quad (5)$$

where (4) follows from Claim 6.1 and (5) holds because $\alpha\beta \geq (1 - \gamma p_1)^{2m}$ and $t < 0$. \square

Partitioning Indices According to Positive/Negative Correlation. Observe that for indices i and j such that $|i - j| \leq m$, then at most one of them can be a convergence opportunity, i.e., the convergence opportunity events at the two indices are negatively correlated. On the other hand, for $m < |i - j| < 2m$, the corresponding opportunity events are positively correlated, because tails outcomes for queries strictly between i and j will make both convergence opportunity events at i and j more likely.

For $1 \leq \ell \leq 3$ and $j \geq 1$, define an *epoch* of indices as $I_j^{(\ell)} := [\ell m + 3(j - 1)m + 1 : \ell m + 3(j - 1)m + m]$. It suffices to consider the intervals such that $I_j^{(\ell)} \cap [m + 1 : T - m]$ is non-empty. For $1 \leq \ell \leq 3$, define the *portion* $I^{(\ell)} := (\cup_{j \geq 1} I_j^{(\ell)}) \cap [m + 1 : T - m]$. For simplicity of notation, we assume that $T - 2m$ is divisible by $3m$ such that each $I^{(\ell)}$ contains the same number of complete epochs.

We need the following result on negative associativity [DR98] and for completeness we present its proof in the supplemental materials.

Lemma 6.5 (Negative Association for Competing Random Variables). *Let $\mathcal{X} := \{X_i : i \in [n]\}$ be a collection of non-negative random variables such that with probability 1, at most one of the X_i 's is non-zero. Then, the collection \mathcal{X} of variables are negatively associated.*

Lemma 6.6 (Moment Generating Function of a Portion). *Suppose the adversary is (p_1, γ) -admissible. For $\ell \in [3]$, let \mathbf{C}_ℓ denote the work received by m -convergence opportunities due to queries with indices in $I^{(\ell)}$. Then, for $t < 0$,*

$\mathbf{E}[\exp(t\mathbf{C}_\ell)] \leq [\mathbf{M}(t)]^{|I^{(\ell)}|}$, where $\mathbf{M}(t) := (1 - \gamma p_1)^{2m} \cdot \frac{p_1}{\gamma} \cdot (e^{\frac{t\gamma}{p_1}} - 1) + 1$ and $|I^{(\ell)}|$ is the number of indices in $I^{(\ell)}$.

Proof. For $j \geq 1$, let Y_j be the contribution towards \mathbf{C}_ℓ due to indices in epoch $I_j^{(\ell)}$. Then, it suffices to prove by induction on k that $\mathbf{E}[\exp(t \sum_{j=1}^k Y_j)] \leq [\mathbf{M}(t)]^{km}$.

Induction. The base case $k = 0$ is trivial. Assume that for some $k \geq 0$, $\mathbf{E}[\exp(t \sum_{j=1}^k Y_j)] \leq [\mathbf{M}(t)]^{km}$. We next consider the contribution Y_{k+1} due to epoch $I_{k+1}^{(\ell)}$.

Let \mathcal{F} is the sub-sigma-algebra corresponding to the randomness up to (and including) query with index $\ell m + 3km - m$ such that conditioning on \mathcal{F} , $\sum_{j=1}^k Y_j$ is completely determined, but none of the heads/tails outcomes relevant to Y_{k+1} are revealed yet.

Therefore, we have $\mathbf{E}[\exp(t \sum_{j=1}^{k+1} Y_j) | \mathcal{F}] = \exp(t \sum_{j=1}^k Y_j) \mathbf{E}[\exp(t Y_{k+1}) | \mathcal{F}]$.

To complete this inductive step, we use the following claim.

Claim. $\mathbf{E}[\exp(t Y_{k+1}) | \mathcal{F}] \leq [\mathbf{M}(t)]^m$.

Assuming the truth of this claim, we have $\mathbf{E}[\exp(t \sum_{j=1}^{k+1} Y_j) | \mathcal{F}] \leq \exp(t \sum_{j=1}^k Y_j) \cdot [\mathbf{M}(t)]^m$. Hence, taking expectation again and using the induction hypothesis finishes the inductive step.

Proof of Claim. For $\iota \in [m]$, let Z_ι denote the contribution to Y_{k+1} due to index $\ell m + 3km + \iota$; we have $Y_{k+1} = \sum_{\iota \in [m]} Z_\iota$. Observe that the collection $\{Z_\iota : \iota \in [m]\}$ of random variables are non-negative. Moreover, even conditioning on \mathcal{F} , with probability 1, at most one of them is non-zero. Hence, Lemma 6.5 states that they are negatively associated.

Hence, we have $\mathbf{E}[\exp(t \sum_{\iota=1}^m Z_\iota) | \mathcal{F}] \leq \prod_{\iota=1}^m \mathbf{E}[\exp(t Z_\iota) | \mathcal{F}] \leq \mathbf{M}(t)^m$, where the last inequality follows from Lemma 6.4. \square

Lemma 6.7 (Convergence opportunities). *Suppose that adversary \mathcal{A} is (p_1, γ) -admissible, and m is an integer such that $\rho := (1 - \gamma p_1)^{2m}$. Let $\mathbf{C}^{\mathcal{A}}$ denote the total work received by m -convergence opportunities during the above randomized experiment. Then, for any $0 < \epsilon \leq 1$,*

$$\Pr[\mathbf{C}^{\mathcal{A}} \leq (1 - \epsilon) \cdot \rho \cdot (T - 2m)] \leq \exp\left(-\frac{\epsilon^2}{2} \cdot \frac{\rho p_1}{\gamma} \cdot \frac{T - 2m}{3}\right).$$

Proof. Fix some $t < 0$. By the convexity of the exponential function, we have $\exp(t \sum_{\ell=1}^3 \mathbf{C}_\ell) \leq \frac{1}{3} \sum_{\ell=1}^3 \exp(3t \mathbf{C}_\ell)$.

Hence, we have $\mathbf{E}[\exp(t \sum_{\ell=1}^3 \mathbf{C}_\ell)] \leq \frac{1}{3} \sum_{\ell=1}^3 \mathbf{E}[\exp(3t \mathbf{C}_\ell)] \leq [\mathbf{M}(3t)]^{\hat{T}}$, where $\hat{T} := \frac{T-2m}{3}$, $\mathbf{M}(3t) := \rho \cdot \frac{p_1}{\gamma} \cdot (e^{\frac{3t\gamma}{p_1}} - 1) + 1$, $\rho := (1 - \gamma p_1)^{2m}$, and the last inequality follows from Lemma 6.6.

Observe that $[\mathbf{M}(3t)]^{\hat{T}} = \mathbf{E}[\exp(t \sum_{i=1}^{\hat{T}} G_i)]$, where the G_i 's are independently and identically distributed random variables, each of which equals $\frac{3\gamma}{p_1}$ with probability $\frac{\rho p_1}{\gamma}$ and 0 with probability $1 - \frac{\rho p_1}{\gamma}$.

Finally, observing that $\mathbf{E}[\sum_{i=1}^{\hat{T}} G_i] = \rho \cdot (T - 2m)$, the standard argument of Chernoff bound using moment generating function gives the result. \square

7 Ideal-World Protocol

To prove our main theorem, our next step is to analyze an ideal protocol that sits somewhere in between our core randomized experiment and the real-world protocol. In comparison with the core randomized experiment described earlier, this ideal protocol is more close in nature to the real-world protocol. However, in the ideal protocol, we do not explicitly deal with the specific difficulty adjustment function that is part of the real-world protocol. To prove security properties about

the ideal protocol, we only require that mining difficulties for honest and corrupt nodes do not change too fast (and some additional standard compliance rules), a technical condition that we will specify precisely later. As mentioned earlier, this bounded difficulty change condition is necessary for a Nakamoto-like blockchain to be secure. Later in the analysis of the real-world protocol, we will show through an inductive argument that given the real world’s difficulty change function, the real world execution can be regarded as a special case of the ideal world execution — and thus all security properties proven in the ideal world will carry over to the real world.

Syntax. In every round, \mathcal{Z} inputs some (p, txs) to each honest node where p is a mining difficulty parameter, and txs is a set of transactions. A mining difficulty parameter of p indicates that a block is successfully mined with probability p .

In every round, each honest node outputs to the environment \mathcal{Z} an abstract blockchain denoted $\text{chain} := \{(p_i, \text{txs}_i)\}_{i \in [|\text{chain}|]}$ where each abstract block consists of a difficulty parameter p_i and a set of transactions txs_i .

Measuring work. The work of a block $(p_i, -)$ is defined as $\frac{1}{p_i}$, and the total work of chain is defined as

$$\|\text{chain}\| := \sum_{i=1}^{|\text{chain}|} \frac{1}{p_i}$$

We note that the notation $\|\text{chain}\|$ is different from $|\text{chain}|$ — the latter refers to the length of chain , while the former refers to the work contained in $\|\text{chain}\|$ and is often referred to as the *work-length* of chain .

Ideal protocol. Figure 3 presents a simple ideal protocol where nodes interact with an ideal functionality $\mathcal{F}_{\text{tree}}$ to mine blocks. $\mathcal{F}_{\text{tree}}$ keeps track of all valid chains. Upon receiving a query $(\text{chain}, (p, \text{txs}))$ from a node, if chain is presently a valid chain, then $\mathcal{F}_{\text{tree}}$ flips a coin of probability p to decide if a next block is mined. In the ideal protocol, the difficulty parameters p submitted to $\mathcal{F}_{\text{tree}}$ must respect a bounded change condition which is enforced by $\mathcal{F}_{\text{tree}}$. We now state this bounded difficulty change condition.

Bounded difficulty change. Let γ be some constant. Notice that $\mathcal{F}_{\text{tree}}$ performs a coin flip upon each **extend** query. Let p_i denote the difficulty parameter of the i -th such coin flip where $i \in [N]$. We say that p_i is (γ, K) -admissible the following holds: let $i, i-1, i-2, \dots, i-T$ be the longest preceding sequence of coin flips (in reverse order) such that $\sum_{j=i-T}^i p_j \leq K$, and let p_{\min} and p_{\max} denote the min and max difficulty parameters among the $(i, i-1, i-2, \dots, i-T)$ -th coin flips, then it must hold that $p_{\max} \leq \gamma \cdot p_{\min}$.

In the above, to deal with boundary conditions, we simply pretend that for any $j \leq 0$, the difficulty parameter of the j -th coin flip is p_0 . Similarly, throughout this paper, we assume $\text{chain}[i] = (p_0, \text{genesis})$ for every $i \leq 0$.

Compliant executions. Let $0 < \nu < 1$ and $\phi > 0$ be any constant. We say that $(\mathcal{A}, \mathcal{Z})$ is $\Pi_{\text{ideal}}(\gamma, p_0, K)$ -compliant w.r.t. parameter (ϕ, ν) iff the following hold for every view of non-zero support:

$\mathcal{F}_{\text{tree}}(\gamma, p_0, K)$
<p>On init: $\text{tree} := (p_0, \text{genesis})$</p> <p>On receive extend(chain, (p, txs)):</p> <p style="padding-left: 20px;">assert $\text{chain} \in \text{tree}$, $\text{chain} \parallel (p, \text{txs}) \notin \text{tree}$</p> <p style="padding-left: 20px;">assert p is (γ, K)-admissible</p> <p style="padding-left: 20px;">$\text{coin} \leftarrow_{\S} \text{Bernoulli}(p)$, assert $\text{coin} = 1$</p> <p style="padding-left: 20px;">append (p, txs) to chain in tree, and return “succ”</p> <p>On receive verify(chain): return $(\text{chain} \in \text{tree})$</p>

Figure 2: Ideal functionality $\mathcal{F}_{\text{tree}}$.

Protocol $\Pi_{\text{ideal}}(\gamma, p_0, K)$
<p>On init: $\text{chain} := (p_0, \text{genesis})$, henceforth let $\mathcal{F}_{\text{tree}} := \mathcal{F}_{\text{tree}}(\gamma, p_0, K)$</p> <p>On receive chain': if $\ \text{chain}'\ > \ \text{chain}\$ and $\mathcal{F}_{\text{tree}}.\text{verify}(\text{chain}') = 1$: $\text{chain} := \text{chain}'$, gossip chain</p> <p>Every time step:</p> <ul style="list-style-type: none"> • receive input (p, txs) from \mathcal{Z} • if $\mathcal{F}_{\text{tree}}.\text{extend}(\text{chain}, (p, \text{txs}))$ outputs “succ”: $\text{chain} := \text{chain} \parallel (p, \text{txs})$ and gossip chain • output chain to \mathcal{Z}

Figure 3: Ideal protocol Π_{ideal} .

- *Bounded difficulty change.* $\mathcal{F}_{\text{tree}}$ never aborts due to receiving a p value that is not (γ, K) -admissible.
- *Majority honest.* Honest mining power must exceed corrupt mining power by a constant margin for every round. More formally, let $\alpha_{\max} := \frac{\nu}{2(\Delta+1)}$, the following must hold:

$$\text{for any round } t \leq |\text{view}|, \quad \frac{m^t(\text{view})}{n^t(\text{view})} > \frac{1 + \phi}{1 - \nu} = \frac{1 + \phi}{1 - 2\alpha_{\max}(\Delta + 1)}$$

- *Bounded mining rate.* For any round t in **view**, for any p submitted by either honest or corrupt nodes to $\mathcal{F}_{\text{tree}}$ in round t , it holds that

$$p \cdot m^t(\text{view}) < \alpha_{\max} \quad \text{where } \alpha_{\max} := \frac{\nu}{2(\Delta + 1)}$$

Deferred proofs. In the interest of space, we provide a full analysis of security properties retained by the ideal protocol in the supplementary materials. Also in the supplementary materials, we show through an inductive proof that the real-world execution is a special case of the ideal world, and thus all security properties in the ideal world carry over to the real world.

References

- [BCL⁺05] Boaz Barak, Ran Canetti, Yehuda Lindell, Rafael Pass, and Tal Rabin. Secure computation without authentication. In *CRYPTO*, pages 361–377, 2005.
- [DN92] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *CRYPTO*, 1992.
- [DR98] Devdatt P. Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *Random Struct. Algorithms*, 13(2):99–124, 1998.
- [GKL15] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Eurocrypt*, 2015.
- [GKL17] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol with chains of variable difficulty. In *Crypto*, 2017.
- [MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995.
- [Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
- [PS17a] Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In *PODC*, 2017.
- [PS17b] Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. In *DISC*, 2017.
- [PS17c] Rafael Pass and Elaine Shi. The sleepy model of consensus. In *Asiacrypt*, 2017.
- [PSS17] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Eurocrypt*, 2017.
- [SZ15] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In *Financial Cryptography*, 2015.

A Analysis of the Ideal Protocol

Notations. We define some useful notations for the ideal protocol.

- `chain` is said to be an honest chain in view if some honest node outputs `chain` to \mathcal{Z} in some round in view;
- A block denoted `chain`[i] is said to be an honest block in view if some honest node called $\mathcal{F}_{\text{tree}}$.`extend`(`chain`[: $i - 1$], `chain`[i]) in view, and $\mathcal{F}_{\text{tree}}$'s corresponding coin comes up heads.
- A block denoted `chain`[i] is said to be a corrupt block in view if some corrupt node called $\mathcal{F}_{\text{tree}}$.`extend`(`chain`[: $i - 1$], `chain`[i]) in view, and $\mathcal{F}_{\text{tree}}$'s corresponding coin comes up heads.
- Let $m^{t_0:t_1}(\text{view}) := \sum_{r=t_0}^{t_1} m^r(\text{view})$ be the honest mass during the window $[t_0, t_1]$;
- Let $n^{t_0:t_1}(\text{view}) := \sum_{r=t_0}^{t_1} n^r(\text{view})$ be the honest mass during the window $[t_0, t_1]$;

- Given a `chain` such that $\|\text{chain}\| \geq w_1$, we define $\text{chain}\langle w_0 : w_1 \rangle$ to be an alias for $\text{chain}[i : j]$ where i is the latest block such that $\|\text{chain}[i - 1]\| \leq w_0$ and j is the earliest block such that $\|\text{chain}[j]\| \geq w_1$.

Similarly, given `chain` such that $\|\text{chain}\| \geq w$, we use the notation $\text{chain}\langle w : \cdot \rangle$ to be an alias for $\text{chain}[i : \cdot]$ where i is the latest block such that $\|\text{chain}[i - 1]\| \leq w_0$; and we use the notation $\text{chain}\langle \cdot : w \rangle$ to be an alias for $\text{chain}[\cdot : j]$ where j is the earliest block such that $\|\text{chain}[j]\| \geq w$.

Fact A.1. *For any view of non-zero support, for any duration $[t_0, t_1]$ in which either $m^{t_0:t_1}(\text{view}) \leq K/(2\gamma p^{t_0}(\text{view}))$ or $m^{t_0:t_1}(\text{view}) + n^{t_0:t_1}(\text{view}) \leq K/(\gamma p^{t_0}(\text{view}))$ or $n^{t_0:t_1}(\text{view}) \leq K/(C \cdot \gamma p^{t_0}(\text{view}))$ where $C = \frac{2+\phi-\nu}{1-\nu}$ and $p^{t_0}(\text{view})$ denotes the minimum value of difficulty parameter p received by $\mathcal{F}_{\text{tree}}$ from an honest node in round t_0 , it must hold that all difficulty parameters received by $\mathcal{F}_{\text{tree}}$ during $[t_0, t_1]$ have difficulty parameters γ apart (unless the execution aborted).*

Proof. Follows directly from honest majority and bounded difficulty change compliance rules. \square

Additional shorthand terminology. Henceforth in all of our ideal-world proofs, we will assume that $(\mathcal{A}, \mathcal{Z})$ is $\Pi_{\text{ideal}}(\gamma, p_0, K)$ -compliant w.r.t. parameter (ϕ, ν) for some positive constant ϕ, ν . Henceforth in this section, whenever we say that “except with negligible probability over the choice of view, some event $\text{ev}(\text{view})$ holds”, we formally mean that for any positive constants ϕ, ν, γ , for any possibly unbounded $(\mathcal{A}, \mathcal{Z})$ that is $\Pi_{\text{ideal}}(\gamma, p_0, K)$ -compliant w.r.t. parameter (ϕ, ν) , for any $\kappa \in \mathbb{N}$,

$$\Pr[\text{view} \leftarrow_{\text{s}} \text{EXEC}_{\text{ideal}}^{\Pi}(\mathcal{A}, \mathcal{Z}, \kappa) : \text{ev}(\text{view}) = 1] \geq 1 - \text{negl}(\kappa)$$

A.1 Convergence Opportunities

We define a notion of convergence opportunity which will be useful both for the work growth proof and for the consistency proof later.

Convergence opportunity. We say that round $t \leq |\text{view}| - \Delta$ is a *convergence opportunity* in view iff

- No honest node mines a block in round $[t - \Delta, t)$;
- In round t , exactly one honest node mines a block;
- No honest node mines a block in round $(t, t + \Delta]$;

Let p denote the difficulty parameter of the honest block mined in round t , the work received by a convergence opportunity is defined as $\frac{1}{p}$. We also use the notation (t, p) -convergence opportunity to denote the fact that t is a convergence opportunity where the single honest block mined in round t has difficulty parameter p .

Lower bound on convergence opportunities. We now prove a lower bound on the total work received by convergence opportunities over any sufficiently long window. Henceforth, let $\mathbf{C}[t_0 : t_1](\text{view})$ be a random variable representing the total work received by convergence opportunities in the time window $[t_0 : t_1]$ in view.

Lemma A.2 (Lower bound on work received by convergence opportunities). *Let $K = \Theta(\kappa)$. For any positive constant ϵ, ϵ' , except with negligible probability over the choice of **view**, the following holds: for any time window $[t_0, t_1]$ such that $m^{t_0:t_1}(\mathbf{view}) \geq \epsilon K / (\gamma p^{t_0})$ where $p^{t_0} = p^{t_0}(\mathbf{view})$ denotes the difficulty parameter of the first honest coin flip in round t_0 , it holds that*

$$\mathbf{C}[t_0 : t_1](\mathbf{view}) > (1 - \epsilon')(1 - \nu)m^{t_0:t_1}(\mathbf{view})$$

Proof. Due to the union bound, it suffices to prove the above statement for any window $[t_0, t_1]$ such that $\epsilon K / (\gamma p^{t_0}) \leq m^{t_0:t_1}(\mathbf{view}) \leq K / (2\gamma p^{t_0})$ — since proving the statement for any longer window can be broken down to reasoning about polynomially many smaller windows.

Let us now consider a sequential view of the ideal world protocol, where in each round, the ideal functionality $\mathcal{F}_{\text{tree}}$ first flips coins one by one sequentially for honest nodes (henceforth referred to as honest coin flips), and then flips coins one by one for corrupt nodes. Henceforth, we use the notation $p^t(\mathbf{view})$ to denote the difficulty parameter of the first honest coin flip in round t in **view**.

Let $m_\Delta := 2\alpha_{\max}(\Delta + 1) = \nu$ be the maximum number of honest coin flips in a span of $\Delta + 1$ rounds. We say that the i -th honest coin in **view** is a m_Δ -convergence opportunity iff the following hold (where we focus on only honest coin flips performed by $\mathcal{F}_{\text{tree}}$ in **view**):

- honest coin flips $i - m_\Delta$ to $i - 1$ all come up tails;
- i -th honest coin flip comes up heads; and
- honest coin flips from $i + 1$ to $i + m_\Delta$ all come up tails.

Further, the work received by an honest coin that is a m_Δ -convergence opportunity is defined as $1/p$ where p is the difficulty parameter associated with the convergence opportunity.

Now, let $\tilde{\mathbf{C}}[t_0 : t_1](\mathbf{view})$ a random variable denoting the total work received by m_Δ -convergence opportunities for all honest coins flipped during rounds $[t_0, t_1]$. Now, it is not hard to see that for any **view**, for any time window $[t_0 : t_1]$,

$$\mathbf{C}[t_0 : t_1](\mathbf{view}) \geq \tilde{\mathbf{C}}[t_0 : t_1](\mathbf{view})$$

Having observed this, the rest of the proof follows in a straightforward manner from Lemma 6.7. In particular, due to the union bound, it suffices to show that for any fixed t_0 and $\epsilon K / (\gamma p^{t_0}) \leq T \leq K / (2\gamma p^{t_0})$, except for a negligible fraction of the **views**, $\tilde{\mathbf{C}}[t_0 \xrightarrow{T}](\mathbf{view})$ must be larger than $(1 - \epsilon')(1 - \nu)T$ where $\tilde{\mathbf{C}}[t_0 \xrightarrow{T}](\mathbf{view})$ denotes the work received by the next T honest coin flips in **view**, starting at the first honest coin flip in round t_0 . Let Υ denote all other random bits related to a **view** except for the T consecutive honest coin flips generated by $\mathcal{F}_{\text{tree}}$ starting at the beginning of t_0 . By Lemma 6.7, conditioned on any choice of Υ (since the choice of Υ can be regarded as being hard-coded in the adversary \mathcal{A} in Lemma 6.7), it must be that

$$\Pr[\tilde{\mathbf{C}}[t_0 \xrightarrow{T}](\mathbf{view}) > (1 - \epsilon')(1 - \nu)T] \geq 1 - \text{negl}(\kappa)$$

Thus it must be that except for a negligible fraction of **views**, $\tilde{\mathbf{C}}[t_0 \xrightarrow{T}] > (1 - \epsilon')(1 - \nu)T$. This concludes the proof. \square

A.2 Work Growth Lower Bound

Fact A.3 (Convergence opportunities contribute to work growth). *For any view of non-zero support, for any t_0, t_1 , any node i at time t_0 and any node j honest at time t_1 , we have the following where chain_i^t denotes the chain of node i that is honest in round t :*

$$\mathbf{C}[t_0 : t_1 - \Delta] \leq \|\text{chain}_j^{t_1}\| - \|\text{chain}_i^{t_0}\|$$

Proof. By the definition of a convergence opportunity and the honest protocol, if the minimum honest chain's work-length at the beginning of a convergence opportunity is w , then for any round after the convergence opportunity, every honest chain's work-length must be greater than $w + 1/p$ where $1/p$ is the work received by this convergence opportunity. The remainder of the proof follows by a straightforward inductive argument. \square

Theorem A.4 (Honest work growth lower bound). *Let $K = \Theta(\kappa)$. For any positive constants ϵ, ϵ' , except with negligible probability over the choice of view, the following holds: for any t_0 and t_1 such that $m^{t_0:t_1}(\text{view}) \geq \epsilon K / (\gamma p^{t_0}(\text{view}))$ for any node i at time t_0 and any node j honest at time t_1 , it holds that*

$$\|\text{chain}_j^{t_1}(\text{view})\| - \|\text{chain}_i^{t_0}(\text{view})\| \geq (1 - \epsilon')(1 - \nu)m^{t_0:t_1}(\text{view})$$

Proof. Due to Fact A.3 and Lemma A.2, for any positive constant ϵ'' , except with negligible probability over the choice of view, it must be that

$$\|\text{chain}_j^{t_1}(\text{view})\| - \|\text{chain}_i^{t_0}(\text{view})\| \geq (1 - \epsilon'')m^{t_0:t_1-\Delta}(\text{view})$$

Due to our compliance rules, $m^{t_0:t_1-\Delta}(\text{view}) \geq m^{t_0:t_1}(\text{view}) - m_\Delta = m^{t_0:t_1}(\text{view}) - \nu$. Since $0 < \nu < 1$, $m^{t_0:t_1}(\text{view}) \geq \epsilon K / (\gamma p^{t_0})$, and $K = \Theta(\kappa)$, for sufficiently large κ , it must be that for any positive constant ϵ' there exists a positive constant ϵ'' such that

$$(1 - \epsilon'')m^{t_0:t_1-\Delta}(\text{view}) \geq (1 - \epsilon'')(m^{t_0:t_1}(\text{view}) - \nu) \geq (1 - \epsilon')m^{t_0:t_1}(\text{view})$$

\square

A.3 Work Quality

Given t_0 and t_1 , let $\mathbf{A}(\text{view})[t_0 : t_1]$ be a random variable representing the amount of adversarial successful work during the window $[t_0, t_1]$ where adversarial successful work is defined as the total work contained in all blocks mined by corrupt nodes.

Lemma A.5 (Adversarial successful work upper bound). *Let $K = \Theta(\kappa)$. For any positive constants ϵ, ϵ' , except with negligible probability over the choice of view, the following holds: for any t_0, t_1 , it must be that*

$$\mathbf{A}[t_0 : t_1](\text{view}) < \max \left((1 + \epsilon)n^{t_0:t_1}(\text{view}), \frac{\epsilon' K}{\gamma p^{t_0}(\text{view})} \right)$$

Proof. It suffices to prove that except over negligible probability over the choice of `view`, for any t_0, t_1 , such that $n^{t_0:t_1} \geq \epsilon'K/((1+\epsilon)\gamma p^{t_0}(\text{view}))$, it holds that

$$\mathbf{A}[t_0 : t_1](\text{view}) < (1 + \epsilon)n^{t_0:t_1}(\text{view})$$

Now, due to the union bound, it suffices to prove that for any fixed t_0 , and $\epsilon'K/((1+\epsilon)\gamma p^{t_0}(\text{view})) \leq T \leq K/(C\gamma p^{t_0}(\text{view}))$ where $C = \frac{2+\phi-\nu}{1-\nu}$, for any positive constants ϵ, ϵ' , except with negligible probability over the choice of `view`, it holds that $\mathbf{A}[t_0 \xrightarrow{T}](\text{view})(1+\epsilon)n^{t_0:t_1}(\text{view})$ where $\mathbf{A}[t_0 \xrightarrow{T}]$ is a random variable denoting the amount of work earned by the adversary starting at the beginning of round t_0 and over the next T coin flips made by $\mathcal{F}_{\text{tree}}$ upon receiving `extend` requests from corrupt nodes.

Let Υ denote all other random bits in `view` except for the T consecutive coin flips made by $\mathcal{F}_{\text{tree}}$ upon adversarial `extend` queries starting at the beginning of round t_0 . Due to Lemma 6.3, conditioned on any choice of Υ , the probability that $\mathbf{A}[t_0 : t_1] \geq (1 + \epsilon)T$ must be negligible in κ . This concludes the proof. \square

Lemma A.6 (Weak work quality). *Let $K = \Theta(\kappa)$. For any positive constants ϵ and ϵ' , except with negligible probability over the choice of `view`, the following holds: for any honest chain denoted `chain` in `view`, for any positive w such that the block preceding `chain` is honest, and let p^* denote the difficulty parameter of this preceding honest block, for any $W \geq \epsilon K/(\gamma p^*)$, at least $\mu = 1 - \frac{1+\epsilon'}{1+\phi}$ fraction of the work in `chain` is contributed by honest nodes.*

Proof. Let `chain`[L] denote the honest block immediately preceding `chain` with $w + W$. If the block immediately after `chain` with $w + W$ is not mined by an honest node, we expand `chain` with $w + W$ to the right until we either encounter an honestly-mined block or end of `chain` — let `chain`[R] denote this honest block, and if end of `chain` is encountered, let $R = |\text{chain}| + 1$. Let t_0 and t_1 be the time at which `chain`[L] and `chain`[R] are mined respectively (or if $R = |\text{chain}| + 1$, let t_1 be the current time). Notice that due to our bounded difficulty change rule, since some honest node queried $\mathcal{F}_{\text{tree}}$ with p^* in round t_0 , all honest queries to $\mathcal{F}_{\text{tree}}$ in round t_0 must have a difficulty parameter within the range $[p^*/\gamma, \gamma p^*]$.

It suffices to prove that `chain`[$L + 1 : R - 1$] contains at least μ fraction of honest work — since our earlier expansion to the right could only make chain quality worse. Notice that all work contained in `chain`[$L + 1 : R - 1$] must be mined during the time frame $[t_0, t_1]$. Let W^* denote the total work contained in `chain`[$L + 1 : R - 1$]. Henceforth we ignore the negligible fraction of views where relevant bad events happen.

Due to work growth lower bound, it must be that for any positive constant $0 < \epsilon'' < 1$, $m^{t_0:t_1} < \frac{W^*}{(1-\epsilon'')(1-\nu)}$, i.e., the honest mass during the window $[t_0, t_1]$ is small.

Due to our compliance rule, it must be that $n^{t_0:t_1} < \frac{W^*}{(1-\epsilon'')(1+\phi)}$. Due to Lemma A.5, for any positive constant η , the amount of adversarial successful work must be upper bounded by $\frac{(1+\eta)W^*}{(1-\epsilon'')(1+\phi)}$.

Now, for any positive constant ϵ' , we must be able to find sufficiently small η and ϵ'' , such that

$$\frac{(1+\eta)W^*}{(1-\epsilon'')(1+\phi)} < \frac{(1+\epsilon')W^*}{1+\phi}$$

This concludes the proof. \square

Fact A.7 (Bounded difficulty change in blockchain). *Let $K = \Theta(\kappa)$. Except with negligible probability over the choice of view, the following holds: for any honest chain denoted chain in view, let $\text{chain}[j]$ be an honest block, and let $\text{chain}[j']$ be the earliest honest block such that $j' > j$ or if no such block exists, then let $\text{chain}[j'] = \text{chain}[-1]$, i.e., end of chain. Then, the following must hold: for any $j \leq i \leq i' \leq j'$, it must hold that*

$$\frac{1}{\gamma} \leq \frac{\text{chain}[i].p}{\text{chain}[i'].p} \leq \gamma$$

Proof. Ignore the negligible fraction of views where relevant bad events happen. Due to Lemma A.6, let $p^* = \text{chain}[j].p$, then for any positive constant ϵ , $\text{chain}[j + 1 : j']$ cannot contain more than $\epsilon K / (\gamma p^*)$ work. Let t and t' be the rounds in which some honest node mines $\text{chain}[j]$ and $\text{chain}[j']$ respectively — if $\text{chain}[j']$ is end of chain, let t' be the current time. Due to work growth lower bound, for any positive constant ϵ' , it must hold that $m^{t:t'} < \epsilon K / ((1 - \epsilon')\gamma p^*)$. Obviously every block in $\text{chain}[j + 1 : j']$ must be mined within the duration $(t, t']$. The remainder of the proof follows directly from our bounded difficulty change compliance rule, and by plugging in sufficiently small constants ϵ and ϵ' . \square

Fact A.8. *Let $K = \Theta(\kappa)$. Except with negligible probability over the choice of view, the following holds: let node i be honest in round r , and let chain be node i 's chain in round i , and let $p_{-1} := \text{chain}[-1].p$. Suppose that node i submits a query at difficulty p to $\mathcal{F}_{\text{tree}}$, it must hold that $\frac{1}{\gamma} \leq \frac{p}{p_{-1}} \leq \gamma$.*

Proof. Straightforward due to Fact A.7 and the fact that there is a non-negligible probability that the honest mining attempt is successful. \square

Theorem A.9 (Work quality). *Let $K = \Theta(\kappa)$. For any positive constants ϵ and ϵ' , except with negligible probability over the choice of view, the following holds: for any honest chain denoted chain in view, for any positive w , let p^* denote the difficulty parameter of the block preceeding $\text{chain}\langle w : \cdot \rangle$, for any $W \geq \epsilon K / (\gamma p^*)$, at least $\mu = 1 - \frac{1+\epsilon'}{1+\phi}$ fraction of the work in $\text{chain}\langle w : w + W \rangle$ is contributed by honest nodes.*

Proof. Denote $\text{chain}\langle w : w + W \rangle$ as $\text{chain}[L' : R]$. If the block preceeding $\text{chain}\langle w : w + W \rangle$ is not an honestly mined block, we expand $\text{chain}\langle w : w + W \rangle$ to the left, until we either encounter either an honestly mined block or genesis — let $\text{chain}[L]$ be this block where we stop. It suffices to prove that $\text{chain}[L + 1 : R]$ contains at least μ fraction of honest work except for a negligible fraction of the views. This follows in a straightforward manner from Lemma A.6 and Fact A.7. \square

A.4 Consistency

We say that adversarial successful work *covers* the range $(w, w + W]$ iff for any $w' \in (w, w + W]$, there exists some valid chain in $\mathcal{F}_{\text{tree}}$, such that the first block in $\text{chain}\langle w : \cdot \rangle$ is mined by corrupt nodes.

Fact A.10. *Let (t, p) be a convergence opportunity in view. Let $(w, w + 1/p]$ denote the work range covered by the single honest block mined in round t , and suppose that the adversarial successful work in view does not cover the range $(w, w + 1/p]$. Then, for any honest chain chain in view, $\text{chain}\langle w, w + 1/p \rangle = \mathbf{B}$ where \mathbf{B} is the single honest block mined in round t .*

Proof. After time $t + \Delta$ in view, all honest chains will have work-length at least $w + 1/p$. Further, by definition of a convergence opportunity, we know that besides the work mined in round t , there is no other honest successful work in view that covers the work range $(w, w + 1/p]$. Thus, if adversarial successful work does not cover this work range, then this work range in all honest chains must correspond to the block \mathbf{B} as defined above. \square

Theorem A.11 (Consistency after removing trailing work). *Let $K = \Theta(\kappa)$. For any positive constant ϵ , except with negligible probability over the choice of view, the following holds: for any r, t where $t \geq r$, let chain_r and chain_t denote two honest chains in round r and round t respectively. Then, for any $w > 0$ such that $\text{chain}_r\langle w : \rangle$ has at least $\epsilon K / (\gamma p^*)$ work where p^* is the difficulty parameter of the first block in $\text{chain}_r\langle w : \rangle$ it holds that*

$$\text{chain}_r\langle : w \rangle \prec \text{chain}_t$$

Proof. Suppose for the sake of reaching a contradiction that $\text{chain}_r\langle : w \rangle$ is not a prefix of chain_t . Let $w' < w$ be the largest value such that $\text{chain}_r\langle : w' \rangle \prec \text{chain}_t$, i.e., $\text{chain}_r\langle : w' \rangle \prec \text{chain}_t$ is the longest common prefix of chain_r and chain_t . Let $\text{chain}_r[i] = \text{chain}_t[i]$ be the latest honest block in $\text{chain}_r\langle : w' \rangle$, and let s be the round in which it was mined. It holds that all blocks in $\text{chain}_r[i + 1 :]$ and $\text{chain}_t[i + 1 :]$ must be mined after round s .

By Fact A.10, it must be the case that $\mathbf{C}[s : r - \Delta](\text{view}) \leq \mathbf{A}[s : r](\text{view})$, since otherwise, there must be an honest block \mathbf{B} mined during a convergence opportunity in $[s, r - \Delta]$, and \mathbf{B} must appear in both chain_r and chain_t . Below we prove that except with negligible probability over the choice of view, it must be that $\mathbf{C}[s : r - \Delta](\text{view}) > \mathbf{A}[s : r](\text{view})$ — if we can do so, then we reach a contradiction, and thus we can conclude the proof.

Below we ignore the negligible fraction of views where relevant bad events take place. By Lemma A.2, for any positive constant ϵ_c , it holds that

$$\mathbf{C}[s : r - \Delta](\text{view}) > (1 - \epsilon_c)(1 - \nu)m^{s:r-\Delta}(\text{view})$$

By Lemma A.5, for any positive constant ϵ_a , it holds that

$$\mathbf{A}[s : r](\text{view}) < (1 + \epsilon_a)n^{s:r}(\text{view})$$

By work growth lower bound and the fact that $\text{chain}_r[i + 1 :]$ contains more than $\epsilon K / (\gamma p^*)$ work, it must be that $m^{s:r} \geq 0.5\epsilon K / (\gamma p^*)$. Thus for any positive constants ϕ , ϵ , and $0 < \nu < 1$, there exist sufficiently small constants $\epsilon_c, \epsilon_a, \epsilon_1$ such that the following holds for sufficiently large κ :

$$\begin{aligned} & \mathbf{C}[s : r - \Delta](\text{view}) > (1 - \epsilon_c)(1 - \nu)m^{s:r-\Delta} \\ & \geq (1 - \epsilon_c)(1 - \nu)(m^{s:r} - m_\Delta) = (1 - \epsilon_c)(1 - \nu)(m^{s:r} - \nu) \\ & > (1 - \epsilon_1)(1 - \nu)m^{s:r} && \# \text{ recall that } m^{s:r} \geq 0.5\epsilon K / (\gamma p^*) \\ & > (1 - \epsilon_1)(1 + \phi)n^{s:r} && \# \text{ honest majority compliance} \\ & > (1 + \epsilon_a)n^{s:r}(\text{view}) > \mathbf{A}[s : r](\text{view}) \end{aligned}$$

\square

A.5 Work Growth Upper Bound

Fact A.12 (Total work upper bound). *Let $K = \Theta(\kappa)$. For any positive constants ϵ, ϵ_0 , except with negligible probability over the choice of view, the following holds: for any window $[t_0, t_1]$, total successful work during $[t_0, t_1]$ must be upper bounded by*

$$\max \left((1 + \epsilon)(m^{t_0:t_1}(\text{view}) + n^{t_0:t_1}(\text{view})), \frac{\epsilon_0 K}{\gamma p^{t_0}(\text{view})} \right)$$

Proof. The proof is almost identical to that of Lemma A.5, except that now we are concerned about $\mathcal{F}_{\text{tree}}$'s coin flips for both honest and corrupt nodes. \square

Theorem A.13 (Honest work growth upper bound). *Let $K = \Theta(\kappa)$. For any positive constants ϵ, ϵ_0 , except with negligible probability over the choice of view, the following holds: for any window $[t_0, t_1]$, for any node i at time t_0 and any node j honest at time t_1 , it holds that*

$$\|\text{chain}_j^{t_1}(\text{view})\| - \|\text{chain}_i^{t_0}(\text{view})\| < \max \left((1 + \epsilon)(m^{t_0:t_1}(\text{view}) + n^{t_0:t_1}(\text{view})), \frac{\epsilon_0 K}{\gamma p^{t_0}(\text{view})} \right)$$

Proof. Henceforth let

$$z^{r:r'}(\text{view}) := m^{r:r'}(\text{view}) + n^{r:r'}(\text{view})$$

Due to the union bound, it suffices to prove that for any positive constants ϵ, ϵ_0 , except with negligible probability over the choice of view, we have that for any $[t_0, t_1]$ such that $\frac{\epsilon_0 K}{(1+\epsilon)\gamma p^{t_0}(\text{view})} \leq z^{t_0:t_1}(\text{view}) \leq \frac{K}{\gamma p^{t_0}(\text{view})}$, for any node i at time t_0 and any node j honest at time t_1 , $\|\text{chain}_j^{t_1}(\text{view})\| - \|\text{chain}_i^{t_0}(\text{view})\| < (1 + \epsilon)z^{t_0:t_1}(\text{view})$. Suppose for the sake of contradiction that there exist positive constants ϵ, ϵ_0 , for a polynomial fraction of the views, we can find some $[t_0, t_1]$ such that $\frac{\epsilon_0 K}{(1+\epsilon)\gamma p^{t_0}(\text{view})} \leq z^{t_0:t_1}(\text{view}) \leq \frac{K}{\gamma p^{t_0}(\text{view})}$, and moreover, $\|\text{chain}_j^{t_1}(\text{view})\| - \|\text{chain}_i^{t_0}(\text{view})\| \geq (1 + \epsilon)z^{t_0:t_1}(\text{view})$. Among these polynomial fraction of views as said above, henceforth ignore the negligible fraction where relevant bad events happen, and for the remainder of views, the following statements hold.

Fist, for such a choice of $[t_0, t_1]$, it must be that all coin flips by $\mathcal{F}_{\text{tree}}$ during $[t_0, t_1]$ have difficulty parameters at most γ apart. Let chain_0 denote the least-work honest chain in round t_0 , and let chain_1 denote the most-work honest chain in round t_1 . Suppose that chain_0 and chain_1 are mined in rounds r_0 and r_1 respectively — by the definition of the honest protocol, it must be that $r_0 \leq t_0$ and $r_1 \leq t_1$.

- By Fact A.12, there exists a positive constant $\eta \geq \epsilon$ such that $z^{r_0:r_1} \geq (1 + \eta)z^{t_0:t_1}$ — since otherwise by Fact A.12, there cannot be more than $(1 + \epsilon)z^{r_0:r_1}$ total successful work between $[r_0, r_1]$.
- Since $z^{r_0:r_1} \leq z^{r_0:t_1}$, we have that $z^{r_0:t_1} = z^{r_0:t_0} + z^{t_0:t_1} \geq (1 + \eta)z^{t_0:t_1}$. This means that $z^{r_0:t_0} \geq \eta z^{t_0:t_1} \geq \epsilon z^{t_0:t_1}$.
- Now let $\text{chain}[i^*]$ denote the latest honest block in chain. By work quality, it must be that for any positive constant η , $\text{chain}[i^* + 1 :]$ has at most $\eta K / (\gamma p^*)$ work where $p^* = \text{chain}[i^*].p$. Further, by Fact A.7 and Fact A.8, it holds that p^* is at most γ^3 apart from $p^{t_0}(\text{view})$.

- The above means that there exists an honest node whose work length is at least $\|\text{chain}\| - \eta K/(\gamma p^*)$ at some time $s < r_0$. We also know that there is an honest node whose work length is $\|\text{chain}\|$ at time t_0 . This means that the minimal honest work growth between s and t_0 is at most $\eta K/(\gamma p^*)$. But earlier, we have shown that $z^{s:t_0} \geq z^{r_0:t_0} \geq \epsilon z^{t_0:t_1} \geq \frac{\epsilon \epsilon_0 K}{(1+\epsilon)\gamma p^{t_0}(\text{view})}$. Thus for any positive constants ϵ_0, ϵ , there exists a sufficiently small η such that by work growth lower bound, the minimal honest work growth between s and t_0 cannot be as small as $\eta K/(\gamma p^*)$. Thus we reach a contradiction. □

A.6 Some Corollaries

Corollary A.14 (Bounded blockchain difficulty change). *Let $K = \Theta(\kappa)$. For any positive constants ϵ , except with negligible probability over the choice of view, the following holds: for any honest chain denoted chain in view, for any w , for any $\text{chain}\langle w : w + (1 - \epsilon)K/(2\gamma p^*) \rangle$ where p^* is the probability of the first block in $\text{chain}\langle w : \cdot \rangle$, it holds that for any two blocks $\text{chain}[i], \text{chain}[j] \in \text{chain}\langle w : w + (1 - \epsilon)K/(2\gamma p^*) \rangle$, $\frac{1}{\gamma} \leq \frac{p_i}{p_j} \leq \gamma$ where $p_i = \text{chain}[i].p$ and $p_j = \text{chain}[j].p$.*

Proof. Below we ignore the negligible fraction of views where relevant bad events take place. Let $\text{chain}[L]$ the first honest block (or genesis) to the left of $\text{chain}\langle w : w + (1 - \epsilon)K/(2\gamma p^*) \rangle$, let $\text{chain}[M]$ the last honest block inside $\text{chain}\langle w : w + (1 - \epsilon)K/(2\gamma p^*) \rangle$. and let $\text{chain}[R]$ be the first honest block to the right of $\text{chain}\langle w : w + (1 - \epsilon)K/(2\gamma p^*) \rangle$ — or if no such block exists, let $R = |\text{chain}| + 1$. By Fact A.7, it holds that $\text{chain}[L].p$ and p^* are at most a multiplicative factor of γ apart. Let $\text{chain}[\ell : \ell']$ be an alias for $\text{chain}\langle w : w + (1 - \epsilon)K/(2\gamma p^*) \rangle$. By work quality, for any positive constant ϵ_0 , it holds that $\text{chain}[L + 1 : \ell - 1]$ cannot contain more than $\epsilon_0 K/(\gamma p^*)$ work. For a sufficiently small constant ϵ_0 , it holds that $\text{chain}[L + 1 : M]$ contains less than $(1 - 0.5\epsilon)K/(2\gamma p^*)$ work. Let r_0, r_1 , and r_2 denote the times at which $\text{chain}[L], \text{chain}[M], \text{chain}[R]$ are mined respectively (and if $R > |\text{chain}|$, let r_2 be the current time). By work growth lower bound, it holds that $m^{r_0:r_1}(\text{view}) < (1 - 0.25\epsilon)K/(2\gamma p^*)$. Since all blocks between $\text{chain}[L + 1 : M]$ must be mined in between $[r_0, r_1]$, it holds that for every $j, j' \in [L + 1 : M]$, $\frac{1}{\gamma} \leq \frac{\text{chain}[j].p}{\text{chain}[j'].p} \leq \gamma$.

Now, since all blocks in $\text{chain}[L + 1 : M]$ have difficulty parameters at most γ apart, and also by work quality, it holds that for any positive constant ϵ_1 , $\text{chain}[M + 1 : R - 1]$ has at most $\epsilon_1 K/(\gamma p^*)$ work. Thus, for a sufficiently small constant ϵ_1 , we have that $m^{r_0:r_2}(\text{view}) < (1 - 0.25\epsilon)K/(2\gamma p^*)$. We now have that for every $j, j' \in [L + 1 : R - 1]$, $\frac{1}{\gamma} \leq \frac{\text{chain}[j].p}{\text{chain}[j'].p} \leq \gamma$. □

Corollary A.15 (Consistency after removing trailing blocks). *Let $K = \Theta(\kappa)$. For any positive constant ϵ , except with negligible probability over the choice of view, the following holds: for any r, t where $t \geq r$, let chain_r and chain_t denote two honest chains in round r and round t respectively. Then, it holds that*

$$\text{chain}_r[: -\epsilon\kappa] \prec \text{chain}_t$$

Proof. We ignore the negligible fraction of views where relevant bad events happen. Due to Corollary A.14, and since $K = \Theta(\kappa)$, it holds that for a sufficiently small positive constant ϵ , $\|\text{chain}_r[-\epsilon\kappa :]\| \geq \epsilon\kappa/(\gamma p^*)$ where $p^* := \text{chain}_r[-\epsilon\kappa].p$. The remainder of the proof follows from Theorem A.11. □

B Hybrid Protocol

Before we analyze the real-world protocol, we first consider a hybrid-world protocol that is closer to the real-world protocol. In the hybrid world, every L_{epoch} blocks in the blockchain is called an *epoch*. All blocks in the same epoch has a unique difficulty parameter in every view where the difficulty parameter is chosen by the adversary, and difficulty parameters for adjacent epochs must have bounded change γ .

In the hybrid world, we will mainly prove a lemma that says even if the adversary mines into the past by extending chains that are much shorter than present honest chains, it does not help the adversary to break the relevant security properties including work growth, work quality, and consistency.

B.1 Hybrid Protocol Π_{hyb} : Epoch-Based Difficulty Adjustment

The hybrid-world protocol $\Pi_{\text{hyb}}(\gamma, p_0, L_{\text{epoch}})$ is almost identical to $\Pi_{\text{ideal}}(\gamma, p_0, K)$, except with the following modification: in $\Pi_{\text{hyb}}(\gamma, p_0, L_{\text{epoch}})$, instead of having each node specify the difficulty parameter with each **extend** query, the adversary \mathcal{A} is now responsible for informing \mathcal{F}_{hyb} of the difficulty parameter p^e for each epoch $e \in \mathbb{N}$. All blocks at lengths $[(e-1) \cdot L_{\text{epoch}} + 1, e \cdot L_{\text{epoch}}]$ belong to epoch e , and must adopt difficulty parameter p^e . The adversary \mathcal{A} is allowed to choose the difficulty parameters adaptively; however, the difficulty parameter for epoch e must already have been chosen before any node (corrupt or honest) calls \mathcal{F}_{hyb} to extend a block at length $(e-1)L_{\text{epoch}} + 1$ or greater; or else \mathcal{F}_{hyb} aborts.

B.2 Π_{hyb} is Secure Provided No Mining into the Past

We say that $(\mathcal{A}, \mathcal{Z})$ is *strongly compliant* w.r.t. Π_{hyb} iff

1. *Bounded difficulty change.* For every view of non-zero support, \mathcal{A} chooses the difficulty parameter for every epoch in a way that \mathcal{F}_{hyb} never aborts; further, for any adjacent epochs e and $e+1$, it must be the case that

$$\frac{1}{\gamma} \leq \frac{p^e}{p^{e+1}} \leq \gamma$$

2. *Majority honest and bounded mining rate.* Same as Π_{ideal} 's corresponding compliance rules — except that here the p for each **extend** query is decided by the corresponding epoch's difficulty.
3. *No mining in the past.* Corrupt nodes never ask $\mathcal{F}_{\text{tree}}$ to extend a chain whose length is more than $0.25L_{\text{epoch}}$ shorter than the current shortest honest chain.

Fact B.1 ((γ, K) -admissibility of strongly compliant hybrid-world execution). *Suppose that $L_{\text{epoch}} = \Theta(\kappa)$. Let view be an execution trace of $\Pi_{\text{hyb}}(\gamma, p_0, L_{\text{epoch}})$. Then, there exists a $K = \Theta(\kappa)$, such that except with negligible probability over the choice of view, the following holds: if view has not aborted, then all honest and corrupt \mathcal{F}_{hyb} queries in view are (γ, K) -admissible, i.e., the bounded difficulty rule adopted in Π_{ideal} is respected for K .*

Proof. By induction. The base case is obvious. It suffices to prove that if by round $t-1$, all \mathcal{F}_{hyb} queries are (γ, K) -admissible for a sufficiently small choice of $K = \Theta(\kappa)$, then all queries in round t , must be (γ, K) -admissible too (except for a negligible fraction of bad views).

Notice that as long as all \mathcal{F}_{hyb} queries have respected (γ, K) -admissibility, the security properties we have proven for Π_{ideal} are respected (except with negligible probability over the choice of view) thus far. Henceforth we ignore the negligible fraction of views where relevant bad events happen.

Let ϵ be a sufficiently small constant. Due to consistency, by the end of any round $r \leq t - 1$, all honest chains must be prefixes of each other except for the last $\epsilon\kappa$ blocks. Thus by the end of any round $r \leq t - 1$, honest nodes can be in at most two adjacent epochs.

Let e be the largest epoch any honest node reaches by the end of round $t - 1$. Let t' be the earliest round in which all honest nodes have reached length at least $(e - 2)L_{\text{epoch}} + 0.5L_{\text{epoch}}$, i.e., the middle point of epoch $e - 1$. Note that in this round t' , some honest node must still be in epoch $e - 1$, since otherwise, there must be an honest node whose chain length grows by at least $0.5L_{\text{epoch}}$ blocks in a single round — and this is impossible by work growth upper bound.

Thus during $[t', t]$, all honest nodes must be mining at difficulties either p^{e-1} or p^e that are at most γ apart. Since corrupt nodes are not allowed to mine off a chain that is $0.25L_{\text{epoch}}$ shorter than the shortest honest chain, between $[t', t]$, all corrupt nodes must be mining at either p^{e-1} or p^e too.

The remainder of the induction step proof is obvious by observing that due to work growth upper bound, $m^{t':t} + n^{t':t} \geq 0.5L_{\text{epoch}}/p^{e-1}$, and assuming that $K = \Theta(\kappa)$ is appropriately small. \square

We therefore conclude that work growth lower and upper bounds, work quality, and consistency hold for strongly compliant executions of Π_{hyb} as stated in the following lemma.

Lemma B.2 (Security properties of Π_{hyb} in strongly compliant executions). *Suppose that $L_{\text{epoch}} = \Theta(\kappa)$. Then for any $(\mathcal{A}, \mathcal{Z})$ that is strongly compliant w.r.t. Π_{hyb} , work growth lower and upper bound, work quality, consistency (after removing either trailing work or blocks) hold as stated in Section 7.*

Proof. Follows directly from Fact B.1. \square

B.3 Π_{hyb} is Secure Even with Mining into the Past

$(\mathcal{A}, \mathcal{Z})$ is *weakly* compliant w.r.t. Π_{hyb} iff $(\mathcal{A}, \mathcal{Z})$ satisfies “bounded difficulty change”, “majority honest”, and “bounded mining rate” as stated earlier, but now $(\mathcal{A}, \mathcal{Z})$ no longer needs to respect “no mining into the past”. In a weakly compliant execution, corrupt nodes are no longer required to not query \mathcal{F}_{hyb} on chains that are much shorter than current honest chains, i.e., the adversary is now more powerful. We show, however, that this added power does not help the adversary in breaking the security properties of Π_{hyb} .

Lemma B.3 (Security properties of Π_{hyb} in weakly compliant executions). *Suppose that $L_{\text{epoch}} = \Theta(\kappa)$. Then for any $(\mathcal{A}, \mathcal{Z})$ that is weakly compliant w.r.t. Π_{hyb} , work growth lower and upper bound, work quality, consistency (after removing either trailing work or blocks) hold as stated in Section 7.*

Proof. Consider some view such that in round r , the shortest honest chain is of length ℓ , and the adversary queries \mathcal{F}_{hyb} with `chain` and `txs`, where `chain`’s length is less than $\ell - 0.25L_{\text{epoch}}$ and `chain||(p, txs)` is not in \mathcal{F}_{hyb} where *p* is the difficulty for length $|\text{chain}| + 1$. Suppose that this mining attempt is successful in view. Let `chain'` := `chain||(p, txs)` be the result of the successful mining attempt. It suffices to show that except for a negligible fraction of views where relevant bad events happen, `chain'` cannot be a prefix of any honest chain in view.

Henceforth, we ignore the negligible fraction of views where relevant bad events happen. For the sake of contradiction, suppose that there is some honest chain chain^* in view such that $\text{chain}' \prec \text{chain}^*$. Let $\text{chain}[i]$ be an alias for chain' , and let $\text{chain}[j]$ be the first honest block in chain^* after the prefix $\text{chain}[i]$. By definition of the honest protocol, $j - i > 0.25L_{\text{epoch}}$. Let r' be the round in which $\text{chain}[j]$ was mined by an honest node.

Let W be the total work contained in $\text{chain}^*[i+1 : j-1]$. By work growth lower bound, it holds that $m^{r:r'} \leq (1 + \epsilon)W$ for any arbitrarily small constant ϵ . By the honest majority rule, $n^{r:r'} < (1 - \epsilon)W$ if ϵ is sufficiently small. Recall our core randomized experiment analysis for adversarial successful work upper bound. Then, it is not hard to see that for every $0.25L_{\text{epoch}}$ number of consecutive blocks in $\text{chain}^*[i+1 : j-1]$ that contain a total of W^* work, the adversary must make at least $(1 - 0.5\epsilon)W^*$ queries to \mathcal{F}_{hyb} to successfully mine all these blocks. Thus, there cannot be more than W adversarial successful work between $[r, r']$ — but now by construction, $\text{chain}^*[i+1 : j-1]$ must all be adversarial. Thus we reach a contradiction. \square

C Analysis of the Real-World Protocol

C.1 Simulator Construction

Modified hybrid-world protocol. Before we describe our simulator, we describe a slight variant of our hybrid protocol.

- a) Each hybrid-world block is now of the form (p, txs, t) , where t is referred to a block's *timestamp* similar to the timestamp field denoted **time** in the real-world protocol.
- b) Rather than calling $\mathcal{F}_{\text{hyb}}.\text{extend}(\text{chain}, \text{txs})$, honest nodes would now call $\mathcal{F}_{\text{hyb}}.\text{extend}(\text{chain}, (\text{txs}, t))$ where t denotes the current time. Recall that in the hybrid world, the difficulty parameter is specified by \mathcal{A} on a per-epoch basis to \mathcal{F}_{hyb} , and thus each **extend** query need not specify p .

Simulator construction. We construct a simulator $\mathcal{S}^{\mathcal{A}}$ that interacts with a real-world adversary \mathcal{A} in a blackbox manner, and participates in Π_{hyb} .

1. **Random oracle queries.** \mathcal{S} answers **H** and **H.ver** queries from \mathcal{A} in the following way.

- \mathcal{S} remembers all hash queries, and whenever \mathcal{A} asks a **H**(x) query that has been seen before, it answers in the same way as before.
- Whenever \mathcal{A} asks a **H.ver**(x, y) query, if there was a previous **H**(x) query where the answer returned was y , return 1; else return 0.
- If at any time, \mathcal{S} is about to return the same answer y to a hash query **H**(x) but y has been returned for a different hash query **H**(x') where $x' \neq x$, \mathcal{S} aborts outputting **collision-failure**.
- Whenever \mathcal{A} asks an **H**(x) query that has not been seen before, the simulator \mathcal{S} performs the following.
 - If x is of the form $x := (h_{-1}, \text{txs}, \eta, t, p)$, and the simulator has recorded a *chain* where $\text{chain}[-1] = (h_{-1}, -, -, t_{-1}, -)$ such that $t > t_{-1}$ and $p = \text{getdiff}(\text{chain})$: \mathcal{S} computes $\text{chain} := \text{extract}(\text{chain})$, Now, if $\mathcal{F}_{\text{hyb}}.\text{extend}(\text{chain}, (\text{txs}, t))$ outputs 1, \mathcal{S} picks a random y subject to $y < D_p$ and records the chain $\text{chain}' := \text{chain} || (h_{-1}, \text{txs}, \eta, t, y)$; else \mathcal{S} picks a random y subject to $y \geq D_p$.

- Else \mathcal{S} picks a random y .
 - Regardless of which case, \mathcal{S} checks to see if \mathcal{A} has included y in any earlier query. If so, \mathcal{S} aborts outputting **predict-failure**. Else, \mathcal{S} returns y .
2. **Simulated real-world internal state for honest nodes.** For each honest node j , the simulator simulates its honest behavior in the real world, and keeps track of its “simulated” real-world chain denoted $chain_j$. We shall describe how each honest node’s simulated real-world chain is updated later.
 3. **Process messages received from \mathcal{A} .** Whenever \mathcal{S} receives a $chain$ from \mathcal{A} destined for honest node j , \mathcal{S} checks the validity of $chain$ where $\mathbf{H.ver}$ is answered internally by \mathcal{S} itself. If $chain$ is valid but $\mathcal{F}_{\text{hyb}}.\text{verify}(\text{extract}(chain)) = 0$, abort outputting **tree-failure**. If $chain$ is valid and $chain.time \leq t_{\text{cur}}$ where t_{cur} is the current time,
 - if $chain$ is longer than node j ’s simulated real-world chain, \mathcal{S} updates node j ’s simulated real-world chain to $chain$;
 - \mathcal{S} forwards $\text{extract}(chain)$ to honest node j .
 4. **Simulate honest nodes’ successful mining attempts.** Whenever the simulator \mathcal{S} receives a message chain from some honest node j , if $chain$ has not been received before, parse $chain[-1] := (p, \text{txs}, t)$. Let $chain$ be node j ’s current simulated real-world chain that \mathcal{S} internally keeps track of as mentioned earlier. \mathcal{S} performs the following:
 - Parse $chain[-1] := (-, -, -, -, h_{-1})$.
 - Pick a random h subject to $h < D_p$; pick a random η . If h or η has been observed in any earlier query from \mathcal{A} , abort outputting **predict-failure**. If either h or η has been generated by \mathcal{S} internally before, abort outputting **collision-failure**. Else, remember h to be the answer to any future hash query $(h_{-1}, \text{txs}, \eta, p, t)$.
 - Record $chain || (h_{-1}, \text{txs}, \eta, t, p, h)$, and remember it as the simulated real-world chain for j .
 - Send $chain || (h_{-1}, \text{txs}, \eta, t, p, h)$ to \mathcal{A} .
 5. **Simulate honest nodes’ failed mining attempts.** At the end of every round, for every honest node j that did not send a message to \mathcal{S} in $\text{EXEC}^{\text{Ihyb}}$, the simulator looks up node j ’s simulated real-world chain denoted $chain$, and parses $chain[-1] := (-, -, -, -, h_{-1})$. The simulator generates a random η and a random y subject to $y \geq D_p$. If either h or η has been generated by \mathcal{S} internally before, abort outputting **collision-failure**. If either η or y is ever observed in any earlier query from \mathcal{A} (either in the past or in the future), abort outputting **predict-failure**. Else, record y as the answer to any future hash query $(h_{-1}, \text{txs}, \eta, p, t)$ where p is the difficulty \mathcal{F}_{hyb} would have chosen for mining off $\text{extract}(chain)$, t is the current time, and finally, txs is the set of transactions the honest nodes use for mining in this round — without loss of generality, we modify \mathcal{F}_{hyb} to disclose $(chain, (\text{txs}, t))$ to \mathcal{S} upon every honest mining request $\text{extend}(chain, (\text{txs}, t))$, note that this does not affect the analysis of our ideal or hybrid protocol.
 6. **Choose difficulty.** The simulator \mathcal{S} uses the following strategy to choose difficulty for the hybrid-world execution: whenever the simulator \mathcal{S} records a real-world chain denoted $chain$: for every $e \in N$, if $eL_{\text{epoch}} \leq |chain|$ and \mathcal{S} has not registered the difficulty for epoch e with \mathcal{F}_{hyb} ,

it computes p^e using $chain$ like in the real-world algorithm, and registers p with \mathcal{F}_{hyb} as the difficulty for epoch e .

If at any time during the execution, \mathcal{S} notices that a simulated honest chain would have computed a different difficulty parameter for any epoch e than the one \mathcal{S} submitted to \mathcal{F}_{hyb} , it aborts outputting consistency-failure.

7. **Corrupt and uncorrupt.** Whenever some honest node j becomes corrupt, \mathcal{S} discloses to \mathcal{A} the simulated real-world chain for j denoted $chain$ as node j 's internal state. Whenever some corrupt node j becomes uncorrupt, \mathcal{S} sets its simulated real-world chain to be $genesis$ and sets its $p := p_0$.
8. **Internal checks of Π_{hyb} compliance.** The simulator makes the following internal checks throughout the simulated execution. At the beginning of each round, the simulator \mathcal{S} performs internal checks to make sure that all of Π_{hyb} 's compliance rules have been satisfied thus far. If not, the simulator aborts outputting hyb-compliance-failure.

Fact C.1. *For every view of non-zero support from $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$, at the end of every round, as long as the view has not aborted, the following hold:*

- For every chain that \mathcal{S} has recorded, it holds that $\text{extract}(chain) \in \mathcal{F}_{\text{hyb}}$.
- For every chain $\in \mathcal{F}_{\text{hyb}}$, \mathcal{S} has recorded a chain such that $\text{extract}(chain) = chain$.

Fact C.2. *For every view of non-zero support from $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$, at the end of every round, if the execution has not aborted, then the following holds: suppose an honest node j 's chain is $chain_j$, then the simulator maintains a simulated real-world chain for j denoted $chain_j$, such that $\text{extract}(chain_j) = chain_j$.*

Fact C.3. *For every view of non-zero support from $\text{EXEC}^{\Pi_{\text{hyb}}}$, it holds that view does not abort due to tree-failure.*

Proof. Suppose for the sake of reaching a contradiction that at some round r , view aborts due to tree-failure and the relevant bad chain is denoted $chain$ — we now show that this is impossible given view has not aborted due to collision-failure or predict-failure.

By assumption $chain$ is valid but $\mathcal{F}_{\text{hyb}}.\text{verify}(\text{extract}(chain)) = 0$. Let $chain[: \ell]$ where $\ell \geq 1$ be the shortest prefix of $chain$ such that $\mathcal{F}_{\text{hyb}}.\text{verify}(\text{extract}(chain[: \ell])) = 0$. It holds that $\mathcal{F}_{\text{hyb}}.\text{verify}(\text{extract}(chain[: \ell-1])) = 1$. Parse $chain[\ell] := (h_{-1}, \text{txs}, \eta, \text{time}, p, h)$, and $chain[\ell-1] := (h'_{-1}, \text{txs}', \eta', \text{time}', p', h_{-1})$ where $h < D_p$ and $h_{-1} < D_p$. It holds that \mathcal{S} has recorded h to be the answer of the hash query $\text{H}(h_{-1}, \text{txs}, \eta, \text{time}, p)$; similarly, \mathcal{S} has recorded h_{-1} to be the answer of the hash query $\text{H}(h'_{-1}, \text{txs}', \eta', \text{time}', p')$. Since \mathcal{S} has not aborted due to collision-failure or predict-failure, it holds that \mathcal{S} cannot have recorded $\text{H}(h_{-1}, \text{txs}, \eta, \text{time}, p) = h$ when simulating successful honest mining, since if this is the case, due to no collision-failure, it must hold $\text{extract}(chain)[: \ell] \in \mathcal{F}_{\text{hyb}}$ when \mathcal{S} records $\text{H}(h_{-1}, \text{txs}, \eta, \text{time}, p) = h$. Therefore, the only possible way for \mathcal{S} to have recorded $\text{H}(h_{-1}, \text{txs}, \eta, \text{time}, p) = h$ is if \mathcal{A} has made a hash query for $\text{H}(h_{-1}, \text{txs}, \eta, \text{time}, p)$ in prior to r . Let $r' < r$ be the first round in which \mathcal{A} makes such a query. In this round r' , it must hold that \mathcal{S} has not recorded any $chain'$ that ends with the hash h_{-1} — if so, due to no-collision, then this recorded $chain'$ must agree with $chain[: \ell-1]$, and therefore after this hash query, $\text{extract}(chain)[: \ell] \in \mathcal{F}_{\text{hyb}}$. However, by no predict-failure and no collision-failure, it must hold that in round r' , \mathcal{S} must have

already recorded the hash entry $H(h'_{-1}, \text{txs}', \eta', \text{time}', p') = h_{-1}$. Due to no collision-failure, \mathcal{S} never records h_{-1} to be the answer of any other hash query. However, by construction of \mathcal{S} , then \mathcal{S} will also never record a real-world chain ending with h_{-1} — but we know that at round r , $\text{extract}(\text{chain})[: \ell - 1] \in \mathcal{F}_{\text{hyb}}$ and therefore in round r , \mathcal{S} must have recorded $\text{chain}[: \ell - 1]$. This reaches a contradiction. \square

Fact C.4. *For any Π_{real} -compliant p.p.t. pair $(\mathcal{A}, \mathcal{Z})$, all but $\text{negl}(\kappa)$ fraction of views from $\text{EXEC}^{\Pi_{\text{hyb}}}(1^\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$ do not abort due to collision-failure, predict-failure, or consistency-failure.*

Proof. This is easy to see, since collision-failure requires that \mathcal{S} generates two random numbers from $\{0, 1\}^\kappa$ that collide, and predict-failure requires that \mathcal{A} predicts a random number \mathcal{S} generates without seeing it. Since both \mathcal{S} and \mathcal{A} run in time $\text{poly}(\kappa)$, this does not happen except with $\text{negl}(\kappa)$ probability.

No consistency-failure arises directly from the consistency of Π_{hyb} . \square

C.2 Simple Facts about Hybrid-World Compliance

Assumptions and notations. Since our simulator \mathcal{S} internally checks hybrid-world compliance at the beginning of every round, as long as a view from the simulated execution $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}, \mathcal{Z})$ has not aborted, it holds that except with $\text{negl}(\kappa)$ probability over the choice of view, all honest nodes agree on the difficulty for the same length in the blockchain due to consistency of Π_{hyb} . Therefore henceforth we may ignore the bad views where the above fails to hold. For the remaining good views, we may speak about the difficulty of an epoch (as perceived by honest nodes).

We use the following notations:

- We use $p^e(\text{view})$ to denote the difficulty of epoch e in view;
- We use the notation $p^{t_0}(\text{view})$ to denote the difficulty corresponding to the epoch defined by the maximum honest chain length at time t_0 .

Hybrid-world compliance. Recall that \mathcal{S} internally checks its hybrid-world compliance at the beginning of every round. We now wish to show that for any Π_{real} -compliant p.p.t. $(\mathcal{A}, \mathcal{Z})$, all but $\text{negl}(\kappa)$ fraction of views from $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$ do not abort due to **hyb-compliance-failure**. Recall that Π_{hyb} has several compliance rules: “honest majority” follows directly from the same constraint on the real-world \mathcal{A} . Therefore, we only need to make sure that \mathcal{S} respects bounded mining rate, bounded change in difficulty, as well as Δ -bounded network delivery. Among these, Δ -bounded network delivery and bounded change in difficulty are easy to show and covered by the following simple facts.

Fact C.5. *For every Π_{real} -compliant $(\mathcal{A}, \mathcal{Z})$, $\mathcal{S}^{\mathcal{A}}$ delivers honest messages within Δ rounds in the hybrid world.*

Proof. By construction, \mathcal{S} only drops invalid chains from \mathcal{A} or if the chain’s timestamp is greater than the current time. Whenever \mathcal{S} sends some chain to \mathcal{A} on behalf of an honest node, it must be triggered by \mathcal{S} receiving $\text{extract}(\text{chain})$ from some honest node j . Further, such a chain must be valid and has a timestamp in the present. Since \mathcal{A} must forward chain within Δ delay to every other honest node, \mathcal{S} must forward $\text{extract}(\text{chain})$ to all other honest nodes in the hybrid execution too. \square

Fact C.6. For every $\Pi_{\text{real-compliant}}(\mathcal{A}, \mathcal{Z})$, except for a $\text{negl}(\kappa)$ fraction of views from $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$, $\mathcal{S}^{\mathcal{A}}$ respects bounded difficulty change.

Proof. Straightforward due to consistency of Π_{hyb} as well as by definition of the real-world difficulty change function (i.e., that difficulty change must be bounded by γ on both sides). \square

Henceforth, we focus on showing that all but $\text{negl}(\kappa)$ fraction of views from $\text{EXEC}^{\Pi_{\text{hyb}}}(\mathcal{S}^{\mathcal{A}}, \mathcal{Z})$ do not abort due to violation of “bounded mining rate”. In essence, what we need to show is that the difficulty adjustment function always chooses an appropriate difficulty parameter in light of the recent number of nodes.

C.3 Analysis of the Difficulty Adjustment Function

Henceforth, if a view aborts due to violating the “bounded mining rate” compliance rule, we say that it aborts due to high mining rate.

Proof intuition. Henceforth, we say that an honest node enters epoch e in round r in view if its chain length first reaches $(e - 1)L_{\text{epoch}}$ in round r in view. We say that a view enters epoch e in round r if the r is the first round in which some honest node enters epoch e in view.

Henceforth, let

$$C := 6\chi, \quad C_0 := \frac{C}{12\chi^2\gamma} = \frac{1}{2\chi\gamma}$$

Given a view sampled from $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}, \mathcal{Z})$, we say that epoch e starts *safe* in view iff

$$p^e(\text{view}) \cdot m^{t_0}(\text{view}) < \frac{C}{\Delta_{\text{tgt}}}$$

where t_0 is the first round in which an honest node enters epoch e . We say that epoch e starts *calibrated* in view iff

$$\frac{C_0}{\Delta_{\text{tgt}}} < p^e(\text{view}) \cdot m^{t_0}(\text{view}) < \frac{C}{\Delta_{\text{tgt}}}$$

where t_0 is the first round in which an honest node enters epoch e . Henceforth, we sometimes also say that an epoch is safe or calibrated for short — this means the same as the epoch starts safe or calibrated.

More intuitively, an epoch starts safe if at the epoch’s beginning, the honest mining rate is not too high. An epoch starts calibrated if at the epoch’s beginning, the honest mining rate is neither too high nor too low. Clearly, if an epoch starts calibrated in view, it must also start safe in view.

Informally, we wish to prove the following:

- a) If epoch e starts safe, then except with $\text{negl}(\kappa)$ probability, view will not abort due to high mining rate prior to entering epoch $e + 1$; further, epoch $e + 1$ will also start safe.

Note that to show that a view does not abort prior to entering the next epoch $e + 1$, it suffices to show that $p^r(\text{view})m^r(\text{view}) < \alpha_{\text{max}}$ for any round $r \in [t_0, t_1]$ where t_0 is the round in which view enters epoch e and t_1 is the time of entering epoch $e + 1$. Henceforth if this condition fails, we say that view aborts due to high mining rate.

- b) If epoch e not only starts safe but also starts calibrated, then epoch $e + 1$ will also start calibrated.

In our proof, we will analyze the following two cases one by one:

- **Case 1:** an epoch e starts safe but not calibrated, i.e., $p^e(\text{view}) \cdot m^{t_0}(\text{view}) < \frac{C_0}{\Delta_{\text{tgt}}}$. In this case, we wish to show that except with $\text{negl}(\kappa)$ probability, view does not abort due to high mining rate prior to entering the next epoch; and further, the next epoch starts out safe.

In this case, at the start of the epoch, the mining rate can be very small, and therefore the epoch may take much longer than W time to complete. In this case, our estimation of the mining power by taking average over the last epoch may not give an accurate estimate of the most recent mining power — since mining power may have changed quite a lot during this long epoch. However, intuitively, once the mining power becomes sufficiently high such that the expected block interval becomes close to Δ_{tgt} , the current epoch must finish off quickly. Due to the slowing changing nature of the mining power, the current epoch cannot finish with an unusually high mining power. Further, even though our mining power estimate may be off, the worst case is that the difficulty parameter adjusts by γ .

In this case, we can prove that “appropriate mining rate” is maintained till the next epoch, and further, the aforementioned invariant is maintained at the beginning of the next epoch — more specifically, let t_1 be the first round in which an honest node enters the next epoch, it must hold that

$$p^{e+1}(\text{view}) \cdot m^{t_1}(\text{view}) < \frac{C}{\Delta_{\text{tgt}}}$$

- **Case 2:** epoch e starts out calibrated, i.e., $\frac{C_0}{\Delta_{\text{tgt}}} < p^e(\text{view}) \cdot m^{t_0}(\text{view}) < \frac{C}{\Delta_{\text{tgt}}}$.

In this case, the epoch begins with sufficient mining power such that the expected block interval is close to Δ_{tgt} . We can show that the epoch must end within W amount of time, and therefore mining power cannot have changed too much during the epoch. Further, our estimation of mining power by taking average over the epoch must be somewhat accurate. Therefore, we can conclude in this case that appropriate mining rate is respected till the next epoch, and further, the aforementioned invariant is maintained at the beginning of the next epoch.

C.3.1 Calibrated Leads to Calibrated

First, we prove a helpful fact: except with $\text{negl}(\kappa)$ probability, once an epoch enters a state of being calibrated, the epoch will end quickly without aborting. More formally, let $\text{bad_epoch}(\text{view}) = 1$ be the following bad event: there exists a time t_0 , let e denote the epoch corresponding to the maximum honest chain length at time t_0 , we have that

- $\frac{C_0}{\Delta_{\text{tgt}}} \leq p^e(\text{view}) \cdot m^{t_0}(\text{view}) \leq \frac{C}{\Delta_{\text{tgt}}}$;
- however, either view aborted due to high mining rate prior to an honest node enters epoch $e + 1$ or the first honest node enters epoch $e + 1$ after $t_0 + W$.

Fact C.7. For any Π_{real} -compliant $(\mathcal{A}, \mathcal{Z})$, any sufficiently large κ ,

$$\Pr[\text{view} \leftarrow_{\S} \text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z}) : \text{bad_epoch}(\text{view}) = 1] \leq \text{negl}(\kappa)$$

Proof. By union bound over the choice of t_0 , it suffices to prove the above for any fixed t_0 .

Without loss of generality, we assume that for more than $\text{negl}(\kappa)$ fraction of the views it holds that $\frac{C_0}{\Delta_{\text{tgt}}} \leq p^e(\text{view}) \cdot m^{t_0}(\text{view}) \leq \frac{C}{\Delta_{\text{tgt}}}$, since otherwise, the fact trivially holds. Consider such a view. Suppose view aborted due to high mining rate prior to entering epoch $e + 1$. Due to bounded change in mining power and the fact that $\frac{C\chi}{\Delta_{\text{tgt}}} \leq \frac{\nu}{2\Delta}$, it must hold that by time $t_0 + W$, no honest node has entered epoch $e + 1$ yet (and the abort cannot have happened before $t_0 + W$).

Therefore, it suffices to show that for $\text{negl}(\kappa)$ fraction of such views, at some $t' < t_0 + W$, some honest node must have entered epoch $e + 1$.

Suppose that for more than $\text{negl}(\kappa)$ fraction of such views, by time $t_0 + W$, no honest node has entered epoch $e + 1$. Due to bounded change in mining power, it holds that for any such view for any $t \in [t_0, t_0 + W]$, $p^e(\text{view}) \cdot m^t(\text{view}) > \frac{C_0}{\chi \cdot \Delta_{\text{tgt}}}$.

Moreover, for all but $\text{negl}(\kappa)$ fraction of such views, work growth lower bound holds. If we ignore views where work growth lower bound fail, it holds that for any such view, honest chain must have growth more than L_{epoch} by time $t_0 + W$, as long as

$$W > \frac{2L_{\text{epoch}}\Delta_{\text{tgt}}\chi}{C_0} = 4\chi^2\gamma L_{\text{epoch}}\Delta_{\text{tgt}}$$

but this contradicts our assumption. \square

Fact C.8. *Except with $\text{negl}(\kappa)$ probability over the choice of view, the following must hold for $\text{EXEC}^{\text{Ihyb}}(\mathcal{S}^A, \mathcal{Z})$ for any constant $\epsilon > 0$: let e be any epoch, let ℓ denote the ending length of epoch e . Suppose that \mathcal{S}^A calls $\text{getdiff}(\text{chain}[:\ell])$ at some point in view in order to set the difficulty of $\text{extract}(\text{chain}[:\ell - \kappa_0])$ with \mathcal{F}_{hyb} , and let $r := \text{chain}[\ell - L_{\text{epoch}} + \kappa_0].\text{time}$, let $r' := \text{chain}[\ell - \kappa_0].\text{time}$, then the following hold:*

- $t_0 \leq r \leq r' \leq t_1$ where t_0 and t_1 denote the rounds in which view enters epochs e and $e + 1$ respectively;
- $L_{\text{epoch}} - 4\kappa_0 \leq \text{min_honest_gr}^{r:r'}(\text{view}) \leq \text{max_honest_gr}^{r:r'}(\text{view}) \leq L_{\text{epoch}}$
- $(1 - \epsilon) \frac{L_{\text{epoch}} - 4\kappa_0}{2p^e(\text{view})} \leq m^{r:r'}(\text{view}) \leq \frac{1 + \epsilon}{1 - 2\alpha_{\text{max}}\Delta} \cdot \frac{L_{\text{epoch}}}{p^e(\text{view})}$

Proof. Ignore all views where chain quality and work growth fail. It holds that there is an honest block in the range $\text{chain}[\ell - L_{\text{epoch}} : \ell - L_{\text{epoch}} + 2\kappa_0]$ and in the range $\text{chain}[\ell - 2\kappa_0 : \ell]$. The remainder of the proof is obvious: due to work growth upper bound and the majority honest compliance rule, it holds that $(1 - \epsilon) \frac{L_{\text{epoch}} - 4\kappa_0}{2p^e(\text{view})} \leq m^{r:r'}(\text{view})$. Due to work growth lower bound, it holds that $m^{r:r'}(\text{view}) \leq \frac{1 + \epsilon}{1 - 2\alpha_{\text{max}}\Delta} \cdot \frac{L_{\text{epoch}}}{p^e(\text{view})}$. \square

Let $\text{calibrated_failure}(\text{view}) = 1$ be the following bad event that there exists some epoch e that starts in time t_0 (here we say that an epoch e starts in round t_0 if t_0 is the first round in which an honest node's chain reaches that of epoch e) such that

- $\frac{C_0}{\Delta_{\text{tgt}}} < p^e(\text{view}) \cdot m^{t_0}(\text{view}) < \frac{C}{\Delta_{\text{tgt}}}$, i.e., epoch e starts calibrated;
- however, either view aborted prior to the first honest node enters epoch $e + 1$; or epoch $e + 1$ does not start calibrated.

Lemma C.9. For any $\Pi_{\text{real-compliant}}(\mathcal{A}, \mathcal{Z})$, any sufficiently large κ ,

$$\Pr[\text{view} \leftarrow_{\S} \text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z}) : \text{calibrated_failure}(\text{view}) = 1] \leq \text{negl}(\kappa)$$

Proof. Let $\text{calibrated_failure}^e(\text{view}) = 1$ be defined just like $\text{calibrated_failure}(\text{view})$ but for a fixed epoch e . Due to union bound over the choice of e , it suffices to prove that for any fixed e ,

$$\Pr[\text{view} \leftarrow_{\S} \text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z}) : \text{calibrated_failure}^e(\text{view}) = 1] \leq \text{negl}(\kappa)$$

Without loss of generality, we assume that there are more than $\text{negl}(\kappa)$ fraction of views where $\frac{C_0}{\Delta_{\text{tgt}}} < p^e(\text{view}) \cdot m^{t_0}(\text{view}) < \frac{C}{\Delta_{\text{tgt}}}$ since otherwise the lemma trivially holds. Consider any such view. By Fact C.7, except for $\text{negl}(\kappa)$ fraction of such views, some honest node must enter epoch $e + 1$ at some time $t_1 < t_0 + W$ (and further view does not abort prior to t_1). Henceforth ignore views where this fails to hold.

In view, when $\mathcal{S}^{\mathcal{A}}$ computes the difficulty for the prefix $\text{extract}(\text{chain})[: eL_{\text{epoch}} - \kappa_0]$, let r and r' be defined as in Fact C.8 for chain . Due to Fact C.8 and ignore all views where the bad events related to Fact C.8 take place, it must hold that for any $\epsilon > 0$,

$$(1 - \epsilon) \cdot \frac{L_{\text{epoch}} - 4\kappa_0}{2p^e} \leq m^{r:r'}(\text{view}) \leq \frac{1 + \epsilon}{1 - 2\alpha_{\text{max}}\Delta} \cdot \frac{L_{\text{epoch}}}{p^e}$$

We first show that the next epoch starts out safe. If the case $p^{e+1} := \frac{1}{\gamma}p^e$ is triggered, then due to Fact C.7 bounded change in mining power, and the fact that $\gamma > \chi$, it holds trivially that the next epoch starts safe. Otherwise, due to our difficulty adjustment algorithm, we have that

$$p^{e+1} \leq \frac{\Delta'}{\Delta_{\text{tgt}}} p^e$$

Let $\bar{m} := \frac{m^{r:r'}}{r' - r}$; we have that

$$\bar{m}p^{e+1} \leq \frac{1 + \epsilon}{1 - 2\alpha_{\text{max}}\Delta} \cdot \frac{L_{\text{epoch}}}{p^e(r' - r)} \cdot \frac{\Delta'}{\Delta_{\text{tgt}}} p^e = \frac{1 + \epsilon}{1 - 2\alpha_{\text{max}}\Delta} \cdot \frac{L_{\text{epoch}}}{p^e(r' - r)} \cdot \frac{(r' - r)p^e}{(L_{\text{epoch}} - 2\kappa_0)\Delta_{\text{tgt}}} \leq \frac{4}{\Delta_{\text{tgt}}}$$

where the last inequality holds as long as $L_{\text{epoch}} > 8\kappa_0$, $\nu := 2\alpha_{\text{max}}\Delta < \frac{1}{4}$, and ϵ sufficiently small. Due to bounded change in mining power, we have that $m^{t_1} < \chi\bar{m}$. We have that

$$p^{e+1}(\text{view}) \cdot m^{t_1}(\text{view}) < \frac{4\chi}{\Delta_{\text{tgt}}} < \frac{C}{\Delta_{\text{tgt}}}$$

where the last inequality holds as long as $C > 4\chi$.

We now show that the next epoch starts out not only safe, but also calibrated. The argument is similar as the above. If the case $p^{e+1} := \gamma p^e$ is triggered, it trivially holds that epoch $e + 1$ starts calibrated (since we have already shown that the next epoch must start safe for views where relevant bad events do not happen) — this arises from bounded change in mining power, the fact that $\chi < \gamma$, and due to Fact C.7.

Otherwise, we have that

$$p^{e+1} \geq \frac{\Delta'}{\Delta_{\text{tgt}}} p^e$$

Therefore, let $\bar{m} := \frac{m^{r:r'}}{r'-r}$; we have that

$$\bar{m}p^{e+1} \geq (1 - \epsilon) \cdot \frac{L_{\text{epoch}} - 4\kappa_0}{p^e(r' - r)} \cdot \frac{\Delta'}{\Delta_{\text{tgt}}} p^e = (1 - \epsilon) \cdot \frac{L_{\text{epoch}} - 4\kappa_0}{p^e(r' - r)} \cdot \frac{(r' - r)p^e}{(L_{\text{epoch}} - 2\kappa_0)\Delta_{\text{tgt}}} \geq \frac{1}{1.6\Delta_{\text{tgt}}}$$

Due to bounded change in mining power and Fact C.7, we have that $m^{t_1} \geq \frac{\bar{m}}{\chi}$. We have that

$$p^{e+1}(\text{view}) \cdot m^{t_1}(\text{view}) \geq \frac{1}{1.6\chi\Delta_{\text{tgt}}} \geq \frac{C_0}{\Delta_{\text{tgt}}}$$

Note that the last inequality holds since $\gamma > \chi \geq 1$. □

C.3.2 Analysis of a Safe but Non-Calibrated Start

Let $\text{safe_failure}(\text{view}) = 1$ be the following bad event that there exists some epoch e :

- $p^e(\text{view}) \cdot m^{t_0}(\text{view}) \leq \frac{C_0}{\Delta_{\text{tgt}}}$ where t_0 is the first round in which an honest node enters epoch e ;
- however, view aborted prior to any honest node entering epoch $e + 1$; or let t_1 denote when the first honest node enters epoch $e + 1$, we have that $p^{e+1}(\text{view}) \cdot m^{t_1}(\text{view}) > \frac{C}{\Delta_{\text{tgt}}}$.

Lemma C.10. *For any Π_{real} -compliant $(\mathcal{A}, \mathcal{Z})$, any sufficiently large κ ,*

$$\Pr[\text{view} \leftarrow_{\S} \text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z}) : \text{safe_failure}(\text{view}) = 1] \leq \text{negl}(\kappa)$$

Proof. Let $\text{safe_failure}^e(\text{view}) = 1$ be defined just like $\text{safe_failure}(\text{view})$ but for a fixed epoch e . Due to union bound over the choice of e , it suffices to prove that for any fixed e ,

$$\Pr[\text{view} \leftarrow_{\S} \text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z}) : \text{safe_failure}^e(\text{view}) = 1] \leq \text{negl}(\kappa)$$

Without loss of generality, we assume that there are more than $\text{negl}(\kappa)$ fraction of views where $p^e(\text{view}) \cdot m^{t_0(e, \text{view})}(\text{view}) \leq \frac{C_0}{\Delta_{\text{tgt}}}$ since otherwise the lemma trivially holds. Consider any such view. Let t be the first round such that $p^e(\text{view}) \cdot m^t(\text{view}) > \frac{C_0}{\Delta_{\text{tgt}}}$.

Let $C_1 := \chi \cdot C_0$, notice that due to bounded change in mining power, we have that $p^e(\text{view}) \cdot m^t(\text{view}) < \frac{C_1}{\Delta_{\text{tgt}}}$, i.e., the mining power cannot abruptly jump to C_1 or higher in round t . Without loss of generality, we may assume that for more than $\text{negl}(\kappa)$ fraction of such views, it holds that $t \leq t_1$, since otherwise, the lemma trivially holds. Consider such a view where $p^e(\text{view}) \cdot m^{t_0(e, \text{view})}(\text{view}) \leq \frac{C_0}{\Delta_{\text{tgt}}}$ and $t(e, \text{view}) \leq t_1$ where t is defined as above. Now due to Fact C.7, except for $\text{negl}(\kappa)$ fraction of such views, all honest nodes must reach epoch $e + 1$ at some time $t_1 < t + W$ (and the view must not abort before that). Due to bounded change in mining power, it must hold that for any $r \in [t, t_1]$, $p^e(\text{view}) \cdot m^r(\text{view}) \leq \frac{C_1 \cdot \chi}{\Delta_{\text{tgt}}}$. By our difficulty adjustment algorithm, the difficulty at most becomes easier by γ at time t_1 , i.e., $p^{e+1} \leq \gamma p^e$. Therefore, we have that

$$p^{e+1}(\text{view}) \cdot m^r(\text{view}) \leq \frac{\gamma C_1 \cdot \chi}{\Delta_{\text{tgt}}} \leq \frac{C}{\Delta_{\text{tgt}}}$$

□

Lemma C.11 (No abort due to hyb-compliance-failure). *For any Π_{real} -compliant p.p.t. pair $(\mathcal{A}, \mathcal{Z})$, all but $\text{negl}(\kappa)$ fraction of views from $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$ do not abort due to hyb-compliance-failure.*

Proof. Follows in a straightforward fashion from Lemma C.10 and Lemma C.9. \square

We now show that if the real-world protocol starts out in a safe state, then after a polynomially bounded warmup period, the protocol will enter a calibrate state.

Lemma C.12 (A calibrated epoch starts soon). *For any Π_{real} -compliant p.p.t. pair $(\mathcal{A}, \mathcal{Z})$, for all but $\text{negl}(\kappa)$ fraction of views from $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$, it holds that after some $\text{poly}(\kappa)$ rounds, view enters an epoch that starts calibrated.*

Proof. When \mathcal{S} sets the difficulty of the prefix $\text{chain}[eL_{\text{epoch}} - \kappa_0]$ with \mathcal{F}_{hyb} , let r and r' be defined as in Fact C.8 for chain .

Let $0 < \gamma' < \chi < \gamma$ be a constant. If $r' - r > \gamma'(L_{\text{epoch}} - 2\kappa_0)\Delta_{\text{tgt}}$, then p^{e+1} will reduce by a factor of γ' in comparison with p^e . After polynomially many such epochs in which \mathcal{S} discovers $r' - r > \gamma'(L_{\text{epoch}} - 2\kappa_0)\Delta_{\text{tgt}}$, let e' be the next epoch, either epoch e' starts calibrated or in epoch e' , \mathcal{S} measures that $r' - r \leq \gamma'(L_{\text{epoch}} - 2\kappa_0)\Delta_{\text{tgt}}$. When the latter happens, below we argue that epoch $e' + 1$ will start calibrated. We ignore all views where relevant bad events take place.

Due to work growth upper bound, let $\bar{m} := \frac{m^{r:r'}(\text{view})}{r' - r}$, for any sufficiently small constant $\epsilon > 0$, it must hold that

$$\bar{m}p^e \geq \frac{(1 - \epsilon)(L_{\text{epoch}} - 4\kappa_0)}{\gamma'(L_{\text{epoch}} - 2\kappa_0)\Delta_{\text{tgt}}} \geq \frac{1}{2\gamma'\Delta_{\text{tgt}}} \geq \frac{C_0}{\Delta_{\text{tgt}}}$$

Similarly, we can show that

$$\bar{m}p^e \leq \frac{C}{\Delta_{\text{tgt}}}$$

The remainder the the argument follows in the same manner as that of Lemma C.9, by showing that

$$\bar{m}p^{e+1} \leq \frac{1}{1.6\Delta_{\text{tgt}}}$$

and further, by observing that due to Fact C.7, the epoch must complete prior to $r' + W$; and due to bounded change in mining rate, $m^{t'} \geq \frac{\bar{m}}{\chi}$. \square

C.4 Proofs for Our Main Theorems

We can now show that the view of the environment \mathcal{Z} is indistinguishable in the real-world and the hybrid-world executions. If so, and since all the security properties that we care about are defined over honest nodes' outputs which are observable by \mathcal{Z} , we can immediately conclude that the security properties that hold in the hybrid-world execution must hold in the real world as well.

Lemma C.13 (Indistinguishability of simulated and real-world executions). *For any Π_{real} -compliant p.p.t. pair $(\mathcal{A}, \mathcal{Z})$, the view of \mathcal{Z} in $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$ is computationally indistinguishable from the view of \mathcal{Z} in $\text{EXEC}^{\Pi_{\text{real}}}(\kappa, \mathcal{A}, \mathcal{Z})$.*

Proof. Consider \mathcal{S}' that is the same as \mathcal{S} but does not check collision-failure or hyb-compliance-failure. \mathcal{Z} 's view in $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}'^{\mathcal{A}}, \mathcal{Z})$ and $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$ are computationally indistinguishable due to Fact C.4 and Lemma C.11.

Conditioned on the fact that no **predict-failure** and no **consistency-failure**, it is not hard to see that by construction, the view of \mathcal{Z} in $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}'^{\mathcal{A}}, \mathcal{Z})$ is identically distributed as $\text{EXEC}^{\Pi_{\text{real}}}(\kappa, \mathcal{A}, \mathcal{Z})$. \square

Theorem C.14 (Π_{real} under a safe start). *For any admissible parameters such that*

$$\Gamma_{\text{real}}(\phi, \nu, \chi, p_0, L_{\text{epoch}}, \Delta_{\text{tgt}}, \kappa_0, \gamma) = 1$$

for any constants $\epsilon, \epsilon' > 0$, and any $T_0 > \epsilon' \kappa$, it holds that $\Pi_{\text{real}}(p_0, L_{\text{epoch}}, \Delta_{\text{tgt}}, \kappa_0, \gamma)$ satisfies the following properties against any p.p.t. $(\mathcal{A}, \mathcal{Z})$ that is Π_{real} -compliant w.r.t. these parameters:

- $(T_0, g_0, g_1, t_{\text{warm}})$ -chain growth, where $g_0 := \frac{1}{3\chi^2\Delta_{\text{tgt}}}$, $g_1 := \frac{7\chi^2}{\Delta_{\text{tgt}}}$, and t_{warm} is some polynomially bounded function in κ .
- (T_0, μ) -chain quality where $\mu := 1 - \frac{1+\epsilon}{1+\phi}$;
- T_0 -consistency.

Proof. Consistency follows directly from Lemma C.13 and the consistency of Π_{hyb} .

For chain quality, due to the union bound, it suffices to show chain quality for $T_0 = \epsilon' \kappa$ and for any sufficiently small constant ϵ' such that T_0 consecutive blocks span at most 2 epochs. If the T_0 consecutive blocks of interest is within a single epoch, chain quality follows directly from Lemma C.13 and the work quality of Π_{hyb} . If the T_0 consecutive blocks of interest fall into two epochs — let $\epsilon_1 > 0$ be a sufficiently small constant. If both epochs contain more than $\epsilon_1 \kappa$ blocks, then by chain quality for a single epoch, both segments have good chain quality, and the chain quality of the two segments combined must be good too. If one of the epoch contains less than $\epsilon_1 \kappa$ blocks, the segment in the other epoch satisfies $\mu^* = 1 - \frac{1+\epsilon_2}{1+\phi}$ chain quality for any constant ϵ_2 . The overall chain quality over the two epochs must be lower bounded by $\mu = \frac{\mu^*(T_0 - \epsilon_1 \kappa)}{T_0}$. For any positive constants ϵ, ϵ' , for sufficiently small constants ϵ_1, ϵ_2 , we have that $\mu \geq 1 - \frac{1+\epsilon}{1+\phi}$.

For chain growth, due to Lemma C.12, for any Π_{real} -compliant $(\mathcal{A}, \mathcal{Z})$, all but $\text{negl}(\kappa)$ fraction of views from $\text{EXEC}^{\Pi_{\text{hyb}}}(\kappa, \mathcal{S}^{\mathcal{A}}, \mathcal{Z})$ enter a calibrated epoch after $\text{poly}(\kappa)$ rounds. Due to Lemma C.9, for all but $\text{negl}(\kappa)$ fraction of views, once the view enters a calibrated epoch, all future epochs must be calibrated. Therefore, it is not hard to see that once a calibrated epoch e starts in any such view where the relevant bad events do not happen, it must hold that every epoch $e' \geq e$ completes within W time due to Fact C.7, and due to bounded change in mining power, for each epoch $e' \geq e$, the honest mining rate at any time r within the epoch must be bounded from both sides: $\frac{C_0}{\chi \Delta_{\text{tgt}}} \leq p^{e'}(\text{view}) \cdot m^r(\text{view}) \leq \frac{C\chi}{\Delta_{\text{tgt}}}$. The remainder of the proof follows from the work growth of Π_{hyb} . \square

We say that $(\mathcal{A}, \mathcal{Z})$ respects calibrated start if for every view of non-zero support of $\text{EXEC}^{\Pi_{\text{real}}}(\kappa, \mathcal{A}, \mathcal{Z})$ the first epoch starts calibrated. For any such Π_{real} -compliant p.p.t. pair $(\mathcal{A}, \mathcal{Z})$ that additionally respects calibrated start, we can achieve the same security properties but without needing a warmup period for the chain growth rates to stabilize, i.e., $t_{\text{warm}} := 0$. This is formally stated in the following corollary.

Corollary C.15 (Π_{real} under a calibrated start). *For any admissible parameters such that*

$$\Gamma_{\text{real}}(\phi, \nu, \chi, p_0, L_{\text{epoch}}, \Delta_{\text{tgt}}, \kappa_0, \gamma) = 1$$

for any constant $\epsilon, \epsilon' > 0$, and any $T_0 > \epsilon' \kappa$, it holds that $\Pi_{\text{real}}(p_0, L_{\text{epoch}}, \Delta_{\text{tgt}}, \kappa_0, \gamma)$ satisfies the following properties against any p.p.t. $(\mathcal{A}, \mathcal{Z})$ that is Π_{real} -compliant w.r.t. these parameters and additionally respects calibrated start:

- $(T_0, g_0, g_1, 0)$ -chain growth, where $g_0 := \frac{1}{3\chi^2 \Delta_{\text{tgt}}}$, and $g_1 := \frac{7\chi^2}{\Delta_{\text{tgt}}}$.
- (T_0, μ) -chain quality where $\mu := 1 - \frac{1+\epsilon}{1+\phi}$;
- T_0 -consistency.

Proof. The proof follows in the same manner as that of Theorem 5.1, but since the first epoch starts calibrated, we have that $t_{\text{warmup}} = 0$. \square

D Preliminary: Negative Association

Lemma D.1 (Negative Association for Competing Random Variables). *Let $\mathcal{X} := \{X_i : i \in [n]\}$ be a collection of non-negative random variables such that with probability 1, at most one of the X_i 's is non-zero. Then, the collection \mathcal{X} of variables are negatively associated.*

Proof. As remarked in [DR98], the proof idea is due to Colin McDiarmid. Suppose I and J are disjoint subsets of $[n]$. Moreover, $f(a_i : i \in I)$ and $g(a_j : j \in J)$ are either both non-increasing or both non-decreasing functions on non-negative reals for each coordinate. We give the case (as used later in our proof) that both f and g are non-increasing. Then, it suffices to prove that

$$\mathbf{E}[f(X_i : i \in I) \cdot g(X_j : j \in J)] \leq \mathbf{E}[f(X_i : i \in I)] \cdot \mathbf{E}[g(X_j : j \in J)].$$

Define non-negative functions $\widehat{f}(a_i : i \in I) := f(\mathbf{0}) - f(a_i : i \in I)$ and $\widehat{g}(a_j : j \in I) := g(\mathbf{0}) - g(a_j : j \in I)$, where each coordinate is non-negative.

Observe that with probability 1, at most one X_i 's is non-zero. This implies that $\widehat{f}(X_i : i \in I) \cdot \widehat{g}(X_j : j \in I)$ is 0 with probability 1.

Therefore, we have $\mathbf{E}[\widehat{f}(X_i : i \in I) \cdot \widehat{g}(X_j : j \in I)] = 0 \leq \mathbf{E}[\widehat{f}(X_i : i \in I)] \cdot \mathbf{E}[\widehat{g}(X_j : j \in I)]$, which after rearranging gives the required result. \square