Intergalactic Consensus: Asynchronous Bitcoin Clients

Odysseas Sofikitis[1,2] and Dionysis Zindros[2]

[1] National Technical University of Athens
[2] Common Prefix
{odysseas,dionyziz}@commonprefix.com

Abstract. Bitcoin is a *synchronous* proof of work consensus protocol. Its security against a minority adversary rests on the synchrony among the validators, and that assumption is known to be necessary. Bitcoin clients have inherited it: every client construction to date assumes that messages arrive on time and that at least one of its peers is honest. We show that a client needs neither. We keep the validators synchronous and give the client an asynchronous connection to them, over which the adversary may delay or drop any message and may herself answer. The client holds a clock and nothing else. Bitcoin confirms a block once it is k blocks deep in the chain, for a constant k. Our client confirms a block once it is deep enough for the current round and for the height of the chain it received. We prove the rule safe and live in the bitcoin backbone model: the client confirms a prefix of every honest chain, and the confirmed prefix grows at a constant rate. Asynchrony costs latency. A client that receives an honest chain at round r discards a ρ^2 fraction of it, where ρ bounds the mining power of the adversary relative to the honest validators. It discards $\Theta(r)$ blocks where a synchronous client discards a constant, and it confirms a block $\Theta(r)$ rounds after the honest validators do. We prove this tight: against a private mining adversary, every safe rule that reads only the round and the height discards asymptotically as much, and no constant depth is safe.

Keywords: Bitcoin · Proof of work · Longest chain · Asynchrony · Clients

1 Introduction

The year is 2227. Humans have colonized α Centauri.[3] All 21 million bitcoin have been mined. A lone hodler on α Centauri finally decides to sell her bitcoin.

While all validators are still based on earth, where network latency is known and bounded, intergalactic communication is still tricky; our hodler does not trust the dissemination of information from so far away. It could take years for an actual honest message to reach her, or the message might be dropped altogether. How is she going to use bitcoin without network synchrony? Lucky for

[3] Although α Centauri is in our own galaxy, our construction works across galaxies as well.

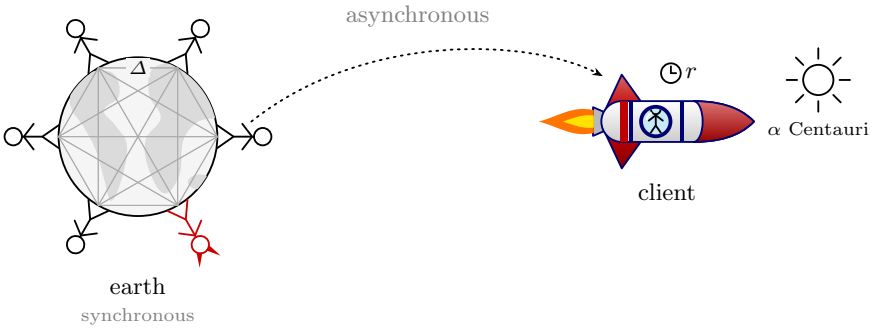

Fig. 1. The network model. The validators are on earth, and they are synchronous with one another: a message from one of them reaches all the others within a bounded delay Δ that everybody knows. One of them, in red, is adversarial. The client is on its way to α Centauri, and its communication with earth is asynchronous: a chain sent to it may take any amount of time to arrive, or may never arrive, and the client cannot tell which. What the client does have is a clock, so it knows the current round r, and the parameters of the protocol, which are public.

her, we come to her rescue! With our new protocol, she can still confirm transactions without assuming any bounded delay over her communication channel with earth.

The necessity of synchrony is well understood, both for proof of work consensus and for dynamic availability in general [7, 9, 12], and the clients of a longest chain protocol have always inherited the assumption from the validators. Previous work on longest chain clients has focused on optimizing the communication load of the messages downloaded by clients [1, 3, 5], and has left the assumption untouched; no result has shown that synchrony is in fact not necessary for the security of clients. Our construction is safe even in asynchrony, while preserving the benefits of dynamic availability. We highlight that our model differs from the usual assumption of sleepy clients, where the client knows when it was asleep, gets everything it missed when it wakes, and assumes synchrony while it is awake.

We frame this in a hybrid network model. On the one hand we model the network of validators as synchronous, with a bounded network delay known to everybody. On the other hand we allow the clients' connections to be asynchronous. This network model is the most realistic today, and we explore it in this work for the first time. Validators are specific machines, optimized to stay well connected. Clients are everyday devices: wallets on airplanes, laptops behind a router under someone else's control, or any device that may have a poor connection or is under a denial-of-service attack. In this model the clients can still confirm transactions, even if their communication with the rest of the network is unstable. We put forth a client construction that allows for this, and prove it safe and live. Our construction is backwards compatible with the existing bitcoin construction.

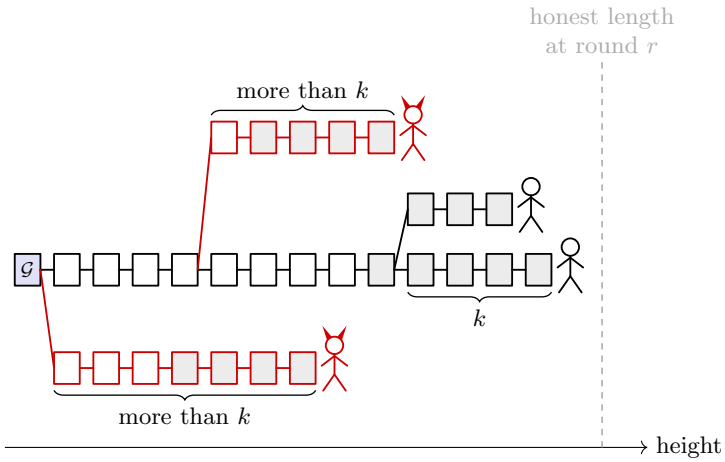

Fig. 2. The block tree at round r, drawn by height. The dashed line is the height that the client estimates, from its clock, for the chains of the honest validators. Two honest validators hold chains of different lengths that end next to it and share every block up to a short fork. The last k blocks of each are shaded. The adversary, in red, holds two much shorter chains. Each leaves the chain of the honest validators more than k blocks below its tip. Their last k blocks are shaded too, and the rest of each lies off the chain of the honest validators.

Construction overview. When a client wakes up, it asks the network for a chain and waits until it receives one. It has access to a trusty clock and knows the round r when it received the chain $\mathcal{C}$. From the round alone, the client can estimate how long the chains of the honest validators are. From the received chain, it prunes a suffix in order to compute a confirmed prefix of $\mathcal{C}$. Chains close to that length share a long common prefix, and pruning k blocks from their tip leaves a stable chain, as in bitcoin. The adversary, however, can hand the client a shorter chain, which forked from the honest chain much earlier than k blocks before its tip (Figure 2). In contrast to the confirmation rule under a synchronous network, our client therefore does not prune a constant number of blocks, but computes a function $\gamma(r, \alpha)$ of the round r and the height α of the chain $\mathcal{C}$ that arrived. The shorter the chain is than what is expected for the current round, the more blocks the client prunes. If the chain is much shorter than expected for the current round, the client discards all of it. Otherwise, it keeps $\mathcal{C}[:-\gamma(r, \alpha)]$. A client that stays connected to the network and receives honest chains round after round will confirm a prefix that grows linearly with the round number. However, the pruning function $\gamma(r, \alpha)$ also grows linearly with the round. This gives our construction a constant throughput, but a deteriorating liveness.

Contributions. We work in the static synchronous backbone model of [4], while assuming clients have an asynchronous connection to the network and carry an accurate clock.

1. We give a confirmation rule that a client evaluates given only a chain and the current round of the execution (Section 3).
2. We prove that what it returns is a prefix of every chain an honest validator adopts from that round onwards, and that any two clients running it return prefixes of one another (Section 4.1).
3. We prove it live, and we show the confirmed prefix grows at a constant rate (Section 4.2).
4. We prove the pruning depth tight by exhibiting a private mining adversary that attains it, and we show that no constant confirmation depth is safe (Section 5).

Section 2 fixes the model, Section 3 gives the construction, Section 4 proves it safe and live, Section 5 proves it tight, and Section 6 collects the remarks.

Related work. No protocol stays live under dynamic participation and safe under network partitions, which is the availability-finality dilemma [7], and the same tension holds between availability and accountable safety [8]. We keep synchrony for the validators, where these results need it, and drop it only for the clients. The client model itself is made explicit in [12], which separates clients that sleep from clients that are always on, and clients that only listen from clients that also speak. Longest chain consensus reaches its optimal resilience there for sleepy clients that only listen. A sleepy client knows when it slept and assumes synchrony while it is awake. Our client model is strictly stronger, assuming no delay bound and keeping only the clock. Lewis-Pye and Roughgarden [6] observe that, if bitcoin with a known total mining power could be implemented in a timed setting, a chain alone would prove that its blocks are confirmed, and they leave this open. We make it concrete: our client takes the time from its own clock. Previous work on longest chain clients was aimed at optimizing the amount of data downloaded [1, 3, 5], but still assumes synchrony.

2 Model

We work in the static synchronous setting of [4] and we keep some of its notation. This section fixes the notation and states the assumptions. The parts of the model that only the proofs use are in Section A.

Validators. There are n parties called validators. Up to t may be Byzantine and arbitrarily deviate from the protocol.

Rounds. Time proceeds in rounds. Rounds are numbered $1, 2, \ldots$. For $r \leq r'$, we write $[r, r']$ for the set of rounds $\{r, \ldots, r'\}$, and $[r, r'] = \emptyset$ when $r' < r$.

Network. We assume a synchronous network among the validators. A message sent by an honest validator at round r is received by every other validator at the beginning of round $r + 1$. So the network delay among the validators is one round, $\Delta = 1$. The adversary can see the messages of honest parties at any

point of a round. At the end of each round she fixes the messages delivered in the next round and their order. She may reorder messages, spoof their source or add messages of her own, but she cannot drop a message.

Adversary. We assume a *Byzantine, rushing* adversary, who may corrupt up to t parties. A rushing adversary sees every message sent by an honest validator before it is delivered to any other party.

q-bounded random oracle. We assume access to a random oracle $H\colon \{0,1\}^* \to \{0,1\}^\kappa$, where κ is the security parameter. Each validator is allowed q queries to the random oracle H per round. The adversary is allowed tq queries. We assume that she makes all of them in every round. This loses no generality: an adversary that makes fewer queries can make the rest and ignore their answers. A query is successful if its output is below the target T. A single query is successful with probability $p = T/2^\kappa$. A successful query made at round r yields a block. We say that this block was *computed at round r*.

Protocol. Validators run the bitcoin backbone protocol of [4]. Every validator keeps a local chain, initialized with the genesis block $\mathcal{G}$. At round r a validator runs `maxvalid` on its local chain and on the chains it received at the beginning of round r. Function `maxvalid` returns the longest valid chain among its arguments and breaks ties towards the local chain. The validator then makes its q queries. On success it appends the new block to the chain returned by `maxvalid`. The resulting chain becomes its local chain and is sent to all validators. Following [4], we say that a validator *has a chain at round r* if this was the local chain it started the round with. The validator *adopts a chain at round r* if this was the return value of `maxvalid` at round r. The chain a validator adopts at round r is at least as long as the chain it has at round r. The chain it has at round $r+1$ is the chain it adopted at round r, extended by at most one block if the validator was successful at round r. Blocks carry transactions. As in the ledger protocol of [4, Section 6], an honest validator puts into the blocks it computes the transactions it is given as input. We call a transaction *honest* in the sense of [4, Definition 6.1].

Clients. Clients are parties that do not participate in the consensus protocol. A client is only interested in confirming a ledger that is derived from the chain the validators maintain. This ledger must be *safe*. A client's connection to the rest of the network is asynchronous; the adversary can delay messages for any arbitrary amount of time. Thus, the client cannot assume that it is going to receive a message from an honest party within a certain number of rounds. We say that a client *receives a chain at round r* when the adversary delivers that chain to it during round r.

Definition 1 (Stale chain). *Let $D \geq 0$ be an integer and let r be a round. A chain that a client receives at round r is* D-stale *if an honest validator adopted it at round $r - D$.*

Clients have access to a wall clock. They know the current round r, the genesis block $\mathcal{G}$, and the public parameters f, ρ, ϵ and λ that we introduce below. Like every party of [4], a client knows the target T and makes unlimited verification queries to H, so it can check that a chain is valid, as `maxvalid` does.

The ledger of a chain is the sequence of transactions in its blocks. For a client P and a round r, we write $\mathcal{L}_r^P$ for the ledger of the chain P outputs at round r. It is empty while P outputs $\perp$. We write $\mathcal{L} \preceq \mathcal{L}'$ when the ledger $\mathcal{L}$ is a prefix of the ledger $\mathcal{L}'$. A client must keep its ledger safe and live, in the following sense.

Definition 2 (Ledger safety). *Clients are* safe *if, for any two clients P_1 and P_2, not necessarily distinct, and any two rounds r_1 and r_2, $\mathcal{L}_{r_1}^{P_1} \preceq \mathcal{L}_{r_2}^{P_2}$ or $\mathcal{L}_{r_2}^{P_2} \preceq \mathcal{L}_{r_1}^{P_1}$.*

Definition 3 (Ledger liveness). *Clients are* live *if, for every round r, there exists a number of rounds u such that the following holds for every honest transaction tx. If tx is given as input to every honest validator at every round from r on, then, for every client P and every round $r' \geq r + u$, $\mathsf{tx} \in \mathcal{L}_{r'}^P$.*

Chains and the block tree. A chain is a sequence of blocks, each of which contains the hash of its predecessor. Every chain begins with the genesis block $\mathcal{G}$. We write $\mathcal{C}[i]$ for the i-th block of $\mathcal{C}$, counting from $\mathcal{C}[0] = \mathcal{G}$, and we say that $\mathcal{C}[i]$ is at height i. We write $\mathcal{C}[i{:}j]$ for the subsequence of $\mathcal{C}$ at heights $i, \ldots, j-1$. We omit i when the range starts at the beginning of $\mathcal{C}$, and j when it extends to the end. Negative indices count from the end. So $\mathcal{C}[-1]$ is the last block of $\mathcal{C}$, its *tip*, and $\mathcal{C}[:-k]$ is $\mathcal{C}$ without its k last blocks. When $k \geq |\mathcal{C}|$, $\mathcal{C}[:-k] = []$. For a real number x we write $\mathcal{C}[:-x]$ for $\mathcal{C}[:-\lceil x \rceil]$. We write $|\mathcal{C}|$ to denote the length of $\mathcal{C}$, which is the number of blocks of $\mathcal{C}$ including the genesis block. The height of $\mathcal{C}$ is the height of its tip, $|\mathcal{C}| - 1$. We write $\mathcal{C}_1 \preceq \mathcal{C}_2$ when $\mathcal{C}_1$ is a prefix of $\mathcal{C}_2$. That is, $\mathcal{C}_1 \preceq \mathcal{C}_2 \iff \mathcal{C}_2[:|\mathcal{C}_1|] = \mathcal{C}_1$. Every validator has a chain of length 1 at round 1. A block is honest if it was computed by an honest validator and adversarial otherwise. We regard the genesis block as an honest block computed at round 0.

Definition 4 (Block tree). *For a round r, the block tree at round r, denoted $\mathcal{T}_r$, is the set of all blocks computed in a round of $[0, r]$. Its edges are the hash references between these blocks. A block computed by the adversary belongs to $\mathcal{T}_r$ whether or not it was sent. A chain $\mathcal{C}$ is in $\mathcal{T}_r$ if all of its blocks are in $\mathcal{T}_r$.*

In a typical execution, defined in Definition 7, no copies occur, so every block of $\mathcal{T}_r$ has a unique height and $\mathcal{T}_r$ is a tree.

Definition 5 (Common prefix of two chains). *For chains $\mathcal{C}_1$ and $\mathcal{C}_2$, let $\mathcal{C}_1 \cap \mathcal{C}_2$ be the longest chain that is a prefix of both.*

Parameters. A round is *successful* if some honest validator computes a block in it. We write f for the probability that a round is successful. It is public: it is the rate at which the honest validators extend their chain. We write f_2 for the probability that two or more honest validators compute a block in the

same round. It depends on the number of honest validators, so the client uses instead the bound $\bar{f}_2 = f^2/(2(1-f)^2)$, which is at least f_2 (Lemma 3) and small compared with f. Nobody knows n or t. What is public is the bound ρ of the next assumption on their ratio, $t/(n-t) \leq \rho$. It bounds the mining power of the adversary relative to that of the honest validators, and $1+\rho$ bounds the mining power of all validators relative to the honest ones.

Assumption 1 (Honest majority). *There exist* $\epsilon \in (0,1]$ *and* $\rho \in [0,1)$ *such that* $t \leq \rho(n-t)$ *and* $3f + 3\epsilon < 1 - \rho$.

Assumption 1 is the honest majority assumption of [4].

Typical executions. Let $\lambda \geq 2/f$ be an integer. All our statements are about (ϵ, λ)-typical executions, as defined in [4] (Definition 7). Over any window of λ or more rounds, every count of blocks in such an execution stays close to its expectation, within a margin set by ϵ. Moreover, the hash function behaves like a random function: nobody predicts a block, produces two of the same, or inserts one into a chain it does not extend. An execution is typical except with a probability that is negligible in κ for $\lambda = \Omega(\log^2 \kappa)$ (Theorem 5).

3 Construction

We now give the construction that allows the client to confirm transactions. Let the client receive a chain $\mathcal{C}$ of height α at round r, and let $\mathcal{C}'$ be any chain that an honest validator adopts at round r. The goal is to output a prefix of $\mathcal{C}'$. We do this by bounding the number of blocks of $\mathcal{C}$ that are not on $\mathcal{C}'$, and pruning that many blocks from the end of $\mathcal{C}$. What remains is then a prefix of $\mathcal{C}'$. We compute this bound solely with the public parameters of the protocol, the clock the client has and the height of the chain it received. A constant bound, as in the standard bitcoin construction, is not safe for the client (Corollary 4), and the bound has to grow with the round.

We follow the idea of Figure 3, which rests on the different chain growth rates. Every validator makes a fixed number of attempts per round, and each attempt finds a block with a fixed probability. In the terms of Section 2, the chain of the honest validators grows by about f blocks per round, the adversary mines at most about ρf blocks per round and no chain grows faster than about $(1+\rho)f$ blocks per round. The honest chain grows at least that fast because honest validators build on the longest chain they see, and no chain grows faster because every block is somebody's work. The client knows these rates before it sees a single block, and Figure 3 draws them as three lines. The band around each line is how far a typical execution can stray from it, and Lemma 1 makes this exact.

Simple rule. Now look at the chain $\mathcal{C}$ the client received at round r. It has two parts. The first is a prefix that the honest validators also hold. The second is a tail that they do not hold: the adversary mined it in private, after she left the

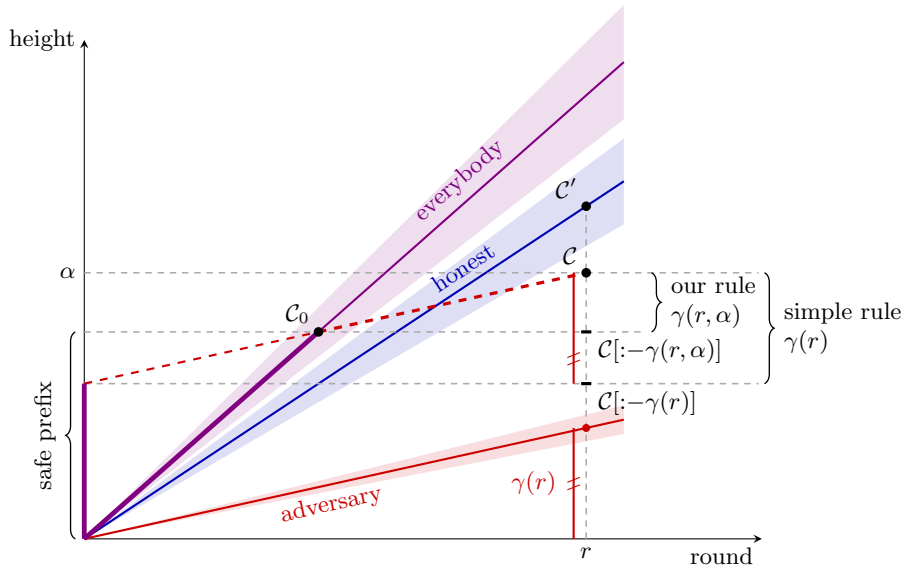

Fig. 3. The pruning depth, drawn for $\rho = \frac{1}{3}$ as height against round. The three lines are the mining rates of everybody together (violet), of the honest validators (blue) and of the adversary alone (red). The band around each line is how far a typical execution can stray from it. At round r the client receives the chain $\mathcal{C}$ of height α, while an honest validator holds the longer chain $\mathcal{C}'$. In the worst case, $\mathcal{C}$ grows at the top rate up to its prefix $\mathcal{C}_0$, and the adversary mines the rest in private, along the dashed red line. Our rule prunes $\gamma(r,\alpha)$ blocks, which takes $\mathcal{C}$ down to the height of $\mathcal{C}_0$. The simple rule assumes instead that the prefix was already there at round 0, on the height axis, and that the adversary mined the rest over all r rounds, along the dashed red line extended to that axis. So it prunes $\gamma(r)$ blocks, the height of the red line at round r.

honest chain. The client wants to keep the prefix and prune the tail. It cannot see where one ends and the other begins, but it can bound the length of the tail. Based on the adversarial chain growth, we can make a simple observation: in the worst case the adversary has been mining in private for all r rounds and has produced at most about $\rho f r$ blocks. That means a client that prunes $\rho f r$ blocks from any chain received at round r should be safe. Figure 3 shows what this simple rule prunes.

This is indeed true, but the rule prunes much more than it has to. A chain as tall as an honest chain has height about fr, and the rule prunes a ρ fraction of it. For $\rho = \frac{1}{2}$, that is half the chain.

Pruning depth. Instead of assuming that the adversary has been mining a fork since genesis, we can compute with much better precision the number of blocks the adversary could have mined if we take into account the height α of the received chain. If $\alpha > \rho f r$, the adversary cannot have mined the whole chain by herself, and the chain is the result of two different mining processes. Up to a

fork point, honest and adversarial blocks extend it together, at a rate of about $(1+\rho)f$. After the fork point, only the adversary extends it, at a rate of about ρf. This gives us a simple intuition: if a chain is long enough, it must have forked late, and a late fork point leaves the adversary few rounds to mine above it. If the chain is not long enough, it must have forked early, and an early fork point leaves the adversary many rounds to mine above it. Figure 3 draws the worst case.

We now put numbers to this. Let r^* be the round in which the fork point was mined, and let d be the number of blocks of $\mathcal{C}$ above it, so that the fork point is at height $\alpha - d$. By round r^* no chain is taller than about $(1+\rho)fr^*$, so $\alpha - d \lesssim (1+\rho)fr^*$. The d blocks above the fork point were mined after round r^*. From round r^* to round r the adversary is mining in private with rate ρf. Strictly, a few of the d blocks may be honest ones, left on a branch that the honest validators abandoned, after a natural fork. Lemma 13 counts them, but they change the bound only by a small term, so for now we have $d \lesssim \rho f(r - r^*)$. Each inequality bounds a duration. By the first, the chain up to the fork point took at least $(\alpha - d)/((1+\rho)f)$ rounds to mine. By the second, the d blocks above it took at least $d/(\rho f)$ more. Both happened within the first r rounds, so the two durations add up to at most r. Multiplying by $\rho(1+\rho)f$ and rearranging gives

$$d \lesssim \rho\left((1+\rho)fr - \alpha\right). \tag{1}$$

The client prunes this many blocks. The bound is the length of the tail in the worst case of Figure 3, where the chain grows as fast as possible up to the fork point and only as fast as the adversary after it. The taller the chain, the shorter this tail. The tallest chain that can exist, of height about $(1+\rho)fr$, has no room for a tail at all. A chain as tall as an honest chain, of height about fr, can have a tail of about $\rho^2 fr$ blocks, a ρ^2 fraction of its height. For $\rho = \frac{1}{2}$ that is a quarter of the chain, instead of the half that the simple rule prunes. A chain of height ρfr or less could have been mined by the adversary alone, so it may be all tail, and the client prunes all of it.

Client. Algorithm 1 gives the client. It keeps asking the network for chains, and it keeps one chain as its output, which is $\perp$ until it confirms something (Line 1). Every time it receives a valid chain $\mathcal{C}$, at some round r, it computes the pruning depth $\gamma(r, |\mathcal{C}| - 1)$ from the round and the height of $\mathcal{C}$ (Line 3). It prunes that many blocks from the end of $\mathcal{C}$ (Line 4). If what remains has more than one block and is longer than the current output, it becomes the new output (Lines 5 and 6). At every round the client outputs the chain it keeps.

The pruning depth is Equation (1) with every approximate rate replaced by a bound that holds in every typical execution. Let $A = 1 + \rho + 2\epsilon + \frac{\rho f}{1-f}$, $\rho_1 = \frac{\rho}{1-f} + \frac{\bar{f}_2}{f} + 2\epsilon$ and $k_0 = 2(\lambda+2)f$. For a chain of height α received at round r, the pruning depth is

$$\gamma(r, \alpha) = \max\left\{k_0,\ \frac{\rho_1\left[Af(r+\lambda) - \alpha\right] + A(2f+1)}{1 - \bar{f}_2/f}\right\}. \tag{2}$$

The next lemma bounds the height of every chain from above, at the rate Af, and the height of every honest chain from below, at the rate $(1-\epsilon)f$.

Lemma 1 (Growth bounds). *In a typical execution the following hold for every round $r \geq 0$.*

1. *Every chain $\mathcal{C} \in \mathcal{T}_r$ satisfies $|\mathcal{C}| - 1 < Af \max(r, \lambda)$.*
2. *If $r > \lambda$, every chain $\mathcal{C}$ adopted by an honest validator at round r satisfies $|\mathcal{C}| - 1 > (1-\epsilon)f(r-1)$.*

The intuition behind this rule is the following: no chain grows faster than Af blocks per round (Lemma 1), so Af takes the place of $(1+\rho)f$ in the bound on the fork point. A chain gains at most $\rho_1 f$ blocks per round above its fork point with an honest chain (Lemma 13 and Definition 7), so $\rho_1 f$ takes the place of ρf in the bound on d. The term $\bar{f}_2/f$ accounts for natural forks that may happen between honest parties. The constants A and ρ_1 exceed $1+\rho$ and ρ only by terms of order f and ϵ, and tend to them as f and ϵ tend to 0. With these rates, the two bounds of the derivation of Equation (1) become $\alpha - d < Af(r^* + \lambda)$ and $d < \rho_1 f(r - r^*) + 2f + 1$. The term λ is there because the bounds of Definition 7 hold only over windows of at least λ rounds. So Lemma 1 bounds the height of the fork point by $Af \max(r^*, \lambda)$, which is at most $Af(r^* + \lambda)$. The term $2f + 1$ accounts for the blocks computed in the rounds r^* and r, at the two ends of the fork. The same steps as for Equation (1) then give Equation (2). Its denominator is $A - \rho_1$, which equals $1 - \bar{f}_2/f$. It is positive, since Assumption 1 gives $f < 1/3$, and then $\bar{f}_2/f = f/(2(1-f)^2) < 1$.

The floor k_0 is of the order of the common prefix parameter of [4]. It separates short forks from long ones. A fork with fewer than k_0 blocks above its fork point is pruned by the floor itself. The blocks of a fork with k_0 blocks or more above its fork point were computed in more than λ consecutive rounds (Lemma 9). This is long enough for the bounds of Definition 7 to apply to the fork.

Algorithm 1 The client. It outputs $\mathcal{C}_{\text{out}}$ at every round.

$\mathcal{C}_{\text{out}} \leftarrow \bot$
for every valid chain $\mathcal{C}$ received, at the round r of its receipt **do**
  $k \leftarrow \gamma(r, |\mathcal{C}| - 1)$ ▷ Equation (2)
  $\mathcal{C} \leftarrow \mathcal{C}[:-k]$
  if $|\mathcal{C}| > 1$ **and** ($\mathcal{C}_{\text{out}} = \bot$ **or** $|\mathcal{C}| > |\mathcal{C}_{\text{out}}|$) **then**
    $\mathcal{C}_{\text{out}} \leftarrow \mathcal{C}$
  end if
end for

Section 4 shows that this client is safe and live, and Section 5 that it prunes no more than it has to.

4 Security

We now show that the client is safe and live. Throughout, the execution is typical, Assumption 1 holds, and $\lambda \geq \frac{(2f+1)(1-\epsilon)}{2f(1-\epsilon-\rho_1)} - 2$, a bound of the same order as $\lambda \geq 2/f$ that the proof of safety needs. The proofs are in Section B.

The proofs make the argument of Section 3 exact. They count the blocks of a received chain above its fork point with an honest chain, and charge each of them to an adversarial block, or to a round in which two or more honest validators compute a block. For the round of receipt, this bounds the number of blocks of the received chain that are on a private fork, off the honest chain. Later rounds split into two cases. If the received chain is shorter than some honest chain, what the client keeps lies at least k_0 blocks deep in that honest chain, and the common prefix property of [4] keeps it there. If the received chain is at least as long as every honest chain, the client is in the position of a validator that receives the longest chain in the network, and pruning k_0 blocks is enough, as it is for the validator.

4.1 Safety

Theorem 1 (Safety). *In a typical execution, let a client running Algorithm 1 receive a valid chain $\mathcal{C}$ of height α at round r, and compute from it the chain $\mathcal{C}[:-\gamma(r,\alpha)]$. Then, for every round $r' \geq r$ and every chain $\mathcal{C}'$ adopted by an honest validator at round r', $\mathcal{C}[:-\gamma(r,\alpha)] \preceq \mathcal{C}'$.*

Safety is a property of each chain the client computes, so it extends at once to any number of clients.

Corollary 1 (Consistency). *In a typical execution, let two clients running Algorithm 1, not necessarily distinct, receive chains at rounds r_1 and r_2, and let $\mathcal{C}_1$ and $\mathcal{C}_2$ be the chains they compute from them. Then $\mathcal{C}_1 \preceq \mathcal{C}_2$ or $\mathcal{C}_2 \preceq \mathcal{C}_1$. In particular, the output of a client is well defined, it never shrinks, and it is a prefix of every chain adopted by an honest validator at every round from the receipt that produced it onwards.*

The two clients never communicate, and either one may receive every chain from the adversary. A client that hears only the adversary may confirm nothing, but it never confirms a block that is off the honest chain.

The ledger of a client is read off the chain it outputs, and ledgers of chains that are prefixes of one another are prefixes of one another.

Corollary 2 (Ledger safety). *In a typical execution, clients running Algorithm 1 are safe.*

4.2 Liveness

Liveness needs a chain to arrive. The adversary controls the link of the client. If she never delivers an honest chain, the client confirms nothing. When an honest

chain does arrive, even a stale one, the client confirms every transaction that the honest validators have had long enough. The next theorem says how long is long enough. For a round r and an integer $D \geq 0$, let

$$u(r, D) = \frac{(1 - \rho_1/A)\, r + D + 13\lambda}{1 - \epsilon - \rho_1} - r. \tag{3}$$

Theorem 2 (Liveness). *In a typical execution, let r be a round and $D \geq 0$ an integer, and let an honest transaction be given as input to every honest validator at every round from r on. Every client running Algorithm 1 that receives a D-stale chain at a round $r' \geq r + u(r, D)$ outputs a chain that contains the transaction at every round from r' on.*

Liveness in the sense of Definition 3 asks every client to confirm the transaction, so it needs every client to receive chains. If each receives a D-stale chain at every round, it follows from Theorem 2.

Corollary 3 (Ledger liveness). *In a typical execution, let $D \geq 0$ be an integer and let every client receive a D-stale chain at every round after round D. Then clients running Algorithm 1 are live: for every round r, the number of rounds $u(r, D)$ of Equation* (3) *satisfies Definition 3.*

The client prunes only a ρ^2 fraction of an honest chain. A chain adopted by an honest validator at round r has height at least $(1 - \epsilon)f(r - 1)$ (Lemma 1), and the pruning depth falls as the height grows, so from such a chain the client prunes at most $\gamma(r, (1 - \epsilon)f(r - 1))$ blocks. By Equation (2), this is about $\rho^2 fr$ for large r and small f and ϵ. For small f and ϵ, ρ_1/A is about $\rho/(1 + \rho)$ and $1 - \epsilon - \rho_1$ is about $1 - \rho$, so $u(r, D)$ is about $\frac{\rho^2}{1-\rho^2} r + \frac{D+13\lambda}{1-\rho}$. A transaction given to the honest validators at round r is confirmed at most about $\frac{\rho^2}{1-\rho^2} r$ rounds later, and every round by which the chain the client receives is stale costs at most $\frac{1}{1-\rho}$ more. In bitcoin this delay is a constant. For the client it grows with the round, and Section 5 shows that every safe rule that reads only the round and the height prunes about as much.

Theorem 2 also shows that the confirmed prefix grows at a constant rate. The condition $r' \geq r + u(r, D)$ holds if and only if $r \leq \big((1 - \epsilon - \rho_1)r' - D - 13\lambda\big)/(1 - \rho_1/A)$. So a client that receives a D-stale chain at round r' outputs every honest transaction that the honest validators have been given at every round from such a round r on. This bound on r grows linearly with r', at a rate of about $1 - \rho^2$ for small f and ϵ.

5 Tightness

In this section we show that the pruning depth of Algorithm 1 is optimal, up to small terms, among rules that read only the round and the height. We also show that no constant confirmation depth is safe for a client. Both results use the private mining attack below. Throughout this section $\rho = t/(n - t) > \epsilon$,

so the adversary has the largest mining power that Assumption 1 allows. The proofs are in Section C.

In the attack, the adversary first mines on the public chain for r_0 rounds and publishes her blocks at once. The honest chain then grows at a rate of about $(1+\rho)f$. Next, she mines on this chain in private for s rounds, and sends the result to the client. This is the chain $\mathcal{C}$ of Figure 3.

Definition 6 (Private mining). *Let $r_0, s \geq 1$ be integers and $r = r_0 + s$. The strategy π_{private} with parameters r_0 and s is the following.*

1. *In every round of $[1, r_0]$, after the honest validators have sent their messages, the adversary spends every query on extending a longest chain among the genesis chain and the chains sent by any party so far, each block extending the one before. She sends every block to all validators at once.*
2. *Let $\mathcal{C}_0$ be a longest chain adopted by an honest validator at round $r_0 + 1$. In every round of $[r_0 + 1, r]$ the adversary spends every query on extending $\mathcal{C}_0$, each block extending the one before. She sends nothing to the validators.*
3. *At round r she sends the client the chain $\mathcal{C}$ formed by $\mathcal{C}_0$ and her blocks of the rounds of $[r_0 + 1, r]$.*

For $\alpha \in \mathbb{N}$, the strategy $\pi^{\alpha}_{\text{private}}$ is the same, except that at round r she sends the client $\mathcal{C}[:\alpha+1]$, the chain $\mathcal{C}$ cut at height α, whenever $\mathcal{C}$ has height at least α.

Lemma 2 (Private mining). *Let $r_0, s \geq \lambda$ be integers and $r = r_0 + s$, and let $\mathcal{C}_0$ and $\mathcal{C}$ be as in Definition 6. In a typical execution against π_{private}, for every chain $\mathcal{C}'$ adopted by an honest validator at round r we have $\mathcal{C} \cap \mathcal{C}' \preceq \mathcal{C}_0$, and*

$$|\mathcal{C}| - |\mathcal{C}_0| > (\rho - \epsilon) f s, \tag{4}$$

$$|\mathcal{C}| - 1 > (1 + \rho - 2\epsilon) f r_0 + (\rho - \epsilon) f s. \tag{5}$$

Constant depth. By Equation (4), the private part of $\mathcal{C}$ has more than $(\rho - \epsilon)fs$ blocks. A client that prunes a constant number of blocks k therefore keeps a private block once s is large enough.

Corollary 4 (No constant depth). *Let $k \geq 0$ be an integer, and let r_0, s and r be as in Lemma 2, with $(\rho - \epsilon) f s \geq k$. In a typical execution against π_{private}, a client that prunes k blocks from the chain $\mathcal{C}$ that π_{private} sends it at round r computes a chain that is a prefix of no chain adopted by an honest validator at round r.*

Lower bound. A *rule* is a function R that maps a round r and a height α to a number of blocks $R(r, \alpha) \in \mathbb{N}$. A client *runs* R if it computes $\mathcal{C}[:-R(r, \alpha)]$ from each chain $\mathcal{C}$ of height α that it receives at round r. Algorithm 1 runs $\lceil \gamma \rceil$.

If α is at most the right-hand side of Equation (5), then against $\pi^{\alpha}_{\text{private}}$ the client receives a chain of height α. Every block of this chain above $\mathcal{C}_0$ is private. By Lemma 1, $\mathcal{C}_0$ has height less than Afr_0, since $r_0 \geq \lambda$. So a safe rule prunes more than $\alpha - Afr_0$ blocks.

Theorem 3 (Lower bound). *Let R be a rule, and let r_0, s and r be as in Lemma 2. Let $\alpha \in \mathbb{N}$ be at most the right-hand side of Equation (5), with $R(r, \alpha) \leq \alpha - Afr_0$. In a typical execution against $\pi^{\alpha}_{\text{private}}$, a client running R computes at round r a chain that is a prefix of no chain adopted by an honest validator at round r.*

Comparison with the construction. Fix a round r and a height α. The adversary chooses r_0 and $s = r - r_0$ so that her chain reaches height α with r_0 as small as possible. Her chain has height about $(1+\rho)fr_0 + \rho fs = f(r_0 + \rho r)$, so $r_0 \approx \alpha/f - \rho r$. The bound of Theorem 3 is then about $\alpha - (1+\rho)fr_0 \approx \rho\,((1+\rho)fr - \alpha)$, which is Equation (1). Theorem 4 makes this precise: every safe rule and Algorithm 1 both prune Equation (1) blocks, up to a constant multiple of an error term c. This term is a fraction of order $f + \epsilon$ of the height fr of an honest chain, plus a number of blocks of order λf.

Theorem 4 (Tightness). *Let r be a round and α a height with $\rho fr + \lambda f \leq \alpha \leq (1 + \rho - 2\epsilon)fr - 2\lambda f$, and write $c = (f + \epsilon)fr + \lambda f + 1$.*

1. *Let R be a rule such that, in every typical execution, a client running R computes at round r a chain that is a prefix of every chain adopted by an honest validator at round r. Then $R(r, \alpha) > \rho\,((1+\rho)fr - \alpha) - 10c$.*
2. *$\gamma(r, \alpha) \leq \rho\,((1+\rho)fr - \alpha) + 16c$.*

Honest chains. When the adversary does not mine, a chain adopted by an honest validator at round r has height about fr, and Equation (1) is then $\rho^2 fr$. A rule prunes the same number of blocks from this chain as from the chain of the same height sent by $\pi^{\alpha}_{\text{private}}$. Hence every safe rule prunes at least about a ρ^2 fraction of an honest chain, which is what Algorithm 1 prunes. A transaction that enters the honest chain at round r' lies at height about fr'. At round r a safe rule keeps a chain of height at most about $(1-\rho^2)fr$, so it confirms the transaction only after about $\rho^2 r'/(1-\rho^2)$ rounds, as Algorithm 1 does (Section 4.2). Our results do not cover rules that also read timestamps, the target or the contents of blocks, since these are outside the model of Section 2.

6 Discussion

Variable difficulty. The analysis assumes a static population and a fixed target, under which the height of a chain stands in for the work inside it. Variable difficulty breaks that. An adversary who controls the timestamps of her own blocks can trigger the largest rise in target that the difficulty adjustment allows, and reach a given height for far less work. BitVM3 [13] reports that about a fifth of the hashing power suffices for this against an on-chain client that checks height alone, and repairs it by binding timestamps to the consensus time of the chain. A version of our rule for variable difficulty would compare accumulated work, rather than height, against the round. Variable difficulty exists because the total mining power is not known in advance. In that setting, Lewis-Pye and

Roughgarden [6] show that a live protocol cannot produce certificates, messages whose existence alone proves that a block is confirmed. We therefore posit that such a rule needs some synchrony. We leave it open.

Other longest chain protocols. Our analysis is for the synchronous network of [4] with $\Delta = 1$. In the bounded delay model [4,10] the honest chain grows at a rate that depends on the delay. Our construction works if the chain growth rates are adjusted accordingly. Longest chain protocols based on proof of stake [2,11] have the same structure of rates, with slots in place of rounds, and we expect the rule to carry over. We leave this for future work.

7 Conclusion

A client does not need synchrony or an honest peer to confirm bitcoin transactions. A clock is enough, together with a rule that prunes more as the chain it receives falls behind the round. The client stays safe whatever it receives, and it stays live whenever an honest chain reaches it, stale or not. The price is a confirmation delay that grows with the round, and no safe rule that reads only the round and the height avoids it.

References

1. Aumayr, L., Avarikioti, Z., Maffei, M., Scaffino, G., Zindros, D.: Blink: An optimal proof of proof-of-work. In: Financial Cryptography and Data Security 2025. pp. 173–190 (2025), IACR Cryptology ePrint Archive, Report 2024/692
2. Badertscher, C., Gaži, P., Kiayias, A., Russell, A., Zikas, V.: Ouroboros Genesis: Composable proof-of-stake blockchains with dynamic availability. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 913–930 (2018), IACR Cryptology ePrint Archive, Report 2018/378
3. Bünz, B., Kiffer, L., Luu, L., Zamani, M.: FlyClient: Super-light clients for cryptocurrencies. In: 2020 IEEE Symposium on Security and Privacy. pp. 928–946 (2020), IACR Cryptology ePrint Archive, Report 2019/226
4. Garay, J.A., Kiayias, A., Leonardos, N.: The Bitcoin backbone protocol: Analysis and applications. Journal of the ACM **71**(4), 25:1–25:49 (2024), journal version of the EUROCRYPT 2015 paper. IACR Cryptology ePrint Archive, Report 2014/765
5. Kiayias, A., Miller, A., Zindros, D.: Non-interactive proofs of proof-of-work. In: Financial Cryptography and Data Security 2020. pp. 505–522 (2020), IACR Cryptology ePrint Archive, Report 2017/963
6. Lewis-Pye, A., Roughgarden, T.: How does blockchain security dictate blockchain implementation? In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. pp. 1006–1019 (2021)
7. Neu, J., Tas, E.N., Tse, D.: Ebb-and-flow protocols: A resolution of the availability-finality dilemma. In: 2021 IEEE Symposium on Security and Privacy. pp. 446–465 (2021)
8. Neu, J., Tas, E.N., Tse, D.: The availability-accountability dilemma and its resolution via accountability gadgets. In: Financial Cryptography and Data Security 2022. pp. 541–559 (2022), arXiv:2105.06075

9. Neu, J., Tas, E.N., Tse, D.: Short paper: Accountable safety implies finality. In: Financial Cryptography and Data Security 2024. pp. 41–50. Springer (2024). https://doi.org/10.1007/978-3-031-78676-1_3
10. Pass, R., Seeman, L., Shelat, A.: Analysis of the blockchain protocol in asynchronous networks. In: Advances in Cryptology – EUROCRYPT 2017, Part II. pp. 643–673 (2017), IACR Cryptology ePrint Archive, Report 2016/454
11. Pass, R., Shi, E.: The sleepy model of consensus. In: Advances in Cryptology – ASIACRYPT 2017, Part II. pp. 380–409 (2017), IACR Cryptology ePrint Archive, Report 2016/918
12. Sridhar, S., Tas, E.N., Neu, J., Zindros, D., Tse, D.: Consensus under adversary majority done right. In: Financial Cryptography and Data Security 2025. pp. 108–126 (2025), IACR Cryptology ePrint Archive, Report 2024/1799
13. Woll, R.L., Alexopoulos, I., Aumayr, L., Avarikioti, Z., Maffei, M., Tse, D.: BitVM3: Efficient Bitcoin bridges via garbled circuits. In: Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (2026), IACR Cryptology ePrint Archive, Report 2026/933

A Model details

This appendix collects the parts of the model that only the proofs use.

A.1 Random variables

Fix a round r. Let $X_r = 1$ if some honest validator computes a block at round r, and $X_r = 0$ otherwise. Let $Y_r = 1$ if exactly one honest validator computes a block at round r, and $Y_r = 0$ otherwise. Let Z_r be the number of successful adversarial queries at round r. The adversary makes all her tq queries (Section 2), so Z_r is the number of successes among tq independent coins with success probability p, and $\mathbb{E}[Z_r] = pqt$. A round with $X_r = 1$ is successful and a round with $Y_r = 1$ is a convergence opportunity. For a set of rounds S we write $X(S) = \sum_{r \in S} X_r$ and define $Y(S)$ and $Z(S)$ in the same way. We set $X_0 = Y_0 = Z_0 = 0$, so that a set of rounds containing the index 0 contributes nothing at that index. So $f = \mathbb{E}[X_r]$ is the parameter of Section 2. From [4, Equation (1)],

$$(1 - f)\, pq(n - t) < f < pq(n - t). \tag{6}$$

Let $W_r = X_r - Y_r$, which is 1 if two or more honest validators compute a block at round r and 0 otherwise. So $f_2 = \mathbb{E}[W_r]$. For a set of rounds S let $W(S) = \sum_{r \in S} W_r = X(S) - Y(S)$, with $W_0 = 0$.

Lemma 3. $f_2 < \dfrac{f^2}{2(1-f)^2}$.

Proof of Lemma 3. Two or more of the $q(n-t)$ honest queries succeed with probability at most $\binom{q(n-t)}{2} p^2 \leq (pq(n-t))^2/2$, and $pq(n-t) < f/(1-f)$ by Equation (6). □

A.2 Typical executions

Definition 7 (Typical execution). *An execution is (ϵ, λ)-typical if two conditions hold. First, every set S of at least λ consecutive rounds satisfies*

$$(1-\epsilon)f|S| < X(S) < (1+\epsilon)f|S|, \qquad Y(S) > (1-\epsilon)(1-f)f|S|,$$
$$\left(\frac{t}{n-t} - \epsilon\right) f|S| < Z(S) < \left(\frac{t}{(n-t)(1-f)} + \epsilon\right) f|S|,$$
$$W(S) < \left(\frac{f_2}{f} + \epsilon\right) f|S|.$$

Second, no insertion, copy, prediction or guess occurs, in the sense of [4, Definition 4.3].

The upper bounds also hold for a set $[0, r]$ with $r \geq \lambda$, because $X([0, r]) = X([1, r])$, and likewise for Y, Z and W, while the right-hand sides increase with $|S|$. The definition is that of [4, Definition 4.4], in the form of [4, Lemma 4.6], with two bounds added, the bound on $W(S)$ and the lower bound on $Z(S)$. Since $t/(n-t) \leq \rho$ and $f_2 < \bar{f}_2$ (Lemma 3), the upper bounds on $Z(S)$ and $W(S)$ also hold with ρ in place of $t/(n-t)$ and $\bar{f}_2$ in place of f_2. The proofs use them in this form.

Theorem 5 (Typicality). *Let L be the total run time of the system, Q a bound on the number of oracle queries, and ν the min-entropy of the input contribution function. An execution is (ϵ, λ)-typical except with probability at most $\epsilon_{\text{typ}} = 6L^2 e^{-\Omega(\epsilon^2 \lambda f)} + 3Q^2 2^{-\kappa} + [(n-t)L]^2 2^{-\nu}$.*

For $\lambda = \Omega(\log^2 \kappa)$ the probability ϵ_{typ} is negligible in κ. The theorem is [4, Theorem 4.5] with two Chernoff bounds added, for $W(S)$ and for the lower bound on $Z(S)$, which is where its first coefficient is 6 rather than 4.

Proof of Theorem 5. By [4, Theorem 4.5], the probability that some condition of Definition 7 other than the two added bounds fails is at most $4L^2 e^{-\Omega(\epsilon^2 \lambda f)} + 3Q^2 2^{-\kappa} + [(n-t)L]^2 2^{-\nu}$. It remains to bound the probability that an added condition fails, and each is a Chernoff bound. $W(S)$ is a sum of $|S|$ independent Bernoulli trials with mean $\mu = f_2|S|$, the W_r of distinct rounds being independent for the same reason as the X_r in [4]. For $a > 0$ the Chernoff bound gives $\Pr[W(S) \geq \mu + a] \leq \exp(-a^2/(2\mu + a))$. Take $a = \epsilon f|S|$. Assumption 1 gives $f < 1/3$, so by Lemma 3 $2f_2 < f^2/(1-f)^2 < 3f^2$ and $3f + \epsilon < 2$, and

$$\frac{a^2}{2\mu + a} = \frac{\epsilon^2 f^2 |S|}{2f_2 + \epsilon f} > \frac{\epsilon^2 f^2 |S|}{3f^2 + \epsilon f} = \frac{\epsilon^2 f |S|}{3f + \epsilon} > \frac{\epsilon^2 f |S|}{2} \geq \frac{\epsilon^2 \lambda f}{2}.$$

So the bound on $W(S)$ fails for a given S with probability at most $e^{-\epsilon^2 \lambda f/2}$. A union bound over the at most L^2 sets S adds one term $L^2 e^{-\Omega(\epsilon^2 \lambda f)}$.

For the lower bound on $Z(S)$, if $t = 0$ then $Z(S) = 0$ and the bound holds, so let $t \geq 1$. $Z(S)$ counts the successes among $tq|S|$ independent coins with

success probability p, so its mean is $\mu = pqt|S|$. By Equation (6), $\mu > \frac{t}{n-t} f|S|$ and $\mu < \frac{t}{n-t} \cdot \frac{f|S|}{1-f}$. Assumption 1 gives $t \leq n - t$ and $f < 1/3$, so $\mu < \frac{3}{2} f|S|$. For $a > 0$ the Chernoff bound gives $\Pr[Z(S) \leq \mu - a] \leq \exp(-a^2/(2\mu))$. Take $a = \epsilon f|S|$. Then $\mu - a > \left(\frac{t}{n-t} - \epsilon\right) f|S|$ and

$$\frac{a^2}{2\mu} > \frac{\epsilon^2 f^2 |S|^2}{3f|S|} = \frac{\epsilon^2 f|S|}{3} \geq \frac{\epsilon^2 \lambda f}{3}.$$

So the lower bound on $Z(S)$ fails for a given S with probability at most $e^{-\epsilon^2 \lambda f/3}$, and a union bound over the at most L^2 sets S adds another term $L^2 e^{-\Omega(\epsilon^2 \lambda f)}$. □

A.3 The block tree

Two facts about the block tree follow from the protocol and from Definition 7.

Lemma 4 (Membership). *In a typical execution, for every round r, every valid chain that a client receives at round r is in $\mathcal{T}_r$, and every chain adopted by an honest validator at round r is in $\mathcal{T}_{r-1}$.*

Lemma 5 (Fork choice). *Let B be an honest block at height i, computed at round r. Then some honest validator has a chain of length greater than i at round $r + 1$. Every honest validator adopts a chain of length greater than i at every round $r' > r$.*

Proof of Lemma 4. Every block of a valid chain, other than the genesis block, passes a verification query to H. No guess occurs (Definition 7), so a computation query for the block came first, and the block was computed in a round of $[0, r]$. For the second claim, let B be a block of a chain adopted by an honest validator at round r, other than the genesis block. If the validator received B, it received it at a round at most r, so B was sent, and hence computed, at a round at most $r - 1$. Otherwise the validator computed B itself, at a round at most $r - 1$, because a validator adopts before it queries the oracle. So every block of the chain is in $\mathcal{T}_{r-1}$. □

Proof of Lemma 5. If B is the genesis block, then $i = r = 0$, every validator has a chain of length 1 at round 1, and every chain has length at least 1. Otherwise the honest validator that computed B appended B at round r to the chain it adopted at round r, made the result its local chain, and sent it. The result has length $i + 1$, so that validator has a chain of length greater than i at round $r + 1$. Every honest validator receives the result at the beginning of round $r + 1$, so it is an argument of `maxvalid`. Hence the chain adopted at round $r + 1$ has length greater than i. We claim that the same holds at every round $r' > r + 1$. The chain a validator has at round r' is the chain it adopted at round $r' - 1$, possibly extended by one block. By induction on r' it has length greater than i. The chain adopted at round r' is at least as long. □

A.4 Facts from the backbone analysis

We restate the following facts from [4] without proof.

Lemma 6 (Pairing Lemma, [4, Lemma 4.1]). *Let B be the block at height i of a chain $\mathcal{C}$, computed by an honest validator in a convergence opportunity. Then the block at height i of any chain is either B or adversarial.*

Lemma 7 (Chain Growth Lemma, [4, Lemma 4.2]). *If at round r an honest validator has a chain of length m, then at every round $r' \geq r$ every honest validator adopts a chain of length at least $m + X([r, r' - 1])$.*

Lemma 8 (Common Prefix Theorem, [4, Theorem 4.10]). *In a typical execution, let $\mathcal{C}_1$ and $\mathcal{C}_2$ be chains adopted by honest validators at rounds r_1 and r_2 with $r_1 \leq r_2$. Then, for every $k \geq 2\lambda f$, $\mathcal{C}_1[:-k] \preceq \mathcal{C}_2$.*

Lemma 9 (Patience Lemma, [4, Lemma 4.8]). *In a typical execution, any $k \geq 2\lambda f$ consecutive blocks of a chain have been computed in more than $k/(2f)$ consecutive rounds.*

Lemma 10 (Ledger liveness of the validators, after [4, Lemma 6.3]). *In a typical execution, let r be a round, let $u_V = \lceil 2\lceil k_0 \rceil / ((1-\epsilon)f) \rceil$, and let an honest transaction be given as input to every honest validator at every round from r on. Then, for every round $r_1 \geq r + u_V$ and every chain $\mathcal{C}$ adopted by an honest validator at round r_1, the transaction is in $\mathcal{C}[:-k_0]$.*

[4, Lemma 6.3] states this at depth $\lceil 2\lambda f \rceil$, with $\lceil 4\lambda/(1-\epsilon) \rceil$ rounds. Its proof uses chain growth and chain quality over $\lceil 2\lambda f \rceil$ consecutive blocks. Both apply at depth $\lceil k_0 \rceil \geq 2\lambda f$, so the proof goes through with u_V as above. Lemmas 7 and 8 carry the conclusion from round $r + u_V$ to every later round.

B Proofs for security

This appendix proves Theorems 1 and 2 and Corollary 1. Throughout, the execution is (ϵ, λ)-typical, Assumption 1 holds, the standing assumption on λ of Section 4 holds, and the constants A, ρ_1 and k_0 are those of Section 3. We use one more constant, $\rho_2 = \rho_1/(1-\epsilon)$.

B.1 Growth bounds

Over a long enough window of rounds, a chain gains at most about $\rho_1 f$ blocks per round above its fork point with an honest chain, and the honest validators extend their chain by at least $(1-\epsilon)f$ blocks per round. So ρ_2 is the number of blocks such a fork gains for every block the honest validators add. It tends to ρ as f and ϵ tend to 0. The first lemma uses honest majority to show that it stays below 1.

Lemma 11. $\rho_2 < 1$.

Proof of Lemma 11. By definition, $\rho_2 < 1$ is equivalent to $\frac{\rho}{1-f} + \frac{\bar{f}_2}{f} < 1 - 3\epsilon$. Here $\bar{f}_2/f = f/(2(1-f)^2)$, and $f < 1/3$ by Assumption 1. So we must show

$$\frac{\rho}{1-f} + \frac{f}{2(1-f)^2} < 1 - 3\epsilon.$$

Multiplying by $(1-f)^2$, this is $\rho(1-f) + \frac{f}{2} < (1-3\epsilon)(1-f)^2$, that is, $(1-\rho)(1-f) > \frac{3f}{2} - f^2 + 3\epsilon(1-f)^2$. Since $f < 1/3$, the right side is at most $(1-f)(2f+3\epsilon)$. So $1 - \rho > 2f + 3\epsilon$ suffices, and it follows from $3f + 3\epsilon < 1 - \rho$. □

The second lemma says that a chain holds at most one honest block from any single round. An honest validator mines on a chain it adopted at the start of the round, and every block of that chain was mined in an earlier round.

Lemma 12. *In a typical execution, a chain contains at most one honest block computed in any given round.*

Proof of Lemma 12. Suppose towards a contradiction that $\mathcal{C}[i]$ and $\mathcal{C}[i']$ with $i < i'$ are both honest and both computed at round r. Only the genesis block was computed at round 0, so $r \geq 1$. The honest validator that computed $\mathcal{C}[i']$ extended at round r the chain it adopted at round r. A block determines all of its ancestors, so that chain is $\mathcal{C}[:i']$, and it contains $\mathcal{C}[i]$. From Lemma 4, $\mathcal{C}[:i'] \in \mathcal{T}_{r-1}$, so $\mathcal{C}[i]$ was computed at a round at most $r-1$. This contradicts the assumption that $\mathcal{C}[i]$ was computed at round r. □

The growth bounds of Lemma 1 follow from Definition 7 and Lemma 12.

Proof of Lemma 1. Let $r' = \max(r, \lambda)$ and let $\mathcal{C} \in \mathcal{T}_r$. The blocks of $\mathcal{C}[1:]$ were computed in rounds of $[1, r']$. From Lemma 12, the honest ones were computed in distinct rounds, each of them successful. So there are at most $X([1, r'])$ honest blocks, and at most $Z([1, r'])$ adversarial ones. Since $r' \geq \lambda$, Definition 7 applies and

$$|\mathcal{C}| - 1 = |\mathcal{C}[1:]| < (1+\epsilon) f r' + \left(\frac{\rho}{1-f} + \epsilon\right) f r' = A f r',$$

where the last equality uses $\frac{\rho}{1-f} = \rho + \frac{\rho f}{1-f}$. This proves the first item. For the second, every honest validator has a chain of length 1 at round 1. From Lemma 7, every honest validator adopts at round r a chain of length at least $1 + X([1, r-1])$. Since $r - 1 \geq \lambda$, Definition 7 gives $X([1, r-1]) > (1-\epsilon) f (r-1)$, and hence $|\mathcal{C}| - 1 > (1-\epsilon) f (r-1)$. □

B.2 Counting

Fix a round r, a chain $\mathcal{C} \in \mathcal{T}_r$, and a chain $\mathcal{C}'$ that an honest validator adopts at round r. As in Section 3, their fork point is the last honest block of $\mathcal{C} \cap \mathcal{C}'$. The next lemma counts the blocks of each chain above the fork point. The honest validators kept extending $\mathcal{C}'$ after the fork point was mined, so $\mathcal{C}'$ gained at least one block in every successful round. Each block of $\mathcal{C}$ above the fork point is of

one of three kinds: either (a) it is adversarial, or (b) it is an honest block mined in a convergence opportunity, and then the block of $\mathcal{C}'$ at the same height is adversarial by Lemma 6, so it can be charged to that block, or (c) it is an honest block mined at round r, or in a round in which two or more honest validators found a block. So $\mathcal{C}$ gained at most as many blocks as the adversary mined, plus one for each round in which two or more honest validators found a block, plus one for round r.

Lemma 13 (Counting). *In a typical execution, let r be a round, let $\mathcal{C} \in \mathcal{T}_r$, and let $\mathcal{C}'$ be a chain adopted by an honest validator at round r. Let B^* be the fork point of $\mathcal{C}$ and $\mathcal{C}'$, that is, the last honest block of $\mathcal{C} \cap \mathcal{C}'$. Let i^* be its height, so that $B^* = \mathcal{C}[i^*] = \mathcal{C}'[i^*]$, and let r^* be the round in which it was computed. Put $S = [r^*+1, r-1]$ and $S' = [r^*, r]$. Then $r^* \leq r-1$, and*

1. *every block of $\mathcal{C}[i^*+1:]$ and of $\mathcal{C}'[i^*+1:]$ was computed in a round of S';*
2. *$|\mathcal{C}'[i^*+1:]| \geq X(S)$;*
3. *$|\mathcal{C}[i^*+1:]| \leq Z(S') + W(S) + 1$.*

Proof of Lemma 13. The genesis block is honest and lies on both chains, so B^* exists. From Definition 4 and Lemma 4, every block of $\mathcal{C}$ was computed in a round of $[0, r]$ and every block of $\mathcal{C}'$ in a round of $[0, r-1]$. The block B^* lies on $\mathcal{C}'$ and was computed at round r^*, so $r^* \leq r-1$. No predictions occur, so along a chain the round at which a block was computed is non-decreasing in the height. Hence every block of $\mathcal{C}[i^*+1:]$ and every block of $\mathcal{C}'[i^*+1:]$ was computed in a round of $[r^*, r] = S'$, which is the first item.

For the second item, B^* is honest, at height i^*, and was computed at round r^*. From Lemma 5, some honest validator has a chain of length at least i^*+1 at round r^*+1. Since $r^*+1 \leq r$, Lemma 7 from round r^*+1 to round r gives that every honest validator adopts at round r a chain of length at least $i^*+1+X([r^*+1, r-1]) = i^*+1+X(S)$. In particular $|\mathcal{C}'[i^*+1:]| = |\mathcal{C}'| - i^* - 1 \geq X(S)$.

For the third item, split the blocks of $\mathcal{C}[i^*+1:]$ into three classes: a adversarial blocks, h_1 honest blocks computed in a convergence opportunity of S, and h_2 remaining honest blocks. Then $|\mathcal{C}[i^*+1:]| = a + h_1 + h_2$. We bound h_2 first and $a + h_1$ second.

By the first item, every honest block of $\mathcal{C}[i^*+1:]$ was computed in a round of S'. None was computed at round r^*: the block B^* is honest and was computed at round r^*, and by Lemma 12 a chain has at most one honest block per round. So every honest block of $\mathcal{C}[i^*+1:]$ was computed in a round of $S \cup \{r\}$. The h_2 blocks are the honest blocks of $\mathcal{C}[i^*+1:]$ computed at round r or in a successful round of S that is not a convergence opportunity. From Lemma 12 again, they were computed in distinct rounds. So at most one of them was computed at round r, and at most $W(S)$ of them were computed in successful rounds of S that are not convergence opportunities. Hence $h_2 \leq W(S) + 1$.

We claim that $a + h_1 \leq Z(S')$. Define a map from the $a + h_1$ blocks of the first two classes to adversarial blocks computed in rounds of S'. An adversarial $\mathcal{C}[i]$ is mapped to itself. It was computed in a round of S' by the first item. An

honest $\mathcal{C}[i]$ computed in a convergence opportunity $r' \in S$ is mapped to $\mathcal{C}'[i]$. We check that $\mathcal{C}'[i]$ is such a block. The honest validator that computed $\mathcal{C}[i]$ sent a chain of length $i+1$ at round $r' \leq r-1$, so from Lemma 5 $|\mathcal{C}'| > i$ and $\mathcal{C}'[i]$ exists. Since $i > i^*$ and $\mathcal{C}[i]$ is honest, $\mathcal{C}'[i] \neq \mathcal{C}[i]$ by the maximality of i^*, so $\mathcal{C}'[i]$ is adversarial by Lemma 6. It lies on $\mathcal{C}'[i^*+1:]$, so by the first item it was computed in a round of S'. The map is injective: two blocks at distinct heights of one chain are distinct, and $\mathcal{C}[i] = \mathcal{C}'[i']$ would force $i = i'$ since no copies occur, with $\mathcal{C}[i]$ adversarial and $\mathcal{C}[i']$ honest. Each adversarial block is the output of a distinct successful adversarial query, and $Z(S')$ counts the successful adversarial queries in the rounds of S'. Hence $a + h_1 \leq Z(S')$, and $|\mathcal{C}[i^*+1:]| = a + h_1 + h_2 \leq Z(S') + W(S) + 1$. □

B.3 Safety

The first step compares $\mathcal{C}$ with $\mathcal{C}'$ directly. Let d be the number of blocks of $\mathcal{C}$ above their fork point. Every block of $\mathcal{C}$ that is not on $\mathcal{C}'$ is among them, so it is enough to bound d. Above the fork point, $\mathcal{C}'$ has d blocks plus the amount by which it is longer than $\mathcal{C}$. The honest validators mined these in successful rounds, and over the same rounds the fork gained at most about ρ_2 blocks for each of them. So d is at most about ρ_2 times the sum of d and the difference in length. Rearranging, $\mathcal{C}$ has about $\rho_2/(1-\rho_2)$ blocks off $\mathcal{C}'$ for every block by which it is shorter, plus a constant.

Lemma 14 (Prefix agreement). *In a typical execution, for every round r, every chain $\mathcal{C} \in \mathcal{T}_r$ and every chain $\mathcal{C}'$ adopted by an honest validator at round r,*

$$|\mathcal{C} \cap \mathcal{C}'| > |\mathcal{C}| - \max\left\{k_0,\ \frac{\rho_2\left(|\mathcal{C}'| - |\mathcal{C}|\right) + 2f + 1}{1-\rho_2}\right\}.$$

Proof of Lemma 14. Let B^*, i^*, r^*, S and S' be as in Lemma 13, and write $d = |\mathcal{C}[i^*+1:]| = |\mathcal{C}| - i^* - 1$. Since $\mathcal{C} \cap \mathcal{C}'$ contains B^*, we have $|\mathcal{C} \cap \mathcal{C}'| \geq i^* + 1 = |\mathcal{C}| - d$, so it suffices to bound d. If $d < k_0$ we are done, so assume $d \geq k_0 = 2(\lambda+2)f$.

We first show that $|S| > \lambda$. The subsequence $\mathcal{C}[i^*+1:]$ consists of d consecutive blocks of $\mathcal{C}$, so from Lemma 9 they were computed in more than $d/(2f) \geq \lambda + 2$ consecutive rounds. From Lemma 13, all these rounds lie in $S' = [r^*, r]$, which has $|S| + 2$ elements. Hence $|S| > \lambda$, and Definition 7 applies to S and to S'.

From Lemma 13 and Definition 7,

$$d < \left(\frac{\rho}{1-f} + \epsilon\right) f(|S|+2) + \left(\frac{\bar{f}_2}{f} + \epsilon\right) f|S| + 1.$$

From Lemma 11, $\frac{\rho}{1-f} + \epsilon < 1$, so the first term is smaller than $\left(\frac{\rho}{1-f} + \epsilon\right) f|S| + 2f$. Therefore

$$d < \rho_1 f|S| + 2f + 1. \tag{7}$$

From Lemma 13 and Definition 7, $|\mathcal{C}'| - i^* - 1 \geq X(S) > (1-\epsilon)f|S|$, so $f|S| < (|\mathcal{C}'| - i^* - 1)/(1-\epsilon)$ and

$$d < \rho_2(|\mathcal{C}'| - i^* - 1) + 2f + 1.$$

Substitute $|\mathcal{C}'| - i^* - 1 = (|\mathcal{C}'| - |\mathcal{C}|) + d$. This gives $d < \rho_2(|\mathcal{C}'| - |\mathcal{C}|) + \rho_2 d + 2f + 1$, and $\rho_2 < 1$ by Lemma 11, so $d < \frac{\rho_2(|\mathcal{C}'| - |\mathcal{C}|) + 2f + 1}{1-\rho_2}$. □

The coefficient $\rho_2/(1-\rho_2)$ tends to $\rho/(1-\rho)$. The client cannot use Lemma 14 as it stands, because it does not know $|\mathcal{C}'|$. The second step replaces $|\mathcal{C}'|$ by the round, with the argument of Section 3. The fork point is at height $|\mathcal{C}| - 1 - d$, and no chain reaches that height before round about $(|\mathcal{C}| - 1 - d)/(Af)$. The fork had only the remaining rounds to grow, at most $\rho_1 f$ blocks in each.

Lemma 15 (Round form). *In a typical execution, for every round r, every chain $\mathcal{C} \in \mathcal{T}_r$ and every chain $\mathcal{C}'$ adopted by an honest validator at round r,*

$$|\mathcal{C} \cap \mathcal{C}'| > |\mathcal{C}| - \max\left\{k_0,\ \frac{\rho_1\left[Af(r+\lambda) - |\mathcal{C}| + 1\right] + A(2f+1)}{1 - \bar{f}_2/f}\right\}.$$

Proof of Lemma 15. Let B^*, i^*, r^* and S be as in Lemma 13 and write $d = |\mathcal{C}[i^*+1:]| = |\mathcal{C}| - i^* - 1$. As in the proof of Lemma 14, $|\mathcal{C} \cap \mathcal{C}'| \geq i^* + 1 = |\mathcal{C}| - d$, so it suffices to bound d by the maximum in the statement, and we may assume $d \geq k_0$, which gives $|S| > \lambda$ and Equation (7).

The prefix $\mathcal{C}[:i^*+1]$ is a chain of $\mathcal{T}_{r^*}$, because its tip B^* was computed at round r^* and no predictions occur. From Lemma 1, $i^* < Af \max(r^*, \lambda) \leq Af(r^* + \lambda)$, hence $r^* > \frac{i^*}{Af} - \lambda$. Together with $|S| = r - r^* - 1 < r - r^*$ and Equation (7) this gives

$$d < \rho_1 f(r+\lambda) - \frac{\rho_1}{A} i^* + 2f + 1.$$

Put $c = \rho_1/A$ and observe that $c < 1$: $\rho_1 = (1-\epsilon)\rho_2 < 1$ by Lemma 11, and $A > 1$ by its definition. Substitute $i^* = |\mathcal{C}| - 1 - d$ and rearrange,

$$d < \frac{\rho_1 f(r+\lambda) - c\,(|\mathcal{C}| - 1) + 2f + 1}{1 - c}.$$

Multiplying numerator and denominator by A gives

$$d < \frac{\rho_1\left[Af(r+\lambda) - |\mathcal{C}| + 1\right] + A(2f+1)}{A - \rho_1},$$

and the definitions of A and ρ_1 give $A - \rho_1 = 1 - \bar{f}_2/f$. □

The maximum in Lemma 15 is the pruning depth $\gamma(r, |\mathcal{C}|-1)$ of Equation (2). So the chain the client computes from $\mathcal{C}$ is a prefix of every chain that an honest validator adopts at the round of receipt. For later rounds, Section 4 splits into two cases. The first uses the common prefix property (Lemma 8). The second

needs the same property for a chain that no honest validator holds, and the next lemma repeats the proof of [4, Theorem 4.10] for such a chain.

The proof rests on a statement about a single round [4, Lemma 4.9]. Ours is Lemma 14 for a chain that is at least as long as an honest chain, where the second term of the maximum is at most $(2f+1)/(1-\rho_2)$. Since $1-\rho_2 = (1-\epsilon-\rho_1)/(1-\epsilon)$, the standing assumption on λ reads $\lambda \geq \frac{2f+1}{2f(1-\rho_2)} - 2$. Then $k_0 = 2(\lambda+2)f \geq (2f+1)/(1-\rho_2)$, and Lemma 14 gives, for every round r, every chain $\mathcal{C}_1 \in \mathcal{T}_r$ and every chain $\mathcal{C}_2$ adopted by an honest validator at round r with $|\mathcal{C}_1| \geq |\mathcal{C}_2|$,

$$|\mathcal{C}_1 \cap \mathcal{C}_2| > |\mathcal{C}_1| - k_0. \tag{8}$$

A chain of the block tree that is at least as long as an honest chain thus agrees with it on all but fewer than k_0 of its blocks, as two honest chains do.

Lemma 16 (Long chains). *In a typical execution, let r be a round, and let $\mathcal{C} \in \mathcal{T}_r$ be at least as long as every chain adopted by an honest validator at round r. Then, for every round $r' \geq r$ and every chain $\mathcal{C}'$ adopted by an honest validator at round r', $\mathcal{C}[:-k_0] \preceq \mathcal{C}'$.*

Proof of Lemma 16. Write $k = \lceil k_0 \rceil$, so that $\mathcal{C}[:-k_0] = \mathcal{C}[:-k]$. If $k \geq |\mathcal{C}| - 1$ then $\mathcal{C}[:-k]$ has at most one block, so it is the genesis chain or empty and is a prefix of every chain. There is nothing to prove, so assume $k < |\mathcal{C}| - 1$, which gives $|\mathcal{C}[:-k]| = |\mathcal{C}| - k$.

At round $r' = r$, the chain $\mathcal{C}'$ is no longer than $\mathcal{C}$, so Equation (8) gives $|\mathcal{C} \cap \mathcal{C}'| > |\mathcal{C}| - k_0 \geq |\mathcal{C}[:-k]|$. Both $\mathcal{C}[:-k]$ and $\mathcal{C} \cap \mathcal{C}'$ are prefixes of $\mathcal{C}$, so $\mathcal{C}[:-k] \preceq \mathcal{C}'$.

For later rounds we follow the proof of [4, Theorem 4.10]. Suppose, towards a contradiction, that the claim fails. Let $r_B > r$ be the first round at which an honest validator V adopts a chain $\mathcal{C}_B$ with $\mathcal{C}[:-k] \not\preceq \mathcal{C}_B$, and let $\mathcal{C}_A$ be the chain V adopted at round $r_B - 1$. Then $\mathcal{C}[:-k] \preceq \mathcal{C}_A$: by the case $r' = r$ if $r_B - 1 = r$, by the choice of r_B otherwise. At round r_B the chain V has is $\mathcal{C}_A$ extended by at most one block. It contains $\mathcal{C}[:-k]$, so it differs from $\mathcal{C}_B$, and since `maxvalid` returns a longest argument, $|\mathcal{C}_B| \geq |\mathcal{C}_A|$.

First compare $\mathcal{C}_B$ with $\mathcal{C}_A$. Both $\mathcal{C}[:-k]$ and $\mathcal{C}_B \cap \mathcal{C}_A$ are prefixes of $\mathcal{C}_A$ and $\mathcal{C}[:-k] \not\preceq \mathcal{C}_B$, so $|\mathcal{C}_B \cap \mathcal{C}_A| < |\mathcal{C}[:-k]|$. On the other hand, $\mathcal{C}_B \in \mathcal{T}_{r_B-1}$ by Lemma 4 and $|\mathcal{C}_B| \geq |\mathcal{C}_A|$, so Equation (8) at round $r_B - 1$ gives $|\mathcal{C}_B \cap \mathcal{C}_A| > |\mathcal{C}_B| - k_0 \geq |\mathcal{C}_B| - k$. Hence $|\mathcal{C}_B| - k < |\mathcal{C}| - k$, that is, $|\mathcal{C}_B| < |\mathcal{C}|$.

Now compare $\mathcal{C}$ with $\mathcal{C}_B$. Since $\mathcal{C} \in \mathcal{T}_{r_B}$ and $|\mathcal{C}| > |\mathcal{C}_B|$, Equation (8) at round r_B gives $|\mathcal{C} \cap \mathcal{C}_B| > |\mathcal{C}| - k_0 \geq |\mathcal{C}| - k = |\mathcal{C}[:-k]|$. Both $\mathcal{C}[:-k]$ and $\mathcal{C} \cap \mathcal{C}_B$ are prefixes of $\mathcal{C}$, so $\mathcal{C}[:-k] \preceq \mathcal{C} \cap \mathcal{C}_B \preceq \mathcal{C}_B$. This contradicts the choice of $\mathcal{C}_B$. □

Proof of Theorem 1. From Lemma 4, $\mathcal{C} \in \mathcal{T}_r$. Write $k = \lceil \gamma(r,\alpha) \rceil$, so that $\mathcal{C}[:-\gamma(r,\alpha)] = \mathcal{C}[:-k]$. If $\mathcal{C}[:-k]$ has at most one block, it is the genesis chain or empty, and it is a prefix of every chain. Otherwise $|\mathcal{C}[:-k]| = |\mathcal{C}| - k \leq |\mathcal{C}| - \lceil k_0 \rceil$, since $\gamma(r,\alpha) \geq k_0$.

Suppose first that some chain $\mathcal{C}_1$ adopted by an honest validator at round r is longer than $\mathcal{C}$. Lemma 15 gives $|\mathcal{C} \cap \mathcal{C}_1| > |\mathcal{C}| - \gamma(r,\alpha) \geq |\mathcal{C}[:-k]|$, and both $\mathcal{C}[:-k]$

and $\mathcal{C} \cap \mathcal{C}_1$ are prefixes of $\mathcal{C}$, so $\mathcal{C}[:-k] \preceq \mathcal{C}_1$. Moreover $|\mathcal{C}[:-k]| \leq |\mathcal{C}| - \lceil k_0 \rceil < |\mathcal{C}_1| - \lceil k_0 \rceil$, so $\mathcal{C}[:-k] \preceq \mathcal{C}_1[:-k_0]$. Since $k_0 \geq 2\lambda f$, Lemma 8 gives $\mathcal{C}_1[:-k_0] \preceq \mathcal{C}'$ for every chain $\mathcal{C}'$ adopted by an honest validator at any round $r' \geq r$.

Otherwise $\mathcal{C}$ is at least as long as every chain adopted by an honest validator at round r. Since $\gamma(r, \alpha) \geq k_0$, we have $\mathcal{C}[:-k] \preceq \mathcal{C}[:-k_0]$, and Lemma 16 gives $\mathcal{C}[:-k_0] \preceq \mathcal{C}'$. In this case Lemma 15 is not needed: the floor k_0 alone makes the client safe. □

Proof of Corollary 1. Exchanging the two clients if needed, assume $r_1 \leq r_2$. Since $t < n$, some validator is honest, and it adopts a chain $\mathcal{C}'$ at round r_2. Theorem 1, applied to each client with $r' = r_2$, gives $\mathcal{C}_1 \preceq \mathcal{C}'$ and $\mathcal{C}_2 \preceq \mathcal{C}'$. Two prefixes of one chain are prefixes of one another. So the chains a single client computes are pairwise comparable. Its output is the longest of them with more than one block, which is well defined, and each new output is longer than the previous one and therefore extends it. The output is computed from one received chain, so Theorem 1 applies to it. □

Proof of Corollary 2. By Corollary 1, any two chains that clients output, at any two rounds, are prefixes of one another, unless one of them is $\perp$. The ledgers of two chains one of which is a prefix of the other are prefixes of one another, and the empty ledger is a prefix of every ledger. □

B.4 Liveness

The honest chain keeps growing after the transaction is settled in it. The proof shows that, from round $r + u(r, D)$ on, it has grown by more than the client prunes.

Proof of Theorem 2. Write $\theta = 1 - \epsilon - \rho_1$ and $B = A/(1 - \bar{f}_2/f)$. By Lemma 11, $\rho_1 = (1 - \epsilon)\rho_2 < 1 - \epsilon$, so $\theta > 0$ and $\rho_1 < 1$. Assumption 1 gives $f < 1/3$ and $\epsilon < 1/3$, so $1 - \epsilon > 2/3$, and $\lambda \geq 2/f$ gives $\lambda \geq 6$.

Let $r_1 = r + u_V$, with u_V as in Lemma 10, and let $\mathcal{C}_1$ be a chain adopted by an honest validator at round r_1. By Lemma 10, the transaction is in $\mathcal{C}_1[:-k_0]$. Let $\mathcal{C}$ be the D-stale chain the client receives at round r', adopted by an honest validator at round $r'' = r' - D$, and let α be its height. Write $k = \lceil \gamma(r', \alpha) \rceil$, so that the client computes $\mathcal{C}[:-k]$ from $\mathcal{C}$. We show that $\mathcal{C}_1[:-k_0] \preceq \mathcal{C}[:-k]$. Then $\mathcal{C}[:-k]$ contains the transaction, which is not in the genesis block, so $\mathcal{C}[:-k]$ has more than one block. By Corollary 1, the output of the client then has $\mathcal{C}[:-k]$ as a prefix at every round from r' on, and so it contains the transaction.

First, r' is large enough. Since $\lceil k_0 \rceil \leq 2(\lambda + 2)f + 1$ and $2/f \leq \lambda$,

$$u_V \;\leq\; \frac{5\lambda + 8}{1 - \epsilon} + 1 \;<\; \frac{3}{2}(5\lambda + 8) + 1 \;\leq\; 10\lambda.$$

Since $A > 1$, we have $1 - \rho_1/A \geq 1 - \rho_1 \geq \theta$. With $\theta \leq 1$ and Equation (3), the choice $r' \geq r + u(r, D)$ gives

$$r'' = r' - D \;\geq\; \frac{(1 - \rho_1/A)\, r}{\theta} + \frac{D + 13\lambda}{\theta} - D \;\geq\; r + 13\lambda \;\geq\; r_1 + 3\lambda.$$

Also $r_1 \geq u_V \geq 2k_0/f > \lambda$. So $r_1 > \lambda$, $r'' > r_1$, and the set $S = [r_1 + 1, r'' - 1]$ has at least λ rounds.

Next, we compare the lengths of $\mathcal{C}$ and $\mathcal{C}_1$. The honest validator that adopted $\mathcal{C}_1$ at round r_1 has at round r_1+1 a chain at least as long as $\mathcal{C}_1$. Since $r'' \geq r_1+1$, Lemma 7 and Definition 7 give the first of the following bounds, and Lemma 1 gives the second, since $r_1 > \lambda$:

$$\begin{aligned} |\mathcal{C}| \;&\geq\; |\mathcal{C}_1| + X(S) \;>\; |\mathcal{C}_1| + (1-\epsilon) f(r'' - r_1 - 1), \\ |\mathcal{C}_1| - 1 \;&>\; (1-\epsilon) f(r_1 - 1). \end{aligned}$$

We now show that $|\mathcal{C}[:-k]| \geq |\mathcal{C}_1| - \lceil k_0 \rceil$, that is, that the client prunes at most $|\mathcal{C}| - |\mathcal{C}_1| + \lceil k_0 \rceil$ blocks. If $\gamma(r', \alpha) = k_0$, this follows from $|\mathcal{C}| \geq |\mathcal{C}_1|$. Otherwise, put $h(r') = \rho_1 f(r'+\lambda) + 2f + 1$. Since $A - \rho_1 = 1 - \bar{f}_2/f$, Equation (2) gives

$$\alpha - \gamma(r', \alpha) = B\big(\alpha - h(r')\big).$$

Since $|\mathcal{C}[:-k]| = \alpha + 1 - k > \alpha - \gamma(r', \alpha)$, it suffices that $B(\alpha - h(r')) \geq |\mathcal{C}_1|$. As $1/B = 1 - \rho_1/A$, this reads

$$\alpha - h(r') \;\geq\; \Big(1 - \frac{\rho_1}{A}\Big)|\mathcal{C}_1|.$$

By the two bounds on $|\mathcal{C}|$ and $|\mathcal{C}_1|$ above, it holds if

$$(1-\epsilon) f(r'' - r_1 - 1) + \frac{\rho_1}{A}(1-\epsilon) f(r_1 - 1) \;\geq\; h(r') + 1.$$

Substituting $r'' = r' - D$ and the definition of $h(r')$, and dividing by f, this reads

$$\theta r' \;\geq\; (1-\epsilon)\Big(1 - \frac{\rho_1}{A}\Big) r_1 + (1-\epsilon)\Big(D + 1 + \frac{\rho_1}{A}\Big) + \rho_1 \lambda + 2 + \frac{2}{f}.$$

Since $1 - \epsilon \leq 1$, $\rho_1/A < 1$, $\rho_1 < 1$ and $2/f \leq \lambda$, the right-hand side is at most

$$\Big(1 - \frac{\rho_1}{A}\Big)(r + u_V) + D + 2\lambda + 4 \;\leq\; \Big(1 - \frac{\rho_1}{A}\Big) r + D + 13\lambda,$$

where the last step uses $u_V \leq 10\lambda$ and $4 \leq \lambda$. By Equation (3), the right-hand side is $\theta\,(r + u(r, D))$, which is at most $\theta r'$.

Finally, since $k_0 \geq 2\lambda f$ and $r_1 < r''$, Lemma 8 gives $\mathcal{C}_1[:-k_0] \preceq \mathcal{C}$. Also $\mathcal{C}[:-k] \preceq \mathcal{C}$ and $|\mathcal{C}[:-k]| \geq |\mathcal{C}_1| - \lceil k_0 \rceil = |\mathcal{C}_1[:-k_0]|$, so $\mathcal{C}_1[:-k_0] \preceq \mathcal{C}[:-k]$. □

Proof of Corollary 3. Let an honest transaction tx be given as input to every honest validator at every round from r on, let P be a client, and let $r' = \lceil r + u(r, D) \rceil$. Since $1 - \epsilon - \rho_1 < 1$, we have $r + u(r, D) > D$, so P receives a D-stale chain at round r'. By Theorem 2, the chain P outputs contains tx at every round from r' on. So $\mathsf{tx} \in \mathcal{L}^P_{r''}$ for every round $r'' \geq r + u(r, D)$. □

C Proofs for tightness

Throughout, $\rho = t/(n-t) > \epsilon$, as in Section 5. A block computed at round r may be extended by a query made in the same round. Lemma 1 counts adversarial blocks by Z, which counts queries, so it already allows this.

Proof of Lemma 2. Write $S_1 = [1, r_0]$, $S_2 = [r_0 + 1, r]$ and $\ell_0 = |\mathcal{C}_0|$. In the rounds of S_1 the public chain gains one block in every successful round and one block for every adversarial block. In the rounds of S_2 only the private branch of the adversary grows on top of $\mathcal{C}_0$. We first establish three facts, and then bound the random variables in them with Definition 7. The first two hold in every execution, and the third in every typical one.

First, $\ell_0 = 1 + X(S_1) + Z(S_1)$. We show by induction on $r' \in [1, r_0 + 1]$ that every honest validator adopts at round r' a chain of length $L_{r'} = 1 + X([1, r' - 1]) + Z([1, r' - 1])$, and that the longest chain among the genesis chain and the chains sent by round $r' - 1$ has length $L_{r'}$. Both hold at round 1. Assume both at a round $r' \leq r_0$. Honest validators send at round r' chains of length at most $L_{r'} + X_{r'}$, and one of length $L_{r'} + 1$ if $X_{r'} = 1$. So, before the adversary acts, the longest chain among the genesis chain and the chains sent at rounds up to r' has length $L_{r'} + X_{r'}$. The adversary extends it by $Z_{r'}$ blocks and sends the result, of length $L_{r'+1}$. At round $r' + 1$ every honest validator receives it, and the chain it has is at most $L_{r'} + X_{r'}$ long, so it adopts a chain of length $L_{r'+1}$. The case $r' = r_0 + 1$ is the claim.

Second, $|\mathcal{C}| = \ell_0 + Z(S_2)$. In the rounds of S_2 the adversary queries only on the private branch, and each of her blocks extends the one before.

Third, $\mathcal{C} \cap \mathcal{C}' \preceq \mathcal{C}_0$ for every chain $\mathcal{C}'$ adopted by an honest validator at round r. The blocks the adversary computes in the rounds of S_2 go to the client only, and a typical execution has no copies, so none lies on $\mathcal{C}'$. The other blocks of $\mathcal{C}$ are those of $\mathcal{C}_0$.

Now the random variables. The sets S_1 and S_2 have $r_0 \geq \lambda$ and $s \geq \lambda$ rounds, and $t/(n-t) = \rho$, so Definition 7 gives $X(S_1) > (1-\epsilon) f r_0$, $Z(S_1) > (\rho - \epsilon) f r_0$ and $Z(S_2) > (\rho - \epsilon) f s$. Then the third fact is the first claim, the second fact gives Equation (4), and the first two facts give Equation (5), since $|\mathcal{C}| - 1 = X(S_1) + Z(S_1) + Z(S_2)$. □

Proof of Corollary 4. Consider a typical execution, and let $\mathcal{C}'$ be a chain adopted by an honest validator at round r. By Equation (4) and the bound on k, $|\mathcal{C}| - |\mathcal{C}_0| > (\rho - \epsilon) f s \geq k$. So $\mathcal{C}[:-k]$ has length $|\mathcal{C}| - k > |\mathcal{C}_0|$. Suppose that $\mathcal{C}[:-k]$ is a prefix of $\mathcal{C}'$. It is also a prefix of $\mathcal{C}$, so it is a prefix of $\mathcal{C} \cap \mathcal{C}'$, and hence of $\mathcal{C}_0$ by Lemma 2. This contradicts $|\mathcal{C}[:-k]| > |\mathcal{C}_0|$. □

Proof of Theorem 3. The strategy $\pi^{\alpha}_{\text{private}}$ acts as π_{private} in every round, and only the chain that the client receives at round r differs. Clients send nothing, so Lemma 2 applies to the chain $\mathcal{C}$ that the adversary forms at round r. Consider a typical execution, and let $\mathcal{C}'$ be a chain adopted by an honest validator at round r. We show in three steps that the chain the client computes at round r is not a prefix of $\mathcal{C}'$.

First, the client receives a chain of height α. By Equation (5) and the bound on α, $\mathcal{C}$ has height greater than α. So the adversary sends $\mathcal{C}_\alpha = \mathcal{C}[:\alpha+1]$.

Second, $\mathcal{C}_0$ is short. It was adopted by an honest validator at round r_0+1, so $\mathcal{C}_0 \in \mathcal{T}_{r_0}$ by Lemma 4. The execution is typical and $r_0 \geq \lambda$, so $|\mathcal{C}_0|-1 < Afr_0$ by Lemma 1.

Third, the client keeps more than $\mathcal{C}_0$. It computes $\mathcal{C}_\alpha[:-R(r,\alpha)]$. By the bound on $R(r,\alpha)$ and the second step,

$$|\mathcal{C}_\alpha[:-R(r,\alpha)]| = \alpha + 1 - R(r,\alpha) \;\geq\; Afr_0 + 1 \;>\; |\mathcal{C}_0|.$$

Now suppose that $\mathcal{C}_\alpha[:-R(r,\alpha)]$ is a prefix of $\mathcal{C}'$. It is also a prefix of $\mathcal{C}_\alpha$, and hence of $\mathcal{C}$, so it is a prefix of $\mathcal{C} \cap \mathcal{C}'$. By Lemma 2, it is then a prefix of $\mathcal{C}_0$. This contradicts the third step. □

Proof of Theorem 4. Assumption 1 gives $\rho + 3f + 3\epsilon < 1$, so $f, \epsilon < 1/3$, and $\lambda \geq 2/f \geq 6$. Hence

$$\frac{\rho f}{1-f} \;\leq\; \frac{3}{2}f, \qquad A \;\leq\; 1 + \rho + 2\epsilon + \frac{3}{2}f \;<\; 2.$$

Write $I = \rho((1+\rho)fr - \alpha)$. Since $\alpha \leq (1+\rho)fr$, we have $0 \leq I < 2fr$.

For the first item, let $r_0 = \lceil (\alpha/f - (\rho-\epsilon)r)/(1-\epsilon) \rceil$ and $s = r - r_0$. The lower and the upper bound on α give

$$r_0 \;\geq\; \frac{\epsilon r + \lambda}{1-\epsilon} \;\geq\; \lambda, \qquad s \;\geq\; \frac{2\lambda}{1-\epsilon} - 1 \;\geq\; \lambda,$$

and the choice of r_0 gives

$$(1+\rho-2\epsilon)fr_0 + (\rho-\epsilon)fs = f\big[(1-\epsilon)r_0 + (\rho-\epsilon)r\big] \;\geq\; \alpha.$$

So Theorem 3 applies. Consider a typical execution against $\pi^{\alpha}_{\text{private}}$. By the hypothesis on R, a client running R computes at round r a prefix of every chain adopted by an honest validator. By Theorem 3, this cannot happen if $R(r,\alpha) \leq \alpha - Afr_0$. So $R(r,\alpha) > \alpha - Afr_0$. Write $\beta = A/(1-\epsilon) - (1+\rho)$. Since $r_0 < (\alpha - (\rho-\epsilon)fr)/((1-\epsilon)f) + 1$,

$$\begin{aligned} \alpha - Afr_0 &> \alpha - \frac{A}{1-\epsilon}\big(\alpha - (\rho-\epsilon)fr\big) - Af \\ &= I - \beta\big(\alpha - (\rho-\epsilon)fr\big) - \epsilon(1+\rho)fr - Af. \end{aligned}$$

We bound the three subtracted terms. First,

$$A - (1+\rho)(1-\epsilon) = 2\epsilon + \frac{\rho f}{1-f} + (1+\rho)\epsilon \;\leq\; 4\epsilon + \frac{3}{2}f,$$

and $1-\epsilon > 2/3$, so $0 \leq \beta \leq 6\epsilon + \frac{9}{4}f$. Since $0 \leq \alpha - (\rho-\epsilon)fr \leq (1+\epsilon)fr < \frac{4}{3}fr$,

$$\beta\big(\alpha - (\rho-\epsilon)fr\big) \;\leq\; (8\epsilon + 3f)fr.$$

Second, $\epsilon(1+\rho)fr \le 2\epsilon fr$. Third, $Af < 1$. Hence

$$R(r,\alpha) \;>\; I - (10\epsilon + 3f)fr - 1 \;\ge\; I - 10c.$$

For the second item, if $\gamma(r,\alpha) = k_0$, then

$$\gamma(r,\alpha) = 2(\lambda+2)f \;\le\; 2\lambda f + 2 \;\le\; I + 2c.$$

Otherwise, write $g = \bar{f}_2/f = f/(2(1-f)^2)$. Since $f < 1/3$,

$$g \;\le\; \frac{9}{8}f \;\le\; \frac{3}{8}, \qquad \frac{1}{1-g} \;\le\; \frac{8}{5}.$$

By Equation (2),

$$(1-g)\gamma(r,\alpha) - I = \big(\rho_1 A - \rho(1+\rho)\big)fr - (\rho_1 - \rho)\alpha + \rho_1 Af\lambda + A(2f+1).$$

We bound the terms on the right. By the definitions of ρ_1 and A,

$$0 \;\le\; \rho_1 - \rho = \frac{\rho f}{1-f} + g + 2\epsilon \;\le\; \frac{21}{8}f + 2\epsilon, \qquad A - (1+\rho) \;\le\; 2\epsilon + \frac{3}{2}f,$$

so, with $A < 2$,

$$\rho_1 A - \rho(1+\rho) = (\rho_1 - \rho)A + \rho\big(A - (1+\rho)\big) \;\le\; \frac{27}{4}f + 6\epsilon.$$

The term $-(\rho_1 - \rho)\alpha$ is at most 0. By Lemma 11, $\rho_1 < 1$, so $\rho_1 Af\lambda < 2\lambda f$, and $A(2f+1) < 4$. Hence

$$(1-g)\gamma(r,\alpha) - I \;\le\; \Big(\frac{27}{4}f + 6\epsilon\Big)fr + 2\lambda f + 4.$$

Finally, $gI \le \frac{9}{8}f \cdot 2fr$, so

$$\gamma(r,\alpha) - I = \frac{(1-g)\gamma(r,\alpha) - I + gI}{1-g} \;\le\; \frac{8}{5}\Big((9f+6\epsilon)fr + 2\lambda f + 4\Big) \;\le\; 16c.$$

□