On Trees, Chains and Fast Transactions in the Blockchain

Aggelos Kiayias*
School of Informatics,
University of Edinburgh
akiayias@inf.ed.ac.uk

Giorgos Panagiotakos*
School of Informatics,
University of Edinburgh
giorgos.pan@ed.ac.uk

October 13, 2016

Abstract

A fundamental open problem in the area of blockchain protocols is whether the Bitcoin protocol is the optimal solution (in terms of efficiency, security) for building a secure transaction ledger. A recently proposed and widely considered alternative is the GHOST protocol which, notably, was proposed to be at the core of Ethereum as well as other recent proposals for improved Bitcoin-like systems. The GHOST variant is touted as offering superior performance compared to Bitcoin (potentially offering block production speed up by a factor of more than 40) without a security loss. Motivated by this, in this work, we study from both a provable security and attack susceptibility point of view the problem of transaction processing time for both GHOST and Bitcoin.

We introduce a new formal framework for the analysis of blockchain protocols that relies on trees (rather than chains) and we showcase the power of the framework by providing a unified description of the GHOST and Bitcoin protocols, the former of which we extract and formally describe. We then prove that GHOST implements a "robust transaction ledger" (i.e., possesses liveness and persistence) and hence it is a provably secure alternative to Bitcoin. Our proof follows a novel methodology which may be of independent interest.

Given this, we then ask whether GHOST is a better alternative. We focus on the liveness property of both Bitcoin and GHOST, i.e., the worst-case transaction confirmation time that can be expected when playing against an adversary. We present a general attack methodology against liveness and we instantiate it with two attacks for Bitcoin and GHOST. We prove (i) our attack for Bitcoin is optimal and (ii) GHOST, when under our attack, performs, in expectation, *worse* than Bitcoin under the optimal attack, for various parameter choices.

With the above results, our work provides a first example of comparative study between different blockchain designs from a provable security perspective.

*Research supported by ERC project CODAMODA #25915 . Part of this work was based in a technical report published in e-print (https://eprint.iacr.org/2015/1019).

Contents

1 Introduction

The popularity of Bitcoin [19] has lead to a surge in the interest about its core protocol that maintains a distributed data structure called the "blockchain." In [10], the core of the Bitcoin protocol was abstracted under the moniker "Bitcoin Backbone" and it was demonstrated to be a powerful tool for solving consensus, [23, 16], in a synchronous, anonymous and Byzantine setting where (unreliable) broadcast is the communication operation available to the participants, (a problem first considered in [2, 20]). In [10], it was shown that the core protocol provably guarantees two properties: (i) *persistence:* if a transaction is reported as stable by one node, then it will be also reported as such by any other honest node of the system, (ii) *liveness:* all honestly generated transactions that are broadcasted are eventually reported as stable by some honest node. This provides a formal framework for proving the security of systems like Bitcoin, since their security can be reduced to the persistence and liveness of the underlying transaction ledger. Furthermore, it provides a way to argue formally about transaction confirmation time since the liveness property is equipped with a delay parameter that specifies the maximum transaction delay that can be caused by an adversary.

Naturally, implementing a robust transaction ledger may be achieved in various other ways, and it is a fundamental open question of the area whether the Bitcoin protocol itself is an "optimal" implementation of a robust transaction ledger. Indeed, many researchers have challenged various aspects of the Bitcoin system and they have proposed modifications in its core operation. Some of the modified systems maintain the protocol structure but modify the hard-coded parameters (like the block generation rate) or the basic primitives, e.g., the way proof of work is performed (a number of alternative proof of work implementations have been proposed using functions like scrypt [24], lyra2 [26] and others). However, more radical modifications are possible that alter the actual operation of the protocol.

One of the most notable such variants is the `GHOST` protocol, which was suggested by Sompolinsky and Zohar in [27]. After the initial suggestion many cryptocurrencies using variants of the `GHOST` rule were proposed and implemented. The most popular among them, Ethereum [7] has received substantial research attention [15, 14, 4, 25, 12, 21]. Ethereum is essentially a Bitcoin-like system where transaction processing is Turing-complete and thus it can be used to implement any public functionality in a distributed way. Bitcoin-NG [8] is another popular Bitcoin-like system relying on `GHOST` that separates blocks in two categories, namely key blocks and microblocks, reflecting the fact that transaction serialization and leader election may be separated.

Unfortunately, the security analysis of [27] is not as general as [10] (e.g., their attacker does not take advantage of providing conflicting information to different honest parties), while the analysis of [10] does not carry to the setting of `GHOST`. This is because the `GHOST` rule is a natural, albeit radical, reformulation of how each miner determines the main chain. In `GHOST`, miners adopt blocks in the structure of a *tree*. Note that in both Bitcoin and `GHOST` one can consider parties collecting all mined blocks in a tree data structure. However, while in Bitcoin the miners would choose the most difficult chain as the main chain, in `GHOST`, they will determine the chain by greedily following the "heaviest observed subtree." This means that for the same subtree, a Bitcoin miner and a `GHOST` miner may choose a completely different main chain. Furthermore, it means that the difficulty of the main chain of honest parties does not necessarily increase monotonically (it may decrease at times) and thus a fundamental argument (namely that blockchains monotonically increase) that made the analysis of [10] possible, does not hold anymore.

Our Results. We propose a new analysis framework for blockchain protocols focusing on trees of blocks as opposed to chains as in [10]. Our framework enables us to argue about random variables on the trees of blocks that are formed by the participants. In our framework, we can express concepts like a node being **d-dominant**, which means that the block corresponding to that node would be

preferred by a margin of d compared to other sibling nodes according to a specified weight measure. This actually enables us to unify the description of Bitcoin and GHOST by showing they obey the same rule, but simply for a different weight measure.

Using our framework we then provide a first formal security proof of the GHOST rule for blockchain protocols. Specifically, we prove that GHOST is a robust transaction ledger that satisfies liveness and persistence. We achieve this result, by a new methodology, that reduces the properties of the robust transaction ledger to a single lemma, that we call the *fresh block lemma* and is informally stated as follows.

> *Fresh Block Lemma.* (Informally) At any point of the execution and for any past sequence of s consecutive rounds, there exists an honest block mined in these rounds, that is contained in the chain of any honest player from this point on.

As we demonstrate, the fresh block lemma is a powerful tool in the presence of an adversary: we show easily that the properties of the robust transaction ledger reduce to it in a black-box fashion. This provides an alternative proof methodology for establishing the properties of a robust transaction ledger compared to [10], cf. also [13], who reduced the properties of the robust transaction ledger to three other properties called common prefix, chain quality and chain growth, and may be of independent interest as it could be applicable to other blockchain variants.

Having established the provable security of GHOST, we then ask whether it is a more efficient alternative to bitcoin. The focus for this is the liveness property, and more specifically the delay parameter that specifies the worst-case confirmation time that can be caused by an adversary. We present a general attack methodology for attacking transaction confirmation time. Our attack method has three stages: (i) the attack preparation stage, (ii) the transaction denial stage and (iii) the blockchain retarder stage. In the attack preparation stage, our attacker prepares the attack and waits for the transaction that she dislikes to appear in the network (e.g., the attacker may mine a private chain or may interfere with block adoption of the honest nodes to be at an advantageous position). When the disliked transaction appears, the attacker moves to the transaction denial phase where she tries to prevent honest nodes from adopting it. At any moment, the attacker may switch to the third phase where she gives up on preventing the honest nodes from adopting the transaction and tries to slow down the blockchain growth so that the confirmation time might be extended. Using this template, we present two attacks for Bitcoin and GHOST respectively.

We prove that our attack for Bitcoin is optimal in the sense that the cumulative distribution of delay in Bitcoin transaction processing time, when under our attack, dominates the delay that may be caused by any other attack. It follows that our attack can be used as a yardstick to show whether a protocol can improve blockchain liveness compared to Bitcoin in the following manner: a protocol will enjoy provably better liveness than Bitcoin provided that the delay an arbitrary attacker can cause against the new blockchain protocol is strictly bounded by the delay caused by our attacker against Bitcoin.

Interestingly, the attack we present for GHOST breaks this barrier. The attack is more involved than the corresponding for Bitcoin, as it exploits the way that honest nodes pick the main chain in a way that is intrinsic to the GHOST rule. This enables a powerful blockchain retarder phase that slows down chain growth. We prove that the GHOST protocol, under our attack, is outperformed by Bitcoin (when subjected to the optimal attack) for a wide range of parameter settings and numbers of blocks that one wishes to wait in order to confirm a transaction. Given the practical relevance of our results, we also verify our results experimentally. We observe that the gap between the two protocols in favor of Bitcoin becomes particularly significant when the number of blocks required

for confirmation is very high (at the level that is required by various exchanges[1]).

Given that the main claims for the GHOST protocol is its alleged superior capability to allow faster transactions compared to Bitcoin, cf. [27], it is important to reflect that this is untrue in the provable sense within our model. We note that, in order to compare "apples to apples," we compare the two protocols, GHOST and Bitcoin, using the same equally accelerated block production rate. Comparing the two at an equal rate is justified from our provable security analysis for the persistence property which does not enable us to show a security advantage of GHOST over Bitcoin for accelerated rates; (put differently, Bitcoin does not appear to lose security at a higher rate than GHOST when accelerated.[2])

We remark that a number of variants of GHOST have been considered in Ethereum (see [7]) with the one currently selected being termed "uncles-only GHOST" for 7 generations. We remark that recently, in [11], it was suggested that the actual implementation still uses a variant of Bitcoin and resembles GHOST only in the reward mechanism. In any case, we show that our transaction confirmation time attack against GHOST may easily extend to variants such as uncles-only GHOST (albeit with a slightly milder effect). There are other ways to modify GHOST that can be considered (e.g., [17]) and these may also be cast and analyzed both from a provable perspective in our framework as well as from an attack potential perspective using our attack template.

With the above results, our work provides a first example of comparative study between different blockchain designs from a provable security perspective. We believe that this a fruitful direction which may lead to either future improvements of blockchain protocols or optimality results for the protocols that are already known.

On the generality of the adversarial model. The adversarial model we adopt in this work is the one proposed by Garay et al. [10]. This model is quite general in the sense that, it can captures many attack models that were proposed in the literature. For example, it captures the double spending attacker of [19], the block withholding attacker of [9] (which can be simulated because the adversary can change the order that messages arrive for each honest player) and the eclipse attacker of [6] where the communication of a portion of the honest nodes in the network is completely controlled (eclipsed) by the adversary (this can be simulated by simply considering the eclipsed nodes to be controlled by the adversary and having the adversary honestly execute their program while dropping their incoming messages). For a quantitative analysis of these attacks the reader is referred to [11].

Limitations and directions for future research. Our analysis is in the standard Byzantine model where parties fall into two categories, those that are honest (and follow the protocol) and those that are dishonest and may deviate in an arbitrary (and coordinated) fashion as dictated by the adversary. It is an interesting direction for future work to consider the rational setting where all parties wish to optimize a certain utility function. Designing suitable incentive mechanisms, for instance see [18] for a suggestion related to the GHOST protocol, or examining the requirements for setup assumptions, cf. [1], are related important considerations. Our analysis is in the static setting, i.e., we do not take into account the fact that parties change dynamically and that the protocol calibrates the difficulty of the POW instances to account for that; we note that this may open the possibility for additional attacks, say [3], and hence it is an important point for consideration and future work. While we discover an optimal attack against the liveness property for bitcoin,

[1] Kraken and Poloniex are currently the biggest Ethereum exchanges. Kraken initially had used 6000 blocks for confirmation time, while Poloniex 375 blocks. See Figure 7 for experimental results at this level.

[1] Currently the Ethereum Frontier reports an average of about 14 seconds, cf. https://etherchain.org; the 12 seconds rate was discussed by Buterin in [5]. In contrast, Bitcoin block generation rate is 10 minutes.

[2] We note that even though the analysis of [27] suggests that there is an advantage, their analysis is performed in a much more restricted attack model than ours.

the provable security bound for the delay in the liveness property of GHOST is not matched by an attacker. Even though, we demonstrate that our GHOST attacker causes higher delays than Bitcoin for most choices of the parameters, it does not match the worst case provable bound, something that means that the bound might be lowered (or alternatively the attack may be improved). Finally, it is interesting to consider our results in more general models such as the semi-synchronous model of [22].

Organization. In section 2 we overview the model that we use for expressing the protocols and the theorems regarding the security properties. In section 3 we introduce our new tree-based framework. Then, in section 4 we present our security analysis of an abstraction of the GHOST protocol that demonstrates it is a robust transaction ledger in the static setting. In section 5 we present our liveness attacks against Bitcoin and GHOST variants, we prove the optimality of the attack against Bitcoin and we compare the two attacks by performing simulations for various parameter choices.

2 Preliminaries and the GHOST Backbone protocol

2.1 Model

For our model we adopt the abstraction proposed in [10]. Specifically, in their setting, called the q-bounded setting, synchronous communication is assumed and each party is allowed q queries to a random oracle. The network supports an anonymous message diffusion mechanism that is guaranteed to deliver messages of all honest parties in each round. The adversary is rushing and adaptive. Rushing here means that in any given round he gets to see all honest players' messages before deciding his own strategy. However, after seeing the messages he is not allowed to query the hashing oracle again in this round. In addition, he has complete control of the order that messages arrive to each player. The model is "flat" in terms of computational power in the sense that all honest parties are assumed to have the same computational power while the adversary has computational power proportional to the number of players that it controls.

The total number of parties is n and the adversary is assumed to control t of them (honest parties don't know any of these parameters). Obtaining a new block is achieved by finding a hash value that is smaller than a difficulty parameter D. The success probability that a single hashing query produces a solution is $p = \frac{D}{2^\kappa}$ where κ is the length of the hash. The total hashing power of the honest players is $\alpha = pq(n-t)$, the hashing power of the adversary is $\beta = pqt$ and the total hashing power is $f = \alpha + \beta$. A number of definitions that will be used extensively are listed below.

Definition 1. A round is called:

- successful if at least one honest player computes a solution in this round.
- uniquely successful if exactly one honest player computes a solution in this round.

Definition 2. In an execution blocks are called:

- honest, if mined by an honest party.
- adversarial, if mined by the adversary.

Definition 3. (chain extension) We will say that a chain $\mathcal{C}'$ extends another chain $\mathcal{C}$ if a prefix of $\mathcal{C}'$ is a suffix of $\mathcal{C}$.

In [10], a lower bound to the probabilities of two events, that a round is successful or that is uniquely successful (defined bellow), was established and denoted by $\gamma_{\mathsf{u}} = \alpha - \alpha^2$. While this bound is sufficient for the setting of small f, here we will need to use a better lower bound to the probability of those events, denoted by γ, and with value approximately $\alpha e^{-\alpha}$ (see Appendix). Observe that $\gamma > \gamma_{\mathsf{u}}$.

2.2 The GHOST Backbone Protocol

In order to study the properties of the core Bitcoin protocol, the term *Backbone Protocol* was introduced in [10]. On this level of abstraction we are only interested on properties of the blockchain, independently from the data stored inside the blocks. The main idea of the Bitcoin Backbone is that honest players, at every round, receive new chains from the network and pick the longest valid one to mine. Then, if they obtain a new block (by finding a small hash), they broadcast their chain at the end of the round. For more details we refer to [10, Subsection 3.1].

The same level of abstraction can also be used to express the GHOST protocol. The GHOST Backbone protocol, as presented in [27], is based on the principle that blocks that do not end up in the main chain, should also matter in the chain selection process. In order to achieve this, players store a tree of all mined blocks they have received, and then using the greedy heaviest observed subtree (GHOST) rule, they pick which chain to mine.

Algorithm 1 The chain selection algorithm. The input is a block tree T. The $|\cdot|$ operator corresponds to the number of nodes of a tree.

function GHOST(T)
  $B \leftarrow GenesisBlock$
  if $children_T(B) = \emptyset$ **then**
    return $\mathcal{C} = (GenesisBlock, ..., B)$
  else
    $B \leftarrow argmax_{c \in children_T(B)} |subtree_T(c)|$
    return GHOST($subtree_T(B)$)
  end if
end function

At every round, players update their tree by adding valid blocks sent by other players. The same principle as Bitcoin applies; for a block to be added to the tree, it suffices to be a valid child of some other tree block. The adversary can add blocks anywhere he wants in the tree, as long as they are valid. Again, as on Bitcoin, players try to extend the chains they choose by one or more blocks. Finally, in the main function, a tree of blocks is stored and updated at every round. If a player updates his tree, he broadcasts it to all other players.

The protocol is also parameterized by three external functions $V(\cdot)$, $I(\cdot)$, $R(\cdot)$ which are called: the input validation predicate, the input contribution function, and the chain reading function, respectively. $V(\cdot)$ dictates the structure of the information stored in each block, $I(\cdot)$ determines the data that players put in the block they mine, $R(\cdot)$ specifies how the data in the blocks should be interpreted depending on the application.

2.3 Security Properties

In [10, Definitions 2&3] two crucial security properties of the Bitcoin backbone protocol were considered, the common prefix and the chain quality property. The common prefix property ensures

Algorithm 2 The GHOST backbone protocol, parameterized by the *input contribution function* $I(\cdot)$ and the *reading function* $R(\cdot)$. $\mathbf{x}_{\mathcal{C}}$ is the vector of inputs of all block in chain $\mathcal{C}$.

$T \leftarrow GenesisBlock$ ▷ T is a tree.
$state \leftarrow \varepsilon$
$round \leftarrow 0$
while TRUE **do**
  $T_{new} \leftarrow \mathsf{update}(T, \text{blocks found in } \textsc{Receive}())$
  $\tilde{\mathcal{C}} \leftarrow \mathtt{GHOST}(T_{new})$
  $\langle state, x\rangle \leftarrow I(state, \tilde{\mathcal{C}}, round, \textsc{Input}(), \textsc{Receive}())$
  $\mathcal{C}_{\mathsf{new}} \leftarrow \mathsf{pow}(x, \tilde{\mathcal{C}})$
  if $\tilde{\mathcal{C}} \neq \mathcal{C}_{\mathsf{new}}$ or $T \neq T_{\mathsf{new}}$ **then**
    $T \leftarrow \mathsf{update}(\mathrm{T}_{\mathsf{new}}, \mathrm{head}(\mathcal{C}_{\mathsf{new}}))$
    $\textsc{Broadcast}(\mathrm{head}(\mathcal{C}_{\mathsf{new}}))$
  end if
  $round \leftarrow round + 1$
  if $\textsc{Input}()$ contains READ **then**
    write $R(\mathbf{x}_{\mathcal{C}})$ to $\textsc{Output}()$
  end if
end while

that two honest players have the same view of the blockchain if they prune a small number of blocks from the tail. On the other hand the chain quality property ensures that honest players chains' do not contain long sequences of adversarial blocks. These properties are defined as predicates over the random variable formed by the concatenation of all parties views' denoted by $\textsc{view}^{H(\cdot)}_{\Pi,\mathcal{A},\mathcal{Z}}(\kappa, q, z)$.

Definition 4 (Common Prefix Property)**.** The common prefix property Q_{cp} with parameter $k \in \mathbb{N}$ states that for any pair of honest players P_1, P_2 maintaining the chains $\mathcal{C}_1, \mathcal{C}_2$ in $\textsc{view}^{H(\cdot)}_{\Pi,\mathcal{A},\mathcal{Z}}(\kappa, q, z)$, it holds that

$$\mathcal{C}_1^{\lceil k} \preceq \mathcal{C}_2 \text{ and } \mathcal{C}_2^{\lceil k} \preceq \mathcal{C}_1.$$

Definition 5 (Chain Quality Property)**.** The chain quality property Q_{cq} with parameters $\mu \in \mathbb{R}$ and $\ell \in \mathbb{N}$ states that for any honest party P with chain $\mathcal{C}$ in $\textsc{view}^{H(\cdot)}_{\Pi,\mathcal{A},\mathcal{Z}}(\kappa, q, z)$, it holds that for any ℓ consecutive blocks of $\mathcal{C}$ the ratio of adversarial blocks is at most μ.

These two properties were shown to hold for the Bitcoin backbone protocol. Formally, in [10, Theorems 9&10] the following were proved:

Theorem 6. *Assume $f < 1$ and $\gamma_{\mathsf{u}} \geq (1+\delta)\lambda\beta$, for some real $\delta \in (0,1)$ and $\lambda \geq 1$ such that $\lambda^2 - f\lambda - 1 \geq 0$. Let $\mathcal{S}$ be the set of the chains of the honest parties at a given round of the backbone protocol. Then the probability that $\mathcal{S}$ does not satisfy the common-prefix property with parameter k is at most $e^{-\Omega(\delta^3 k)}$.*

Theorem 7. *Assume $f < 1$ and $\gamma_{\mathsf{u}} \geq (1+\delta)\lambda\beta$ for some $\delta \in (0,1)$. Suppose $\mathcal{C}$ belongs to an honest party and consider any ℓ consecutive blocks of $\mathcal{C}$. The probability that the adversary has contributed more than $(1-\frac{\delta}{3})\frac{1}{\lambda}\ell$ of these blocks is less than $e^{-\Omega(\delta^2\ell)}$.*

Robust public transaction ledger. In [10] the robust public transaction ledger primitive was presented. It tries to capture the notion of a book where transactions are recorded, and it is used to implement Byzantine Agreement in the honest majority setting.

A *public transaction ledger* is defined with respect to a set of valid ledgers $\mathcal{L}$ and a set of valid transactions $\mathcal{T}$, each one possessing an efficient membership test. A ledger $\mathbf{x} \in \mathcal{L}$ is a vector of sequences of transactions $\mathsf{tx} \in \mathcal{T}$. Each transaction tx may be associated with one or more *accounts*, denoted $a_1, a_2, \ldots$ Ledgers correspond to chains in the backbone protocols. An oracle Txgen is allowed in the protocol execution that generates valid transactions (this represents transactions that are issued by honest parties). For more details we refer to [10].

Definition 8. A protocol Π implements a *robust public transaction ledger* in the q-bounded synchronous setting if it satisfies the following two properties:

- *Persistence:* Parameterized by $k \in \mathbb{N}$ (the "depth" parameter), if in a certain round an honest player reports a ledger that contains a transaction tx in a block more than k blocks away from the end of the ledger, then tx will always be reported in the same position in the ledger by any honest player from this round on.
- *Liveness:* Parameterized by $u, k \in \mathbb{N}$ (the "wait time" and "depth" parameters, resp.), provided that a transaction either (i) issued by Txgen, or (ii) is neutral, is given as input to all honest players continuously for u consecutive rounds, then there exists an honest party who will report this transaction at a block more than k blocks from the end of the ledger.

These two properties were shown to hold for the ledger protocol Π_{PL} build on top of the Bitcoin backbone protocol. Formally, in [10, Lemma 15&16] the following were proved:

Lemma 9 (Persistence)**.** *Suppose $f < 1$ and $\gamma_{\mathsf{u}} \geq (1+\delta)\lambda\beta$, for some real $\delta \in (0,1)$ and $\lambda \geq 1$ such that $\lambda^2 - f\lambda - 1 \geq 0$. Protocol Π_{PL} satisfies Persistence with probability $1 - e^{-\Omega(\delta^3 k)}$, where k is the depth parameter.*

Lemma 10 (Liveness)**.** *Assume $f < 1$ and $\gamma_{\mathsf{u}} \geq (1+\delta)\lambda\beta$, for some $\delta \in (0,1)$, $\lambda \in [1,\infty)$ and let $k \in \mathbb{N}$. Further, assume oracle Txgen is unambiguous. Then protocol Π_{PL} satisfies Liveness with wait time $u = 2k/(1-\delta)\gamma_{\mathsf{u}}$ and depth parameter k with probability at least $1 - e^{-\Omega(\delta^2 k)}$.*

3 A unified description of Bitcoin and GHOST backbone

Next, we introduce our new analysis framework for backbone protocols that is focusing on trees of blocks and we show how the description of the Bitcoin and GHOST can be unified. In this model, every player stores all blocks "he hears" on a tree, starting from a pre-shared block called the *Genesis* (or v_{root}) block. This is the model where GHOST was initially described. Bitcoin, and other possible backbone variants, can also be seen in this model and thus a unified language can be built. We first define block trees (or just trees) that capture the knowledge of honest players (regarding the block tree on different moments at every round).

Definition 11. We denote by T_r^P (resp. $T_r^{\exists}$) the tree that is formed from the blocks that player P (resp. at least one honest player) has received until the beginning of round r. Similarly, T_r^+ is the tree that contains $T_r^{\exists}$ and also includes all blocks mined by honest players at round r. For any tree T and block $b \in T$, we denote by $T(b)$ the subtree of T rooted on b.

Notice that, due to the fact that broadcasts of honest players always succeed, blocks in T_r^+ are always in T_{r+1}^P. Thus for every honest player P it holds that:

$$T_r^P \subseteq T_r^{\exists} \subseteq T_r^+ \subseteq T_{r+1}^P$$

Intuitively, heavier trees represent more proof of work. However, there is more than one way to define the weight of a tree. For example, in Bitcoin the heaviest tree is the longest one. On the other hand, for GHOST a heavy tree is one with many nodes. To capture this abstraction we condition our definitions on a norm w that assigns weights on trees. This norm will be responsible for deciding which tree has more proof of work, and thus which tree is favored by the chain selection rule. We choose to omit w from the notation since it will always be clear from the context which norm we use.

Definition 12. Let w be a norm defined on trees. For any tree T let $siblings(v)$ denote the set of nodes in T that share the same parent with v. Then node v is **d-dominant** in T (denoted by $\text{Dom}_T(v, d)$) iff

$$w(T(v)) \geq d \wedge \forall v' \in siblings(v) : w(T(v)) \geq w(T(v')) + d$$

The chain selection rule in the Bitcoin protocol can be described using the notion of the d-dominant node. Let $w(T)$ be the height of some tree T. Each player P, starting from the root of his T_r^P tree, greedily decides on which block to add on the chain by choosing one of its 0-dominant children and continuing recursively[3] (ties are broken based on time-stamp, or based on which block was received first). Interestingly, the GHOST selection rule can also be described in exactly the same way by setting w to be the number of nodes of the tree. Thus we have a unified way for describing the chain selection rule in both protocols. Building upon this formalism we can describe the paths that fully informed honest players may choose to mine at round r (denoted by $\text{HonestPaths}(r)$) in a quite robust way, thus showcasing the power of our notation.

$$\text{HonestPaths}(r) = \{p = v_{\text{root}} v_1 \dots v_k | p \text{ is a root-leaf path in } T_r^{\exists} \wedge \forall i \in \{1, .., k\} \ \text{Dom}_{T_r^{\exists}}(v_i, 0)\}$$

We conclude this section by presenting two crucial properties that both the Bitcoin and GHOST backbones satisfy. The first property states that by broadcasting k blocks the adversary can decrease the dominance of some block at most by k. Intuitively, it tells us if the adversary's ability to mine new blocks is limited, then his influence over the block tree is also limited. On the other hand, the second property states that uniquely successful rounds increase the dominance only of nodes in the path from the root to the new block.

We will use the term node and block interchangeably from now on.

Proposition 13. For the Bitcoin and GHOST backbones protocols it holds that:

- If the adversary publishes $k \leq d$ blocks at round $r-1$ then for every block $v \in T_{r-1}^+$ it holds that $\text{Dom}_{T_{r-1}^+}(v, d)$ implies $\text{Dom}_{T_r^{\exists}}(v, d-k)$.
- If r is a uniquely successful round and the newly mined block extends a path in $\text{HonestPaths}(r)$, then for any block v in $T_r^{\exists}$ it holds that: $\text{Dom}_{T_r^{\exists}}(v, d)$ implies $\text{Dom}_{T_r^+}(v, d+1)$ if and only if v is in the path from v_{root} to the new block.

Proof. The lemma stems from the fact that adding only one block in the tree reduces or increases the dominance of some block by at most 1. For the first bullet, adding k blocks one by one, implies that the dominance of any node will reduce or increase by at most k. For the second bullet, notice that dominance increases only for blocks that get heavier. The only blocks that get heavier in this case are the ones in the path from the root to the newly mined block. Since these blocks are in $\text{HonestPaths}(r)$, they are at least 0-dominant and so their dominance will further increase. Furthermore, the newly mined block is 1-dominant since it does not have any siblings.

□

[3] This is exactly algorithm 1 with a minor modification. At line 6 the subtree T that is chosen maximizes $w(T)$.

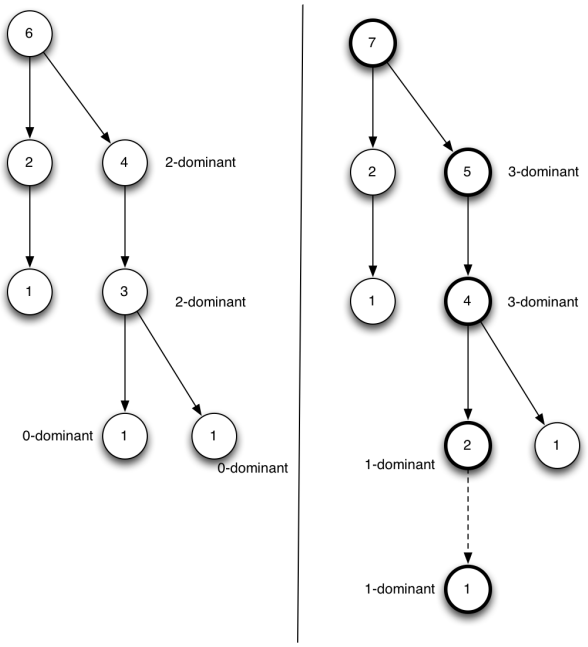

Figure 1: An example of the change in dominance after a uniquely successful round. The only nodes which increase their dominanceare the ones in the path from the root to the newly mined block as stated in Proposition 13.

4 Security Analysis and Applications

Next, we prove that the `GHOST` backbone protocol is sufficient to construct a robust transaction ledger. From now on we assume that $w(T)$ is the total number of nodes of tree T.

4.1 The Fresh Block Lemma

In [10], it was shown that the Bitcoin Backbone satisfies two main properties: common prefix and chain quality. However, another fundamental property needed for their proof, is that the chains of honest players grow at least at the rate of successful rounds. This does not hold for `GHOST`. The reason is that, if an honest player receives a chain that is heavier than the one he currently has, he will select it, *even if it is shorter*. To reflect these facts, we develop an argument that is a lot more involved and leads to a power lemma that we call the "fresh block lemma."

First, we introduce a new notion, that of a path that all of its nodes are dominant up to a certain value. Intuitively, the more dominant a path is, the harder it gets for the adversary to stop honest players from choosing it.

Definition 14. ($\mathrm{p}_{\mathrm{dom}}(r,d)$) For $d>0$, $\mathrm{p}_{\mathrm{dom}}(r,d)$ is the longest path $p = v_{\mathrm{root}}v_1 \dots v_k$ in T_r^+ s.t.

$$p \neq v_{\mathrm{root}} \wedge \forall i \in \{1,\dots,k\}: \ \mathrm{Dom}_{T_r^+}(v_i,d)$$

If no such path exists $\mathrm{p}_{\mathrm{dom}}(r,d) = \perp$.

Note that the dominant path $\mathrm{p}_{\mathrm{dom}}(r,d)$, if it is not $\perp$, will be unique (this stems from the requirement that $d>0$).

In the next lemma, we show that unless the number of blocks the adversary broadcasts in a round interval is at least as big as the number of uniquely successful rounds that have occurred, an

honest block mined in one of these rounds will be deep enough in the chains of honest players. More specifically, for any sequence of m (not necessarily consecutive) uniquely successful rounds starting at some round r', no matter the strategy of the adversary, at round r there will be at least one honest block in $\mathrm{p}_{\mathrm{dom}}(r, m-k)$ where k is the number of adversarial blocks that have been released during rounds $[r'-1, r-1]$.

Lemma 15. *Let $r_1, .., r_m$ be uniquely successful rounds from round r' until round r. If the adversary broadcasts $k < m$ blocks from round $r'-1$ until round $r-1$, then there exists an honest block b, mined in one of the rounds $r_1, .., r_m$ such that b is in $\mathrm{p}_{\mathrm{dom}}(r, m-k)$.*

Proof. We are first going to prove two preliminary claims that show the effect of a uniquely successful round to $\mathrm{p}_{\mathrm{dom}}$. The first claim shows that if a uniquely successful round s is not compensated accordingly by the adversary, a newly mined block will be forced into $\mathrm{p}_{\mathrm{dom}}(s, 1)$.

Claim 1. Let round s be a uniquely successful round and b be the honest block mined at round s. If the adversary does not broadcast any block at round $s-1$ then $b \in \mathrm{p}_{\mathrm{dom}}(s, 1)$.

Proof of Claim. First, notice that since the adversary does not broadcast any block it holds that for any honest player P, $T_s^{\exists} = T_s^P$. Therefore, all nodes in the path from v_{root} to the parent of b are at least 0-dominant in $T_s^{\exists}$ and thus this path is in HonestPaths(s). Since s is uniquely successful, all conditions of the second bullet of Proposition 13 are met, and thus it is implied that all nodes up to the newly mined block in T_s^+ are 1-dominant. It follows that $b \in \mathrm{p}_{\mathrm{dom}}(s, 1)$. ⊣

The second claim shows the effect of a uniquely successful round s to an existing $\mathrm{p}_{\mathrm{dom}}(s-1, d)$ path. Notice that if the adversary broadcasts less than d blocks the same nodes continue to be at least 1-dominant in the following round.

Claim 2. Let round s be a uniquely successful round, b be the honest block mined at round s and $\mathrm{p}_{\mathrm{dom}}(s-1, d) \neq \perp$. If the adversary broadcasts (i) $k < d$ blocks at round $s-1$ then $\mathrm{p}_{\mathrm{dom}}(s-1, d) \subseteq \mathrm{p}_{\mathrm{dom}}(s, d+1-k)$, (ii) $k = d$ blocks at round $s-1$ then either $b \in \mathrm{p}_{\mathrm{dom}}(s, 1)$ or $\mathrm{p}_{\mathrm{dom}}(s-1, d) \subseteq \mathrm{p}_{\mathrm{dom}}(s, 1)$ and b is a descendant of the last node in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$.

Proof of Claim. There are two cases. In the first case suppose the adversary broadcasts $k < d$ blocks. Then, according to the first bullet of Proposition 13, the adversary can lower the dominance in $T_s^{\exists}$ of nodes in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$ by at most k. Thus $\mathrm{p}_{\mathrm{dom}}(s-1, d)$ will be a prefix of all the chains in HonestPaths(s). But because s is a uniquely successful round, the dominance in T_s^+ of all nodes in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$ will increase by one. Therefore $\mathrm{p}_{\mathrm{dom}}(s-1, d) \subseteq \mathrm{p}_{\mathrm{dom}}(s, d+1-k)$ and b will be a descendant of the last node in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$.

In the second case suppose the adversary broadcasts $k = d$ blocks. If he does not broadcast all of these blocks to reduce the dominance in $T_s^{\exists}$ of the nodes in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$, then $\mathrm{p}_{\mathrm{dom}}(s-1, d)$ will be a prefix of all the chains in HonestPaths(s) and as in the previous case, $\mathrm{p}_{\mathrm{dom}}(s-1, d) \subseteq \mathrm{p}_{\mathrm{dom}}(s, d+1-k)$ and b will be a descendant of the last node in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$.

Otherwise the adversary will reduce the dominance in $T_s^{\exists}$ of at least one node in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$ to zero. If b is a descendant of the last node in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$, then all nodes in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$ will be 1-dominant in T_s^+ and $\mathrm{p}_{\mathrm{dom}}(s-1, d) \subseteq \mathrm{p}_{\mathrm{dom}}(s, 1) = \mathrm{p}_{\mathrm{dom}}(s, d+1-d)$. If b is not a descendant of the last node in $\mathrm{p}_{\mathrm{dom}}(s-1, d)$, then for the player P that mined this block it holds that $T_s^P = T_s^{\exists}$, because he would have not mined a chain that does not contain $\mathrm{p}_{\mathrm{dom}}(s-1, d)$ at round s otherwise. Therefore, P at round s was mining a chain that belonged to HonestPaths(s, v_{root}) and thus all nodes in the chain are at least 0-dominant in $T_s^{\exists}$. But because s is a uniquely successful round the dominance of all nodes in the chain that b belongs to will increase by one and thus $b \in \mathrm{p}_{\mathrm{dom}}(s, 1)$. ⊣

Let b_i denote the honest block mined at round r_i. Let us assume that $r = r_m$. We are going to prove the lemma using induction on the number of uniquely successful rounds m.

For the base case suppose $m = 1$. The adversary does not broadcast any block until round $r_1 - 1$ and from the first claim $b_1 \in \mathrm{p_{dom}}(r_1, 1)$. Thus the base case is proved. Suppose the lemma holds for $m-1$ uniquely successful rounds and let k_1 be the number of blocks that the adversary broadcasts in the round interval $[r' - 1, r_{m-1} - 1]$. We have two cases.

(First case) $k_1 = m - 1$ and the adversary broadcasts no blocks in the rest of the rounds. From the first claim it follows that $b_m \in \mathrm{p_{dom}}(r_m, 1)$.

(Second case) $k_1 < m - 1$ and from the induction hypothesis there exist blocks $b'_1, ..., b'_{m-1-k_1}$ mined by honest players at the uniquely successful rounds $r_1, .., r_{m-1}$ where $b'_i \in \mathrm{p_{dom}}(r_{m-1}, i)$. Let k_2 be the number of blocks that the adversary broadcasts until round $r_m - 2$ and k_3 the number of blocks he broadcasts at round $r_m - 1$. If $k_2 = m - 1$ then again from the first claim it follows that $b_m \in \mathrm{p_{dom}}(r_m, 1)$. If $k_2 < m - 1$ then if $k_3 + k_2 = m - 1$ then from the second claim either $b_m \in \mathrm{p_{dom}}(r_m, 1)$ or $b'_{m-1-k_1} \in \mathrm{p_{dom}}(r_m, 1)$. If $k_3 + k_2 < m - 1$ then again from the second claim at round r_m, $b'_i \in \mathrm{p_{dom}}(r_m - 1, i)$ for i in $\{k_2 + k_3 + 1, .., m - 1 - k_1\}$ and either $b'_{k_2+k_3}$ is in $\mathrm{p_{dom}}(r_m, 1)$ or b_m is in $\mathrm{p_{dom}}(r_m, 1)$. This completes the induction proof.

We proved that if $k_4 < m$ is the number of blocks the adversary broadcasts until round $r_m - 1$, then there exists honest blocks $b'_1, .., b'_{m-k_4}$ s.t. b'_i is in $\mathrm{p_{dom}}(r_m, i)$. Now in the case $r > r_m$, let $k_5 < m - k_4$ be the number of blocks the adversary broadcasts in the remaining rounds. The lemma follows easily from the second claim.

Remark 1. Let $r_1, .., r_m$ be uniquely successful rounds up to round r and the honest block mined at round r_1 be in $\mathrm{p_{dom}}(r_1, 1)$. If the adversary broadcasts $k < m$ blocks from round r_1 until round $r - 1$, then there exists an honest block b mined in one of the rounds $r_1, .., r_m$ such that b in $\mathrm{p_{dom}}(r, m - k)$. (to see why the remark holds notice that that blocks that the adversary broadcasts before round r_1 affect only the dominant path at round r_1, and not at the following rounds)

□

The fresh block lemma is stated next. Informally, it states that at any point in time, in any past sequence of s consecutive rounds, at least one honest block was mined and is permanently inserted in the chain that every honest player adopts, with overwhelming probability on s.

Lemma 16. (Fresh Block Lemma) *Assume $\gamma \geq (1 + \delta)\beta$, for some real $\delta \in (0, 1)$ and $f < 1$. Then, for all $s \in \mathbb{N}$ and $r \geq s$ it holds that there exists a block mined by an honest player on and after[4] round $r - s$, that is contained in the chain which any honest player adopts on and after round r with probability $1 - e^{-\Omega(\delta^2 s)}$.*

Proof sketch. The difficulty of proving this lemma stems from the fact that in `GHOST`, the chains of honest players are not always strictly increasing. That is, honest players may switch from a longer to a shorter chain. Monotonicity allows us to prove many useful things; for example that the adversary cannot use very old blocks in order to maintain a fork as in [10].

To overcome this difficulty, we first show that whenever the adversary forces honest players to work on a different branch of the block tree, he has to broadcast as many blocks as the ones that where mined on uniquely successful rounds on this branch of the tree. Hence, it is hard for the adversary to force honest players to change branches all the time, and moreover, after s rounds this will be impossible due to the fact that $\gamma \geq (1 + \delta)\beta$. But if all honest players stay on one branch, the blocks near the root of the branch will permanently enter their chains. We show that at least

[4]Throughout this work, we only consider executions that run for a polynomial number of rounds in the security parameter κ.

one of these blocks will be mined by an honest player. By applying this idea in an iterative manner, the lemma follows.

□

Due to space limitations, for the proof of the lemma we refer to the Appendix.

4.2 A robust public transaction ledger

In [10] it is shown how to instantiate the functions V, R, I so that the resulting protocol, denoted by Π_{PL}, built on top of the Bitcoin backbone, implements a robust transaction ledger (see Appendix, Definition 8). In this section we show how we can achieve the same goal, using exactly the same instantiation of V, R, I, but on top of the GHOST backbone. We call the resulting protocol, $\Pi_{\mathsf{PL}}^{\mathtt{GHOST}}$.

Having established that every s rounds a fresh and honest block is inserted in the chain of all players, we are in a position to prove the main properties of a robust transaction ledger. Liveness stems from the fact that after s^2 rounds, s fresh honest blocks mined on this interval will be in the chain of any honest player. On the other hand, Persistence is implied by the fact that all honest players share a freshly mined block. This block will stay in their chains for the subsequent rounds, therefore the history until this block has become persistent. But this block cannot be very deep in the main chain, because the number of blocks succeeding it are limited by the total block generation rate.

Lemma 17 (Liveness)**.** *Assume $\gamma \geq (1+\delta)\beta$, for some $\delta \in (0,1)$ and $f < 1$. Further, assume oracle* Txgen *is unambiguous. Then for all $k \in \mathbb{N}$ protocol $\Pi_{\mathsf{PL}}^{\mathtt{GHOST}}$ satisfies Liveness with wait time $u = k(k+1)$ rounds and depth parameter k with probability at least $1 - e^{-\Omega(\delta^2 k)}$.*

Proof. We prove that assuming all honest players receive as input the transaction tx for at least u rounds, any honest party at round r with chain $\mathcal{C}$ will have tx included in $\mathcal{C}^{\lceil k}$. Let E_i be the event where no block that is in chain $\mathcal{C}$ was computed during rounds $[r-(i+1)k, r-ik]$ by an honest player. For $i \in \{0, ..., k\}$, by Lemma 16, E_i occurs with probability at most $e^{-\Omega(\delta^2 k)}$. Thus, if none of this events holds, it follows that a total of $k+1$ honest blocks, each mined in the respective round interval, are in $\mathcal{C}$ and the "oldest" of them should contain tx. Hence, tx is included in $\mathcal{C}^{\lceil k}$ at round r. By the union bound the probability that $E_0 \vee \ldots \vee E_k$ occurs is at most $e^{-\Omega(\delta^2 k)}$ and the lemma follows.

□

Lemma 18 (Persistence)**.** *Suppose $\gamma \geq (1+\delta)\beta$, for some real $\delta \in (0,1)$ and $f < 1$. Then for all $k \in \mathbb{N}$ protocol $\Pi_{\mathsf{PL}}^{\mathtt{GHOST}}$ satisfies Persistence with probability $1 - e^{-\Omega(\delta^2 k)}$, where k is the depth parameter.*

Proof. Let $\mathcal{C}$ be the chain that an honest player adopts at round r. It is sufficient to show that the head of $\mathcal{C}^{\lceil k}$ has been computed before round $r - k/((1+\delta)f)$, because then from Lemma 16 there exists an honest block computed at least at this round that is on the chain that players adopt from round r and afterwards.

Suppose, for the sake of contradiction, that the head of $\mathcal{C}^{\lceil k}$ is computed after round $r - k/((1+\delta)f)$. The length of $\mathcal{C}$ cannot be greater than the number of solutions Y obtained from the oracle in this amount of rounds. By the Chernoff bound,

$$\Pr[Y \geq (1+\delta)f(k/((1+\delta)f))] \leq e^{-\delta^2 fs/3}.$$

It follows that, with probability $1 - e^{-\delta^2 fs/3}$, $Y < k$ which is a contradiction and thus the lemma follows. □

Corollary 19. *The protocol* $\Pi_{\mathsf{PL}}^{\mathtt{GHOST}}$ *is a robust transaction ledger.*

As a final note, Lemma 16 is sufficient to prove Persistence and Liveness in a black-box way. Compared to the approach of [10], that was further expanded in [13] and [22], only one property, instead of three, of the underlying "backbone" protocol suffices in order to get a robust public transaction ledger in a black-box manner. On the other hand, the three properties described in these works, common-prefix, chain quality and chain growth, also serve as metrics of the efficiency of the underlying mechanism and provide more information than the fresh block lemma.

5 Liveness Attacks

In this section we introduce and analyze a novel class of attacks on the transaction confirmation time of blockchain protocols, and compare the Bitcoin and `GHOST` protocols from the point of view of these attacks. The attacks we consider try to delay as much as possible the confirmation time (Definition 8) of a target transaction and follow the template depicted in Figure 2.

First, at the *attack preparation* phase, the attacker tries to build a potential advantage against honest players, until the time the target transaction tx is broadcast. For instance, the adversary may build a private chain. Next, in the *transaction denial* phase, the attacker tries to delay a new honest block containing tx from entering honest players' chains. When the attacker decides that further delay is improbable, it may proceed to the *blockchain retarder* phase, where it tries to decrease the rate at which the chain containing tx grows. Remember, that the verifier waits until tx is buried k blocks deep in order to accept a transaction. Therefore, by slowing down the rate at which chains grow, confirmation time is further delayed.

In the following sections we will provide two instantiations of this attack template in the case of Bitcoin and `GHOST`. We will prove the attack against Bitcoin to be optimal and we will compare it with the attack against `GHOST`. In this respect, we will prove that the `GHOST` attack slows down transaction confirmation time substantially more than the Bitcoin can be possibly slowed down for a range of parameters. We also experimentally validate this result for a wide set of parameters.

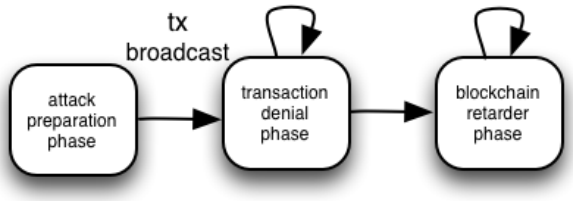

Figure 2: The template of our attacks on confirmation time.

5.1 On Bitcoin Liveness

First, we are going to analyze an attack based on this template against Bitcoin. We are going to present the three phases of the attack separately. In the attack preparation phase, honest players want to ensure that at the point the target transaction is released, they would have the maximum advantage. Advantage here is interpreted as the number of blocks the adversary's secret chain is ahead compared to the honest players' chains. Interestingly, the well known selfish mining attack [9], where the attacker (1) tries to mine a secret chain ahead of the honest players, and when they surpass him, he adopts their chain and (2) only broadcasts blocks from his chain when the honest parties

have mined a block in the same height, has exactly this property. So until tx is released the attacker executes this type of selfish mining attack.

Next, in the transaction denial phase, the attacker tries to extend for as long as possible the secret chain he built in the previous phase, while not helping honest parties catch up. He does this by only mining blocks on his secret chain that do not contain tx, and selectively broadcasting his blocks only when the honest parties have mined a block on the same height. The difference with the previous phase is that the attacker persists on extending the longest chain that does not contain tx, instead of trying to extend the longest one among all chains. For the blockchain retarder phase, the attacker is not going to do anything specific, since, as we will see, it already during the transaction denial phase plays optimally regarding to the chain growth speed. Hence, he can stay at the transaction denial phase until the attacker deems that it is impossible that its private chain is going be selected by any honest player.

In a more compact form our attack goes as follows: (1) the attacker runs a type of selfish mining until tx is released, (2) afterwards it tries to extend its private chain as much as possible, while not including tx on its blocks, and broadcasting parts of its chain only when the honest parties mine blocks on the same height.

In order to show the optimality of our attack we focus our attention to the random variable u_A^{Π}, which is the time a specific transaction tx takes to be confirmed in an execution where honest players run protocol Π against some attacker A. We are going to consider executions on the q-bounded synchronous setting (see Section 2), with the additional restriction that the environment at a specific time t_0 releases the target transaction tx to the network. Therefore, the sample space of u_A^{Π} contains the aforementioned executions against attacker A and that is what we mean anytime some probability involving u_A^{Π} appears in the text from now on. We next show that our attack is optimal against Bitcoin.

Let $A_{\mathtt{BTC}}$ be the attacker we described above and Π_{PL} the "ledger" version of the Bitcoin Backbone mentioned in Section 4.2.

Theorem 20. *For any attacker A' against protocol Π_{PL}, it holds that for any positive k*

$$Pr[u_{A_{\mathtt{BTC}}}^{\Pi_{\mathsf{PL}}} > k] \;\geq\; Pr[u_{A'}^{\Pi_{\mathsf{PL}}} > k]$$

Proof. Let X^e be the sequence of uniquely successful rounds in some execution e and Z^e be the the sequence of the number of blocks mined by the adversary at each round in the same execution. Let $A_{\mathtt{BTC}}$ be the attacker described above and A' be an arbitrary attacker. We are going to first prove that for executions e_1 and e_2 against attackers $A_{\mathtt{BTC}}$ and A' respectively, if $(X^{e_1}, Z^{e_1}) = (X^{e_2}, Z^{e_2})$ then the confirmation time of some transaction tx in e_1, denoted by $u_{A_{\mathtt{BTC}}}^{\Pi_{\mathsf{PL}}}(e_1)$, is at least equal to $u_{A'}^{\Pi_{\mathsf{PL}}}(e_2)$.

Let $\mathcal{C}'$ be the chain of some honest player that confirms transaction tx at e_2, and b_2 be the block of C' that contains tx and was mined at round t_2. Without loss of generality suppose that b_2 was mined by an honest player. Let b_3 be the last honest block in C' before b_2, that was mined at round t_3 and is at height l_3. Then by round t_2 all honest players will have received a chain of length $l_2 = l_3 + \sum_{i=t_3+1}^{t_2-1} X_i^{e_2}$, where X_i is 1 if i is a successful round and 0 otherwise. Since b_2 is honest, the chain the honest miner extended must have had height at least equal to l_2, otherwise no honest player would have selected this chain. Also, all blocks between b_3 and b_2 are by definition adversarial and also descendants of b_3. Hence, it follows that they were mined after round t_3 and are at least $\sum_{i=t_3+1}^{t_2-1} X_i^{e_2}$. Notice also, that round t_3 should be before round t_0, otherwise b_3 would contain tx, which is a contradiction.

Let chain $\mathcal{C}$ be the chain of some honest player that confirms transaction tx at e_1, and b_1 be the block of C that contains tx. We will argue that for attacker $A_{\mathtt{BTC}}$ it holds that if there exists

some round t_3 before t_0 such that the blocks mined by the adversary up to round $t_2 - 1$ are more than the number of uniquely successful rounds in the same period, then b_1 will be mined at least at round t_2. From round t_3 until round $t_0 - 1$, the adversary runs selfish mining and thus at round t_0 he has a secret chain that exceeds the chain of any honest player by $Z_{t_3,t_0-1} - X_{t_3,t_0-1}$ (if positive) blocks, where the random variables represent the blocks mined by the adversary during rounds $[t_3, t_0 - 1]$ and the number of successful rounds respectively. Since $Z_{t_3,t_2-1} \geq X_{t_3,t_2-1}$ it follows that $X_{t_0,t_2-1} \geq Z_{t_3,t_0-1} - X_{t_3,t_0-1} + Z_{t_0,t_2-1}$ and for the remaining rounds the adversary will be able to maintain a secret chain that is as big as the chain of any honest player and thus effectively block tx from entering the chain of any honest player.

Since the condition we argued about holds the sequences (X^{e_2}, Z^{e_2}), it will also hold for (X^{e_1}, Z^{e_1}) an thus b_1 is going to be mined at least at round t_2. Moreover, $A_{\texttt{BTC}}$ never helps honest parties lengthen their chains more than the number of successful rounds. Therefore, the chains of honest players in execution e_1, grow at least as slow as in e_2, and thus $u^{\Pi_{\texttt{PL}}}_{A_{\texttt{BTC}}}(e_1) \geq u^{\Pi_{\texttt{PL}}}_{A'}(e_2)$. It follows that:

$$(X^{e_1}, Z^{e_1}) = (X^{e_2}, Z^{e_2}) \Rightarrow u^{\Pi_{\texttt{PL}}}_{A_{\texttt{BTC}}}(e_1) \geq u^{\Pi_{\texttt{PL}}}_{A'}(e_2) \tag{1}$$

But for any executions e it holds that X^e is independent of the strategy of the adversary. Moreover, we can assume without loss of generality that all adversaries spend all their queries at the random oracle at each round without asking the same query twice. In this case, Z^e is also independent of the strategy of the adversary. Thus for any (x, z), the probability that $(X^e, Z^e) = (x, z)$ in some execution e is the same, independently of the adversary's strategy. Hence:

$$\begin{aligned}
&\forall A', k > 0,\ Pr[u^{\Pi_{\texttt{PL}}}_{A'} > k] = \\
&\sum_{\forall x,z} Pr[u^{\Pi_{\texttt{PL}}}_{A'}(e') > k | (X^{e'}, Z^{e'}) = (x, z)] Pr[(X^{e'}, Z^{e'}) = (x, z)] \leq \\
&\sum_{\forall x,z} Pr[u^{\Pi_{\texttt{PL}}}_{A_{\texttt{BTC}}}(e) > k | (X^{e}, Z^{e}) = (x, z)] Pr[(X^{e}, Z^{e}) = (x, z)] = \\
&Pr[u^{\Pi_{\texttt{PL}}}_{A_{\texttt{BTC}}} > k] \Rightarrow Pr[u^{\Pi_{\texttt{PL}}}_{A_{\texttt{BTC}}} > k] \geq Pr[u^{\Pi_{\texttt{PL}}}_{A'} > k]
\end{aligned}$$

where the probabilities are taken over all executions against attackers A' and $A_{\texttt{BTC}}$ respectively. The first equality follows from the law of total probability. The inequality follows from Inequality 1. □

5.2 On GHOST Liveness

Having described the optimal attack against Bitcoin we turn our attention to GHOST. GHOST was designed to prevent selfish-mining type of attacks. Hence, the attack we described for Bitcoin is going to be much less efficient against GHOST. Instead, a weak point of the protocol is that the length of honest players' chains is not strictly increasing as time goes by. The key idea of our scheme is that the attacker tries to reduce the speed that the chains of honest players grow (also referred to as chain growth speed) and thus is named the GHOST-retarder attack. By succeeding, it can effectively decrease the transaction confirmation time for any observer waiting for a transaction to be k blocks deep in his chain.

The first two stages of our attack are similar to the attack against Bitcoin: the attacker first runs a selfish mining attack against GHOST, trying to build maximum advantage until tx is released, and then tries to extend his secret chain so that the time that honest parties adopt a chain that contains tx is delayed. When honest miners adopt a chain that is longer than his private chain the adversary proceeds to the blockchain retarder phase.

Algorithm 3 The pseudocode of the adversary during the blockchain retarded phase of the attack against the liveness of GHOST with parameter τ (τ must be greater or equal to 3).

$\langle t_H, t_A \rangle \leftarrow \langle 0, 0 \rangle$ ▷ The weight of the competing trees.
Receive new blocks and update the block tree
$\mathcal{C} \leftarrow argmin_{\mathcal{C} \in \text{HonestPaths}} |C|$
Mine head($\mathcal{C}$)
if |blocks mined| $= 0$ **then**
    Restart attack
else
    $B \leftarrow$ newly mined block
    $\langle t_H, t_A \rangle \leftarrow \langle t_H, t_A +$ new adversarial blocks$\rangle$
end if
while $t_H < \tau$ **do**
    Receive new blocks and update the block tree
    Mine blocks on top of B
    $\langle t_H, t_A \rangle \leftarrow \langle t_H +$ new honest blocks, $t_A +$ new adversarial blocks$\rangle$
    if $(t_A \geq t_H)$ and (length of honest subtree $\geq \tau$) **then**
        Broadcast $subtree(B)$
        Restart attack
    end if
end while

In the blockchain retarder phase the attacker exploits the fact that in `GHOST` thin and long trees may have the same or less weight than short and wide trees (see Figure 3). So in this phase, the goal of the adversary is to mine, in secret, a subtree of height two that is heavier than the naturally longer subtree that the honest players are mining by themselves. If the adversary's subtree gets heavier, he can publish it and following the `GHOST` rule force the honest players to switch to a shorter chain.

By doing this repeatedly, every time starting from a recently mined block, and by restarting if honest miners get too far ahead, a concrete reduction of the chain growth speed is achieved. This will be exploited in the proposition below. It is also presented experimentally in Figure 5. A more detailed description of the blockchain retarder phase of our attack is given in Appendix and Figure 4.

Let $A_{\texttt{GHOST}}$ be the attacker we described above for `GHOST`.

Proposition 21. Let $f = 0.3$, $\alpha = 0.17$ and $p < 10^{-4}$. Then, for any attacker A' against Bitcoin it holds that

$$\mathbb{E}[u^{\Pi^{\texttt{GHOST}}_{\texttt{PL}}}_{A_{\texttt{GHOST}}}] - \mathbb{E}[u^{\Pi_{\texttt{PL}}}_{A'}] = \Omega(k)$$

Proof. From Theorem 20 it is implied that

$$\mathbb{E}[u^{\Pi^{\texttt{GHOST}}_{\texttt{PL}}}_{A_{\texttt{GHOST}}}] - \mathbb{E}[u^{\Pi_{\texttt{PL}}}_{A'}] \geq \mathbb{E}[u^{\Pi^{\texttt{GHOST}}_{\texttt{PL}}}_{A_{\texttt{GHOST}}}] - \mathbb{E}[u^{\Pi_{\texttt{PL}}}_{A_{\texttt{BTC}}}]$$

We first focus on $\mathbb{E}[u^{\Pi^{\texttt{GHOST}}_{\texttt{PL}}}_{A_{\texttt{GHOST}}}]$. Let the random variable $\text{NB}(\tau, 1-p)$ denote the number of i.i.d. Bernoulli trials, with probability of success $1-p$, until τ failures occur. The random variable will follow the well known negative binomial distribution. It holds that:

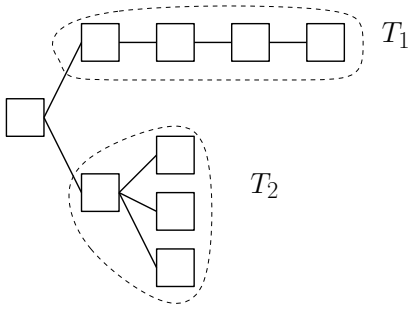

Figure 3: The fundamental idea of our attack. The adversary tries to mine in secret subtree T_2, before the honest players mine subtree T_1. Observe that T_2 has the same weight as T_1, despite being shorter.

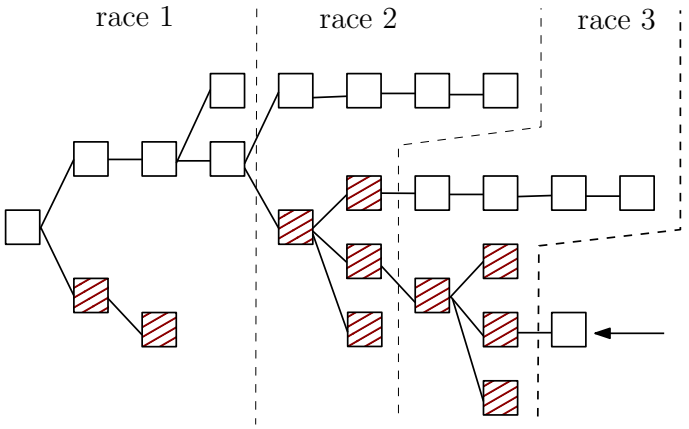

Figure 4: The blockchain retarder phase of the liveness attack on `GHOST`. The adversary has managed to mine first 4 blocks (filled blocks) only on the last two races. Instead of having a chain of length 13 (since 12 successful rounds have occurred), honest parties have chains of length 9.

$$Pr[\mathrm{NB}(\tau, 1-p) \leq k] = 1 - I_{1-p}(k+1, \tau)$$

where $I_{1-p}(k+1, \tau)$ is the regularized incomplete beta function.

Suppose that we launch the blockchain retarder the attack described previously with parameter τ (see Algorithm 3); the adversary tries to mine a short and wide tree with τ nodes before the honest players manage to mine a tree with the same number of nodes. Let E_1 be the event where the number of rounds that the adversary needs in order to mine τ blocks is less than s, and E_2 be the event where the number of rounds that the honest parties needs to have τ successes is more than s. Then, if E_3 is the event where the adversary wins the race and forces honest players to a shorter chain that has grown only by two blocks after s rounds (see Figure 3), it holds that:

$$\begin{aligned} &Pr[E_3] \geq Pr[E_1 \wedge E_2] \geq \\ &Pr[\mathrm{NB}(\tau, 1-p) \leq \frac{\beta s}{p} \wedge \mathrm{NB}(\tau, 1-p) > \frac{\alpha s}{p}] = \\ &(1 - I_{1-p}(\frac{\beta s}{p} + 1, \tau)) I_{1-p}(\frac{\alpha s}{p} + 1, \tau) \end{aligned}$$

The last equality follows from the fact that the two events are independent. For $f = 0.3$, $\alpha = 0.17$, $\beta = 0.13$, $p < 10^{-4}$, $s = 37$ and $\tau = 6$ we get that $Pr[E_3] \geq 0.14$.

Let random variable R_i be equal to 1 if A wins race i and 0 otherwise; we start counting i from the first race after tx has been broadcast. From our previous argument it holds that $Pr[R_i = 1] \geq 0.14$.

We are going to calculate a lower bound on $\mathbb{E}[u_{A_{\texttt{GHOST}}}^{\Pi_{\texttt{PL}}^{\texttt{GHOST}}}]$. Let random variable N be the number of races that have been completed until some honest players confirms tx. On average the adversary is going to win $\mathbb{E}[\sum_{i=1}^{N} R_i] \geq 0.14 \cdot \mathbb{E}[N]$ of them. On the rest of the races, it should take on average $6/\alpha$ rounds for honest players to mine 6 blocks. Thus, by the linearity of expectation

$$\mathbb{E}[u_{A_{\texttt{GHOST}}}^{\Pi_{\texttt{PL}}^{\texttt{GHOST}}}] \geq (0.14 \cdot 37 + (1 - 0.14)6/\alpha)\mathbb{E}[N]$$

In order for some honest party to confirm tx, the chain must have grown by at least k blocks. Hence

$$\mathbb{E}[N](0.14 \cdot 2 + (1 - 0.14)6) \geq k$$

Thus, by combining the inequalities we get that

$$\mathbb{E}[u_{A_{\texttt{GHOST}}}^{\Pi_{\texttt{PL}}^{\texttt{GHOST}}}] \geq \frac{k(0.14 \cdot 37 + (1 - 0.14)6/\alpha)}{0.14 \cdot 2 + (1 - 0.14)6} \geq 6.53k$$

We now turn our attention to the Bitcoin attack. Fix a round t_0 during which a transaction tx is released. Let the random variable F be equal to the minimum round $t_1 \geq t_0$ that has the property $X_{t',t} > Z_{t',t}$ for any t, t' such that $t' < t_0 \leq t_1 < t$, where $X_{t',t}$ is the number of uniquely successful rounds between t', t and similarly $Z_{t',t}$ is the number of blocks mined by the adversary in the same sequence of rounds. Notice, that as we argued in Theorem 20, an honest block containing tx will enter the chains of all honest players permanently at round F. Essentially, the maximum advantage that the adversary will have at round t_0 due to selfish mining, will not be enough so that his secret chain is maintained at the same length as any honest player's chain, and thus a fresh honest block will enter the honest players' chains. Clearly this random variable is independent of k and should have constant in k expected value.

In addition, honest parties need at most $k/\gamma' \leq 6.42k$ rounds on expectation in order to mine k blocks, where γ' is the probability of a successful round. It holds that $\gamma' = 1 - (1 - p)^{q(n-t)} \geq 1 - e^{-\alpha} = 0.156$. Thus, it follows that

$$\mathbb{E}[u_{A_{\texttt{BTC}}}^{\Pi_{\texttt{PL}}}] \leq \mathbb{E}[F] + k/\gamma'$$

But then

$$\mathbb{E}[u_{A'}^{\texttt{GHOST}}] - \mathbb{E}[u_{A_{\texttt{BTC}}}^{\Pi_{\texttt{PL}}}] \geq 6.53k - \mathbb{E}[F] - 6.42k = \Omega(k)$$

where the last inequality follows from the fact that $E[F]$ is constant.

□

The above result, establishes that GHOST can be delayed substantially more than Bitcoin in the presence of an attacker commanding about 43% of the hashing power as long as the target is not too "easy." Note that the expectation becomes worse as k grows, i.e., the stronger the level assurance that is sought by the parties. While this already establishes the lack of efficiency of GHOST we explore further via experiments a wider set of parameters.

Experimental evaluation. The above analysis shows GHOST's Liveness is worse than that of Bitcoin in an asymptotic way. Given the importance of these results also from a practical point of view we present an experimental analysis. Our experiments were obtained via simulating a network of

honest parties running the respective protocol against an adversary that follows the program of liveness attacks. The simulation operates for a certain number of rounds prior to the transaction to be attacked appears, giving the adversary the ability to perform all three phases of our attack. We first examine the rate the honest players' chains grow in the retarder phase of the attacks. The results are shown in Figure 5 where we contrast Bitcoin and `GHOST` miners' chain growth. Then we proceed to test the complete attacks. Given our asymptotic analysis above, as the depth that a transaction needs to find itself in a chain in order to be confirmed, denoted by k, grows bigger, we also expect the confirmation time of Bitcoin to become favorable than that of `GHOST`. This is indeed observed, see Figures 6 and 7. Our simulation also shows (see Figure 10) that the `GHOST` attack performs worse for β approaching γ compared to the Bitcoin optimal attack. So the critical scenario for the `GHOST` attack, is an adversary who even though it does not have enough power to break security, it can still use our attack to slow down confirmation times significantly for the entire network. Since it is not clear whether the `GHOST`-retarder attack is optimal, it remains an open question whether a more efficient attack on confirmation time can be devised when β approaches γ. This question though might be of lesser interest.

(a) (b)

Figure 5: The rate that the chains of honest players grow against our attacks for $f = 1$. Note that as the hashing power of the adversary increases both Bitcoin and `GHOST` speed decrease. However, Bitcoin is clearly favorable to `GHOST` (a) and in fact the ratio of Bitcoin to `GHOST` chain speed increases (b).

(a) (b)

Figure 6: The expected confirmation time plotted for different values of k and parameters (a)$f = 0.3, \beta/f = 36\%$ and (b)$f = 1, \beta/f = 30\%$. Observe that the delay difference when Bitcoin is not attacked and when Bitcoin is attacked remains the same, while the difference with `GHOST` grows.

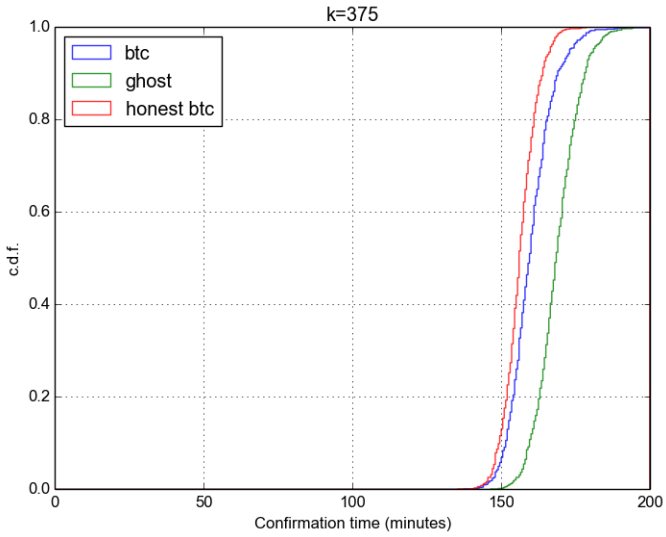

Figure 7: The charts depicts the cumulative distribution function of the confirmation time of the Bitcoin and GHOST based ledgers against the two attacks described in section 5, as well as the expected confirmation of Bitcoin when the attacker stays silent. The parameters used in the experiments are $f = 1$, and adversary hashing power $\beta/f = 30\%$.

Uncle-only GHOST. A prominent GHOST variant is uncle-only GHOST. It was introduced along with Ethereum as a variant between GHOST and Bitcoin. The way uncle-only GHOST works is that each block can refer to a number of uncles (siblings of his ancestor blocks), and for each uncle referred, the chain gains one more unit of weight. Obviously in the same chain, the same uncle can be referred only once. Moreover, in order to reduce the computational overhead of counting uncles deep in the tree, only uncles that are certain levels above (currently suggested 7 in [7]) are counted.

Interestingly, our GHOST-retarder attack still applies to this variant with a small modification. The adversary again tries to mine a short and wide tree. When it decides to release the short tree it has to mine a block under the short tree, in order to capitalize on the blocks mined previously (see Figure 8 for an example).

References

[1] M. Andrychowicz and S. Dziembowski. Pow-based distributed cryptography with no trusted setup. In *Advances in Cryptology–CRYPTO 2015*, pages 379–399. Springer, 2015.

[2] J. Aspnes, C. Jackson, and A. Krishnamurthy. Exposing computationally-challenged byzantine impostors. *Department of Computer Science, Yale University, New Haven, CT, Tech. Rep*, 2005.

[3] L. Bahack. Theoretical bitcoin attacks with less than half of the computational power (draft). Cryptology ePrint Archive, Report 2013/868, 2013. http://eprint.iacr.org/.

[4] J. Bonneau. Ethiks: Using ethereum to audit a coniks key transparency log.

[5] V. Buterin. Toward a 12-second block time, July 2014. https://blog.ethereum.org/2014/07/11/toward-a-12-second-block-time/.

Figure 8: The `GHOST`-retarder attack against (a) the original and (b) uncle-only `GHOST`. Honest (resp. adversarial) blocks are shown with blue (resp. red). Referred uncles are shown with the dotted lines. The score of a chain in uncle-only `GHOST`, is the score of the last block, and the chain ending in the heaviest block is chosen.

[6] S. G. Ethan Heilman. Alison Kendler, Aviv Zohar. Eclipse attacks on bitcoin's peer-to-peer network. Cryptology ePrint Archive, Report 2015/263, 2015. `http://eprint.iacr.org/`.

[7] Ethereum. A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper, 2015.

[8] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse. Bitcoin-ng: A scalable blockchain protocol. *CoRR*, abs/1510.02037, 2015.

[9] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography*, 2014.

[10] J. A. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 281–310, 2015.

[11] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. Cryptology ePrint Archive, Report 2016/555, 2016. `http://eprint.iacr.org/2016/555`.

[12] A. Juels, A. Kosba, and E. Shi. The ring of gyges: Using smart contracts for crime. *aries*, 40:54, 2015.

[13] A. Kiayias and G. Panagiotakos. Speed-security tradeoffs in blockchain protocols. Technical report, IACR: Cryptology ePrint Archive, 2015.

[14] A. Kiayias, H.-S. Zhou, and V. Zikas. Fair and robust multi-party computation using a global transaction ledger, 2015.

[15] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. Technical report, Cryptology ePrint Archive, Report 2015/675, 2015. http://eprint. iacr. org, 2015.

[16] L. Lamport, R. E. Shostak, and M. C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[17] K. Lear. My proposal for ghost protocol, January 2014. `https://bitcointalk.org/index.php?topic=396350.0`.

[18] S. D. Lerner. Even faster block-chains with the decor protocol. Cryptology ePrint Archive, Report 2013/881, May 2014. `https://bitslog.wordpress.com/2014/05/02/decor/`.

[19] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf, 2008.

[20] M. Okun. Agreement among unacquainted byzantine generals. In P. Fraigniaud, editor, *DISC*, volume 3724 of *Lecture Notes in Computer Science*, pages 499–500. Springer, 2005.

[21] S. Omohundro. Cryptocurrencies, smart contracts, and artificial intelligence. *AI matters*, 1(2):19–21, 2014.

[22] R. Pass, L. Seeman, and abhi shelat. Analysis of the blockchain protocol in asynchronous networks. Cryptology ePrint Archive, Report 2016/454, 2016. `http://eprint.iacr.org/`.

[23] M. C. Pease, R. E. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.

[24] C. Percival. Stronger key derivation via sequential memory-hard functions. *Self-published*, pages 1–16, 2009.

[25] J. Peterson and J. Krug. Augur: a decentralized, open-source platform for prediction markets. *arXiv preprint arXiv:1501.01042*, 2015.

[26] M. A. Simplicio Jr, L. C. Almeida, E. R. Andrade, P. C. dos Santos, and P. S. Barreto. The lyra2 reference guide. Technical report, version 2.3. 2. Technical report, 2014.

[27] Y. Sompolinsky and A. Zohar. Secure high-rate transaction processing in bitcoin. *Financial Cryptography and Data Security*, 2015.

A Probability of uniquely successful rounds

In this section we demonstrate a lower bound on the probability of uniquely successful rounds. This bound allows us to argue about the security of `GHOST` even when f is larger than 1.

Lemma 22. *For $p < 0.1$ and $a \in (p, 2k)$: $e^{-a-kp} \leq (1-p)^{\frac{a}{p}-k} \leq e^{-a+kp}$*

Proof. The second inequality is well studied and holds for $p > 0$. For the first inequality by solving for a we get $a \leq k\frac{ln(1-p)}{1+\frac{ln(1-p)}{p}}$ which holds for $p < 0.1$ and $a \in (p, 2k)$. □

Let γ be a lower bound on the probability of a uniquely successful round (a round where only one block is found). From the event where $(n-t)$ players throw q coins each and exactly one coin toss comes head, the probability of a uniquely successful rounds is at least:

$$(n-t)qp(1-p)^{q(n-t)-1} \geq \alpha e^{-\alpha-kp}$$

We set $\gamma = ae^{-a-kp}$, for the minimum k that satisfies the relation $\alpha \in (p, 2k)$. This is a substantially better bound that γ_u and is also a lower bound for the event that at a round is successful. The relation of the two bounds is depicted in Figure 9.

B GHOST Backbone protocol

In this section we present for completeness the remaining procedures of the GHOST backbone protocol. The function pow is the same as the one defined in [10]. The function update gets a block tree and a set of blocks and returns the updated tree containing all new blocks.

Algorithm 4 The *proof of work* function, parameterized by q, D and hash functions $H(\cdot), G(\cdot)$. The input is $(x, \mathcal{C})$.

function pow$(x, \mathcal{C})$
  if $\mathcal{C} = \varepsilon$ **then** ▷ Determine proof of work instance
    $s \leftarrow 0$
  else
    $\langle s', x', ctr' \rangle \leftarrow \text{head}(\mathcal{C})$
    $s \leftarrow H(ctr', G(s', x'))$
  end if
  $ctr \leftarrow 1$
  $B \leftarrow \varepsilon$
  $h \leftarrow G(s, x)$
  while $(ctr \leq q)$ **do**
    if $(H(ctr, h) < D)$ **then**
      $B \leftarrow \langle s, x, ctr \rangle$
      break
    end if
    $ctr \leftarrow ctr + 1$
  end while
  $\mathcal{C} \leftarrow \mathcal{C}B$ ▷ Extend chain
  return $\mathcal{C}$
end function

Algorithm 5 The tree update function, parameterized by q, D and hash functions $H(\cdot), G(\cdot)$. The inputs are a block tree T and an array of blocks.

function update(T,B)
  foreach $\langle s, x, ctr \rangle$ **in** T
  foreach $\langle s', x', ctr' \rangle$ **in** B
  if $((s' = H(ctr, G(s, x))) \wedge (H(ctr', G(x', ctr')) < D))$ **then**
    $children_T(\langle s, x, ctr \rangle) = children_T(\langle s, x, ctr \rangle) \cup \langle s', x', ctr' \rangle$ ▷ Add to the tree.
  end if
  return T
end function

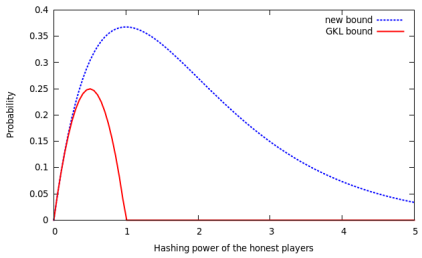

Figure 9: Comparison of the lower bounds on the probability of a uniquely successful round, γ and γ_u, used respectively in this work and [10]. Notice that γ allows as to argue about security when f is greater than 1.

(a) (b)

Figure 10: The expected confirmation time of `GHOST` and Bitcoin for (a) $f = 0.3$, (b) $f = 1$ against the two attacks described in section 5, as well as the expected confirmation of Bitcoin when the attacker stays silent. Notice that when the hashing power of the adversary approaches γ, Bitcoin's confirmation becomes worse than that of `GHOST`.

C Proofs

C.1 Proof of Lemma 16

Proof. Let random variable Z_{s_1,s_2} (resp. $Z^{pub}_{s_1,s_2}$) denote the number of blocks the adversary computes (resp. broadcasts) from round s_1 until round s_2, and random variable X_{s_1,s_2} denote the number of rounds that are uniquely successful in the same interval.

We are first going to prove two preliminary claims. We show that as long as from some round r and afterwards the adversary broadcasts less blocks than the total number of uniquely successful rounds, the chain that any honest player adopts after round r extends $\mathrm{p}_{\mathrm{dom}}(r, X_{1,r} - Z_{1,r})$. More generally we can prove the following claim.

Claim 3. Consider any execution such that for all $s_2 \geq s_1$ it holds that $Z_{1,s_2} < X_{1,s_2}$. Then, the chain that any honest player adopts after round s_1 extends $\mathrm{p}_{\mathrm{dom}}(s_1, X_{1,s_1} - Z_{1,s_1})$.

Proof of Claim. Since $X_{1,s_1} > Z_{1,s_1}$ from Lemma 15 if follows that $p = \mathrm{p}_{\mathrm{dom}}(s_1, X_{1,s_1} - Z_{1,s_1-1}) \neq \perp$. As long as the number of blocks that the adversary broadcasts at round s_2 are less than the dominance of the nodes in p in $T^+_{s_2-1}$, all honest players at round s_2 will adopt chains containing p. Thus uniquely successful rounds will increase the dominance of these nodes. But since from the assumptions made, $Z_{1,s_2} < X_{1,s_2}$, in all rounds after round s_1, the nodes in p are at least 1-dominant in every $T^P_{s_2}$ where P is an honest player; the claim follows. ⊣

Next we will show that if successive u.s. rounds occur such that the blocks mined are on different branches, then the adversary must broadcast an adequate number of blocks, as specified below.

Claim 4. Consider any execution where $s_1 < s_2 < ... < s_m$ are u.s. rounds and s_k is the first u.s. round such that the honest block mined in this round is not a descendant of the honest block mined in round s_{k-1}, for $k \in \{2, .., m\}$. Then either $Z^{pub}_{s_1-1,s_m-1} > X_{s_1,s_m-1}$ or $Z^{pub}_{s_1-1,s_m-1} = X_{s_1,s_m-1}$ and the honest block mined at round s_m will be in $\mathrm{p}_{\mathrm{dom}}(s_m, 1)$.

Proof of Claim. Let $b_1, .., b_m$ denote the honest blocks mined at rounds $s_1, .., s_m$ respectively. We are going to prove the claim for $m = 2$. Suppose, for the sake of contradiction, that $Z^{pub}_{s_1-1,s_2-1} < X_{s_1,s_2-1}$. By the definition of s_2, the honest blocks mined on all u.s. rounds until round $s_2 - 1$ are descendants of b_1. From Lemma 15 at least one honest block b computed in one of the u.s. rounds in $[s_1, s_2 - 1]$ will be in $\mathrm{p}_{\mathrm{dom}}(s_2 - 1, X_{s_1,s_2-1} - Z^{pub}_{s_1-1,s_2-2})$. Since from our hypothesis the adversary will broadcast less than $Z^{pub}_{s_2-1,s_2-1} < X_{s_1,s_2-1} - Z^{pub}_{s_1-1,s_2-2}$ blocks at round $s_2 - 1$, it is impossible for b_2 not to be a descendant of b and thus of b_1 which is a contradiction. Hence, $Z^{pub}_{s_1-1,s_2-1} \geq X_{s_1,s_2-1}$. If $Z^{pub}_{s_1-1,s_2-1} > X_{s_1,s_2-1}$ the base case follows. Otherwise, $Z^{pub}_{s_1-1,s_2-1} = X_{s_1,s_2-1}$ and we have two cases. In the first case, $X_{s_1,s_2-1} = Z^{pub}_{s_1-1,s_2-2}$ and at round round $s_2 - 1$ the adversary does not broadcast any block. From Claim 1 of Lemma 15, b_2 will be in $\mathrm{p}_{\mathrm{dom}}(s_2, 1)$. In the second case, it holds that the adversary broadcasts exactly $X_{s_1,s_2-1} - Z^{pub}_{s_1-1,s_2-2}$ blocks at round $s_2 - 1$. From Claim 2 of Lemma 15, since b_2 cannot be a descendant of the last node of $\mathrm{p}_{\mathrm{dom}}(s_2 - 1, 1)$, b_2 will be in $\mathrm{p}_{\mathrm{dom}}(s_2, 1)$. Hence, the base case follows.

Suppose the lemma holds until round s_m. By the inductive hypothesis we have two cases. In the first case $Z^{pub}_{s_1-1,s_m-1} > X_{s_1,s_m-1}$ which implies $Z^{pub}_{s_1-1,s_m-1} \geq X_{s_1,s_m}$. If no u.s. round happens during rounds $s_m+1, \ldots, s_{m+1}-1$ then from Claim 1 in the proof of Lemma 15 the claim follows. Otherwise, a u.s. round s' happens during these rounds, where the honest block mined is a descendant of b_m. Then we can make the same argument as for the base case starting from round s' and get that either $Z^{pub}_{s'-1,s_{m+1}-1} > X_{s',s_{m+1}-1}$ or $Z^{pub}_{s'-1,s_{m+1}-1} = X_{s',s_{m+1}-1}$ and the honest block mined at round

s_{m+1} will be in $\mathrm{p}_{\mathrm{dom}}(s_{m+1}, 1)$. Since $Z^{pub}_{s'-1,s_{m+1}-1} < Z^{pub}_{s_m-1,s_{m+1}-1}$ and $X_{s',s_{m+1}-1} = X_{s_{m+1},s_{m+1}-1}$, by the inequality of the inductive hypothesis the claim follows.

In the second case $Z^{pub}_{s_1-1,s_m-1} = X_{s_1,s_m-1}$ and the honest block b_m mined at round s_m will be in $\mathrm{p}_{\mathrm{dom}}(s_m, 1)$. From Remark 1 of the proof of claim Lemma 15, for an application of this Lemma from rounds s_m until $s_{m+1} - 1$ we can count the adversarial blocks starting from round s_m. Thus from the same argument as for the base case starting from round s_m we get that either $Z^{pub}_{s_m,s_{m+1}-1} > X_{s_m,s_{m+1}-1}$ or $Z^{pub}_{s_m,s_{m+1}-1} = X_{s_m,s_{m+1}-1}$ and the honest block mined at round s_m will be in $\mathrm{p}_{\mathrm{dom}}(s_m, 1)$. By the equality of the inductive hypothesis the claim follows. $\dashv$

Next, we observe that Lemma 15 as well as Claim 3 and 4 can be applied on a subtree of the block tree, if all honest blocks mined after the round the root of the subtree was mined are on this subtree.

Observation 1. Let b be an honest block computed at round s_1 that is in the chains adopted by all honest players after round s_2. Also, suppose that all blocks mined at u.s. rounds after round s_1 are descendants of b. Then the following hold:

1. Regarding applications of Lemma 15 and Claim 4 on the subtree of the block tree rooted on b after round s_1, we can ignore all blocks that the adversary has mined up to round s_1.
2. Regarding applications of Claim 3 after round s_2, we can ignore all blocks that the adversary has mined up to round s_1.

To see why the observation holds consider the following. Since the adversary receives block b for the first time at round $s_1 + 1$, all blocks that the adversary mines before round $s_1 + 1$ cannot be descendants of b. Regarding the first point, blocks that are not descendants of b do not affect the validity of Lemma 15 and Claim 4 on the subtree of the block tree rooted on b; this is because blocks that are not descendants of b, do not affect the dominance of the nodes of the subtree rooted at b. Regarding the second point, consider the dominant path at round $s_3 > s_2$ in the subtree that is rooted on b. Then, this path can be extended up to the root node, since, by our assumption, b is in the chains adopted by all honest players after round s_2.

We are now ready prove the lemma. First, we are going to define a set of bad events which we will show that hold with probability exponentially small in s. Assuming these events don't occur we will then show that our lemma is implied, and thus the lemma will follow with overwhelming probability.

Let $BAD(s_1, s_2)$ be the event that $X_{s_1,s_2} \leq Z_{s_1,s_2}$. In [10, Lemma 5], by an application of the Chernoff bounds it was proved that assuming that $\gamma \geq (1 + \delta)\beta$ for some $\delta \in (0, 1)$, then with probability at least $(1 - e^{-\frac{\beta}{75}\delta^2 s'})(1 - e^{-\frac{\gamma}{32}\delta^2 s'}) \geq 1 - e^{-(\min(\frac{\beta}{75}, \frac{\gamma}{32})\delta^2 s' - \ln(2))}$ for any $r' > 0, s' \geq s$:

$$X_{r',r'+s'-1} > (1 + \frac{\delta}{2})Z_{r',r'+s'-1} \tag{2}$$

Thus, there exists an appropriate constant $\epsilon = \delta^2 \min(\frac{\beta}{75}, \frac{\gamma}{32})$, independent of r, such that it holds that for any $r' > 0, s' \geq s$, $BAD(r', r' + s' - 1)$ occurs with probability at most $e^{-\epsilon\delta^2 s' + \ln 2}$. From an application of the union bound, we get that for the function $g(s) = \epsilon\delta^2 s - \ln 2 + \ln(1 - e^{-\epsilon\delta^2})$, the probability that $\bigvee_{r' \geq s} BAD(s_1 + 1, s_1 + r')$ happens is:

$$\begin{aligned} Pr[\bigvee_{r' \geq s} BAD(s_1 + 1, s_1 + r')] &\leq \sum_{r' \geq s} e^{-\epsilon\delta^2 r' + \ln 2} \\ &\leq e^{\ln 2} \sum_{r' \geq s} e^{-\epsilon\delta^2 r'} \leq e^{\ln 2} \frac{e^{-\epsilon\delta^2 s}}{1 - e^{-\epsilon\delta^2}} \leq e^{-g(s)} \end{aligned}$$

Until now we have assumed that the execution we are studying is collision-free; no two queries in the oracle return the same value for different inputs. Let $COLL$ denote the event where a collision occurs in our execution. The probability of $COLL$ in a polynomial number of rounds, is exponentially small on κ.

$$Pr[COLL] \leq (f\kappa^c)^2/2^{\kappa+1} = e^{-\Omega(\kappa)} \leq e^{-\Omega(s)}$$

Let $BAD(s_1)$ denote the event where $\bigvee_{r' \geq s} BAD(s_1+1, s_1+r')$ or $COLL$ happens. From the union bound the probability that $BAD(s_1)$ happens, for any s_1 is negligible.

$$Pr[BAD(s_1)] \leq e^{-g(s)} + e^{-\Omega(s)} \leq e^{-\Omega(s)}$$

We are going to show next that, conditioning on the negation of this event the statement of the lemma follows.

We will use the convention that block b_i is mined at round r_i. Let b_1 be the most recent honest block that is in the chains that all honest players have adopted on and after round r, such that the blocks mined at all u.s. rounds after round r_1 are descendants of b_1. This block is well defined, since in the worst case it is the genesis block. If r_1 is greater or equal to $r-s$, then the lemma follows for block b_1 with probability 1.

Suppose round r_1 is before round $r-s$ and that $BAD(r_1)$ does not happen. The negation of $BAD(r_1)$ implies that $X_{r_1+1,r-1+c} > Z_{r_1+1,r-1+c}$, for $c \geq 0$. By Lemma 15 and Claim 3 there exists at least one honest block b_2, mined in a u.s. round and contained in the chains of all honest players on and after round r. W.l.o.g. let b_2 be the most recently mined such block. By the definition of b_1, b_2 is a descendant of b_1. If r_2 is greater or equal to $r-s$ then the lemma follows, since b_2 is an honest block mined on and after round $r-s$ that satisfies the conditions of the lemma.

Suppose round r_2 is before round $r-s$. Let r_3 be the earliest u.s. round, such that b_3 and the blocks mined at all u.s. rounds afterwards are descendants of b_2. Since b_2 will be in the chains of all honest players after round r, round r_3 is well defined. Also let $s_1 < \ldots < s_m < \ldots$ be the sequence of u.s. rounds after round r_1 that satisfy the conditions of Claim 4. That is, s_k is the first u.s. round such that the honest block mined in this round is not a descendant of the honest block mined in round s_{k-1}, for $k \in \{2, .., m\}$. The first u.s. round after round r_1 corresponds to s_1.

We will argue that r_3 is equal to some $s_i > s_1$ in the aforementioned sequence. Suppose, for the sake of contradiction that it does not. This implies that the honest block mined at round r_3 (denoted by b_3) is a descendant of the honest block mined at some round s_i of the sequence. W.l.o.g. suppose that s_i is the largest such round that is before round r_3. There are three cases. In the first case, $r_2 < s_i < r_3$. By the definition of s_i and r_3, the block mined at round s_i is an ancestor of b_3 and also a descendant of b_2. Hence, s_i satisfies the definition of r_3 which is a contradiction (there is an earlier round than r_3 with the same property). In the second case, $s_i = r_4$, where b_4 is a descendant of b_1 and either $b_2 = b_4$ or b_4 is an ancestor of b_2. Then b_4 is a block that satisfies the definition of b_1, and is more recent, which is a contradiction. In the third case, $r_1 < s_i < r_2$ and the block mined at round s_i is not an ancestor of b_2. By the definition of s_i, the honest block mined at round s_i is an ancestor of b_3, that has been mined before round r_2. But this is contradictory, since no honest block can be an ancestor of b_3, mined before round r_2, but not be an ancestor of b_2.

Since we proved that r_3 is equal to some s_i we can apply Claim 4 from round r_1+1 until round r_3. Again, from Observation 1, regarding applications of Claim 4 after round r_1 we can ignore blocks that were mined before round r_1+1. Then either $Z_{r_1+1,r_3-1} \geq Z^{pub}_{r_1+1,r_3-1} > X_{r_1+1,r_3-1}$ or $Z_{r_1+1,r_3-1} \geq Z^{pub}_{r_1+1,r_3-1} = X_{r_1+1,r_3-1}$ and the honest block mined at round r_3 will be in $\mathrm{p_{dom}}(r_3, 1)$.

Suppose, for the sake of contradiction, that round r_3 is after round $r_2 + s$. Then $(r_3 - 1) - (r_1 + 1) \geq s$ and $Z_{r_1+1,r_3-1} \geq X_{r_1+1,r_3-1}$. This is a contradiction, since in this case $\neg BAD(r_1)$ implies $Z_{r_1+1,r_3-1} < X_{r_1+1,r_3-1}$. Therefore, $r_3 \leq r_2 + s < r$. In addition, notice that $\neg BAD(r_1)$ also implies

$$X_{r_1+1,r_2+s} > Z_{r_1+1,r_2+s} \tag{3}$$

We are going to apply Lemma 15 and Observation 1 from round r_3 until round $r_2 + s$ in the subtree rooted at b_2. According to the analysis we made previously there are two cases. In the first case, $Z^{pub}_{r_1+1,r_3-1} > X_{r_1+1,r_3-1}$ or equivalently $Z^{pub}_{r_1+1,r_3-1} \geq X_{r_1+1,r_3}$. Suppose, for the sake of contradiction, that $r_3 = r_2 + s$. Then $Z_{r_1+1,r_2+s-1} \geq X_{r_1+1,r_2+s}$. But this is a contradiction, since $\neg BAD(r_1)$ implies Inequality 3. Therefore, $r_3 < r_2 + s$. From Inequality 3:

$$\begin{aligned} X_{r_3+1,r_2+s} &\geq X_{r_1+1,r_2+s} - X_{r_1+1,r_3} \\ &> Z_{r_1+1,r_k+s} - Z^{pub}_{r_1+1,r_3-1} \geq Z^{pub}_{r_3,r_2+s} \end{aligned}$$

The last inequality, stems from two facts: that we can ignore blocks that were mined before round $r_1 + 1$ regarding applications of Lemma 15 and also that the blocks that the adversary broadcasts at distinct rounds are different (adversaries that broadcast the same block multiple times can be ignored without loss of generality).

In the second case, $Z^{pub}_{r_1+1,r_3-1} = X_{r_1+1,r_3-1}$ and the honest block mined at round r_3 will be in $\mathrm{p}_{\mathrm{dom}}(r_3, 1)$. Again from Inequality 3:

$$\begin{aligned} X_{r_3,r_2+s} &= X_{r_1+1,r_2+s} - X_{r_1+1,r_3-1} \\ &> Z_{r_1+1,r_k+s} - Z^{pub}_{r_1+1,r_3-1} \geq Z^{pub}_{r_3,r_2+s} \end{aligned}$$

The same analysis holds for all rounds after $r_2 + s$. By an application of Claim 3, an honest block b, computed in one of the u.s. rounds after round r_2 and before round r, will be in the chains that honest players adopt on and after round r. Since b_2 is the most recently mined block, before round $r - s$, included in the chain of all honest players, b must have been mined on and after round $r - s$ (since $r_3 > r_2$). Let A be the event that there exists a block mined by an honest player on and after round $r - s$, that is contained in the chain which any honest player adopts after round r. We have proved that $(\neg BAD(r_1))$ implies A. Then:

$$\begin{aligned} Pr[A] =& Pr[A \wedge BAD(r_1)] + Pr[A \wedge \neg BAD(r_1)] \\ \geq& Pr[A \wedge \neg BAD(r_1)] \\ =& Pr[A|\neg BAD(r_1)]Pr[\neg BAD(r_1)] \\ =& Pr[\neg BAD(r_1)] \\ \geq& 1 - e^{-g(s)} \end{aligned}$$

Hence, the lemma holds with probability at least $1 - e^{-g(s)}$. □