Speed-Security Tradeoffs in Blockchain Protocols

Aggelos Kiayias* Giorgos Panagiotakos†

December 6, 2015

Abstract

Transaction processing speed is one of the major considerations in cryptocurrencies that are based on proof of work (POW) such as Bitcoin. At an intuitive level it is widely understood that processing speed is at odds with the security aspects of the underlying POW based consensus mechanism of such protocols, nevertheless the tradeoff between the two properties is still not well understood.

In this work, motivated by recent work [9] in the formal analysis of the Bitcoin backbone protocol, we investigate the tradeoff between provable security and transaction processing speed viewing the latter as a function of the block generation rate. We introduce a new formal property of blockchain protocols, called *chain growth*, and we show it is fundamental for arguing the security of a robust transaction ledger. We strengthen the results of [9] showing for the first time that reasonable security bounds hold even for the faster (than Bitcoin's) block generation rates that have been adopted by several major "alt-coins" (including Litecoin, Dogecoin etc.). We then provide a first formal security proof of the GHOST rule for blockchain protocols. The GHOST rule was put forth in [14] as a mechanism to improve transaction processing speed and a variant of the rule is adopted by Ethereum. Our security analysis of the "GHOST backbone" matches our new analysis for Bitcoin in terms of the common prefix property but falls short in terms of chain growth where we provide an attack that substantially reduces the chain speed compared to Bitcoin. While our results establish the GHOST variant as a provably secure alternative to standard Bitcoin-like transaction ledgers they also highlight potential shortcomings in terms of processing speed compared to Bitcoin. We finally present attacks and simulation results against blockchain protocols (both for Bitcoin and GHOST) that present natural upper barriers for the speed-security tradeoff. By combining our positive and negative results we map the speed/security domain for blockchain protocols and list open problems for future work.

1 Introduction

The capability for fast transaction processing is a major consideration in any payment system and a litmus test for its potential to scale at a global level. For "blockchain" based protocols such as bitcoin [12] the current picture is rather grim: some reported[1] current rates for Bitcoin processing speed is 7 transactions per second (tps) while Paypal handles an average of 115 tps and the VISA network has a peak capacity of 47,000 tps (though it currently needs 2000-4000 tps). It goes without saying that improving transaction processing of cryptocurrencies is one of the major considerations in the research of payment systems like Bitcoin, cf. [3].

Bitcoin relies on the distributed maintenance of a data structure called the blockchain by a set of entities called miners that are anonymous and potentially dynamically changing. The protocol

*Research supported by ERC project CODAMODA.
†Research supported by ERC project CODAMODA.
[1]See https://en.bitcoin.it/wiki/Scalability

that maintains the blockchain relies on proofs of work (POW) for ensuring that miners converge to a unique view of this data structure. The blockchain can be parsed as a ledger of transactions and assuming that the adversarial parties collectively constitute less than half of the network's computational power (also referred to as hashing power since the main computational operation is hashing) it is ensured that all parties have the same view of the ledger. The transactions in the blockchain are organized in blocks and each block is associated with a POW. The number of transactions that fit inside each block is bounded (and is currently restricted by a 1MB cap).

Beyond the obvious engineering factors that affect transaction processing speed of blockchain protocols (such as network speed and computational power needed to verify transactions) the two main factors are the size of blocks and the rate that blocks are generated. The current 1MB cap on transactions is heavily debated and proposals for a 20-fold increase have been recently made[2]. Regarding the block generation rate recall that the original parameter setting for Bitcoin stabilizes it at 1 block per 10 minutes. This is achieved by suitably calibrating the hardness of the POW instances that are solved by the miners. At an intuitive level, the POW difficulty is an intrinsic feature for security as it prohibits the adversary from flooding the network with messages and gives the opportunity to the honest parties to converge to a unified view.

A useful unit of time to measure the block generation rate is a round of full information propagation. Indeed, the effect that the speed of information propagation may have on security is widely understood at least informally and the effect of the former on the latter was predicted by [7]. In [9] a formal relation between the two was proven: it was observed that security can be formally shown if the parameter f, expressing the expected number of POW solutions per complete round of information propagation, is close to 0. In that work it was shown that as f gets closer to 0 the maximum adversarial hashing power that the protocol can withstand approximates 50%, Bitcoin's claimed theoretical limit; on the other hand, as f gets larger the security bound gets worse and it completely vanishes when $f = 1$, i.e., the rate of expected 1 block per round.

In [7] it is argued that for blocks of reasonable sizes (including those currently used), the block size is linearly dependent in the time it takes for a full communication round to be completed. From this one can argue that round duration is linearly related to block size. Furthermore, transaction processing speed is proportional to block size and also proportional to block generation rate per unit of time (say seconds). Given that we measure time in rounds of full communication we can express the following intuitive relation for transaction processing speed (measured in Kb/sec):

$$\text{transaction processing speed} \propto \frac{\text{block size} \times f}{\text{round duration}}$$

As a result, since doubling the block size also doubles the round duration, if we keep the same value of f, the transaction speed remains constant. Hence, the dominant factor for improving transaction processing speed, is not block-size, but rather the block generation rate (per round) represented by f. Given the security critical nature of this parameter it is important to understand how large it can be selected while maintaining the security of the system.

Interestingly, a number of alternative cryptocurrencies (alt-coins) that are based on Bitcoin have tinkered with the block generation rate of Bitcoin (see Figure 1) to achieve faster processing without however providing any formal arguments about the security implications of such choices.

Given the above motivation the fundamental question we seek to answer is the following:

For a given block generation rate expressed as the expected number of blocks per round

[2]See e.g., [6, 15, 13] and http://gavintech.blogspot.gr/2015/01/twenty-megabytes-testing-results.html

[3]Currently the Ethereum Frontier reports an average of about 17 seconds, cf. `https://etherchain.org`; the 12 seconds rate was discussed by Buterin in [5].

Cryptocurrency	block gen. rate (sec)	f (blocks/round)	$1/f$
Bitcoin	600	0.021	47.6
Litecoin	150	0.084	11.9
Dogecoin	60	0.21	4.76
Flashcoin	$6-60$	0.21-2.1	0.476-4.76
Fastcoin	12	1.05	0.95
Ethereum[3]	12	1.05	0.95

Figure 1: A list of the different block generation rates various altcoins have chosen and the corresponding $f, 1/f$ values assuming one full communication round takes 12.6 seconds (this is the average block propagation time as measured in [7]). Notice Bitcoin's conservative choice. The value f is the expected number of POW's per communication round. The value $1/f$ is also given which is roughly the expectation of rounds required to obtain a POW.

(parameter f), what is the maximum adversarial hashing power that can be *provably* tolerated by a population of honest miners?

The above question may be posed for the core of the Bitcoin transaction ledger protocol (the Bitcoin "backbone" protocol as defined in [9]) but also for other similar protocols that attempt to use POW's to maintain a blockchain distributively notably the GHOST rule suggested by Sompolinsky and Zohar [14] and adopted by Etheurem.

Our Results. In this work, we investigate speed-security tradeoffs in blockchain protocols as a relationship between block generation rate f and the bound on the hashing power of the adversary. Specifically, our results are as follows.

- We introduce a new property for blockchain protocols, called *chain growth* that is cast in the model of [9] and complements the two properties suggested there (common prefix and chain quality). We argue that chain growth is a fundamental property of backbone protocols independent of the other two. We illustrate this by showing that a backbone protocol satisfying all three properties implements a "robust transaction ledger" in a black-box fashion (something that we observe to be not true if one relies on just common prefix and chain quality — the two properties by themselves are insufficient to imply a robust transaction ledger[4]). Furthermore, chain growth is a property of interest from an attacker's point of view as it is fundamentally linked to the transaction processing speed and can constitute an adversarial goal in its own right: it captures the class of adversaries that are interested in slowing down processing time.

- We propose a new analysis framework for backbone protocols focusing on trees of blocks as opposed to chains as in [9]. We illustrate the power of our framework by substantially improving the security analysis of the bitcoin backbone protocol and proving for the first time that security can still be attained even at expected rates of f below 1 block per round. At the same time, we substantially improve the level of security for higher rates and in this way we prove security for bounds close to 50% for important (in terms of their market capitalization[5]) alternative cryptocurrencies (including Litecoin and Dogecoin) that have opted for much faster block creation rates compared to Bitcoin. See Figures 3 and 4 for graphs showing our improved security analysis.

[4]This does not suggest an error in [9] but rather points to the fact that the proof given there regarding the implementation of a robust transaction ledger by the bitcoin backbone is not black-box on the two properties of common prefix and chain quality.

[5]See http://coinmarketcap.com/

- Using our framework we also provide a first formal security proof of the GHOST rule for blockchain protocols. The GHOST rule was put forth in [14] as a mechanism to improve transaction processing speed. We formalize the rule as the GHOST backbone protocol and provide a security analysis in our framework that matches our new analysis for the Bitcoin backbone in terms of the common prefix property. Even though we prove chain quality and chain growth as well, contrary to the Bitcoin backbone, we show that the GHOST backbone is susceptible to a chain growth attack. While the analysis presented in [14] suggests that GHOST is as good as Bitcoin in terms of chain growth, our attack, rather surprisingly, shows the contrary and in fact Bitcoin's chain growth is substantially faster than GHOST, cf. Figure 6, when under attack. Our work also highlights the importance of provable security in the exploration of the design space for Bitcoin-like blockchain protocols; for instance, while at first one may see the GHOST-rule as being superior to Bitcoin's "longest-chain wins" simple rule, the enhanced rule opens new opportunities for adversarial manipulation that need to be accounted for in the security proof.
- We finally present simulation results and attacks against blockchain protocols (both for the Bitcoin and GHOST backbone) that present natural upper barriers in the speed-security domain. Interestingly, for common prefix the attacks do not differentiate between GHOST and Bitcoin even for settings of f that correspond to high processing times (the area of the parameter domain where supposedly GHOST was particularly well suited for): both protocols lose security approximately for the same parameter settings, cf. Figure 5. An intuitive explanation for the rather unexpected similarity is the fact that in the GHOST backbone, the chain selection rule permits the use of old blocks, while in Bitcoin the attacker is forced to use recent blocks.

Limitations and directions for future research. Our analysis is in the standard cryptographic model where parties fall into two categories, those that are honest (and follow the protocol) and those that are dishonest that may deviate in an arbitrary (and coordinated) fashion as dictated by the adversary. It is an interesting direction for future work to consider speed-security tradeoffs in the rational setting where all parties wish to optimize a certain utility function. Designing suitable incentive mechanisms is a related important consideration, for instance see [10] for a suggestion related to the GHOST protocol. The analysis we provide for both Bitcoin and GHOST 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, [1], and hence it is an important point for consideration and future work. Our notion of round (borrowed from [9]) assumes complete information propagation between all honest parties; in practice information propagation is a random variable that depends on the peer to peer network topology and some parties learn faster than others the messages communicated. Finally, the positive and negative results we present between speed and security still have a gray area in which it is unknown whether the protocols are secure or there is an attack that breaks security (for instance, while we show the chain growth of the GHOST backbone to be worse than Bitcoin's by providing an upper bound via an attack, the lower bound we prove for chain growth of GHOST is not tight to the attack upper bound and hence the true chain growth speed of GHOST lies somewhere in this interval). While the above four points are limitations (and suggest interesting directions for further research in the area) our model and analysis can be extended to account for such stronger settings and hence our results may serve as the basis for further exploring the tradeoff between transaction processing speed and provable security. Another important aspect is privacy in the transaction ledger (cf. [2, 11]) which our analysis, being at a "lower" level in the blockchain protocol does not interact with directly.

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 the chain growth property as well as our new tree-based framework. In section 4 we present our improved analysis for the Bitcoin backbone protocol. Then, in section 5 we present our security analysis of an abstraction of the GHOST protocol that demonstrates it is a robust transaction ledger in the static setting. Finally, in section 6 we present our attacks against the common prefix and chain growth properties for both GHOST and Bitcoin as well as we graph the speed-security domain in terms of attack and provable security bounds.

2 Preliminaries

2.1 Model

For our model we adopt the abstraction proposed in [9]. Specifically in their setting, called the q-bounded setting, synchronous communication is assumed that allows each party 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. 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. 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 $f = \alpha + \beta$.

In [9] 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 sufficient for the setting of small f in [9], here we will need to use a better lower bound to the probability of these events (see Appendix) and to the probability of a round being *leading branch* (see section 3.2). We will define this bound as $\gamma = \alpha e^{-\alpha}$. Observe that $\gamma > \gamma_{\mathsf{u}}$.

The only difference from the model of [9] is that if an honest player in a given round mines one block, then he continues until all of his queries are spent. So he may find more than one solutions in a round, and thus extend the longest chain by more than one blocks. A number of definitions that will be used extensively are listed below.

Definition 1. [9] (divergence) Two chains diverge at a given round if the last block of their common prefix was computed before that round.

(successful round) A round is called successful if at least one honest player computes a solution in this round.

(uniquely successful round) A round is called uniquely successful if exactly one honest player computes a solution in this round.

Definition 2. (extends) We will say that a chain $\mathcal{C}$ extends another chain $\mathcal{C}'$ if a prefix of $\mathcal{C}'$ is a suffix of $\mathcal{C}$.

(recent) By recent(s) we denote the set of blocks that were computed at round s and afterwards.

2.2 Backbone Protocols

In order to study the properties of the core Bitcoin protocol, the term *Backbone Protocol* was introduced in [9]. On this level of abstraction we are only interested on properties of the blockchain,

independently from the data stored inside the blocks. In the same work the Bitcoin backbone protocol is described in a quite abstract and detailed way. The main idea is that honest players, at every round, receive new chains from the network and pick the longest valid one to mine. Then, if they mine a block, they broadcast their chain at the end of the round. For more details we refer to [9, Subsection 3.1].

The same level of abstraction can also be used to express the GHOST protocol. The GHOST backbone protocol as presented in [14] 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 heard, and then using the greedy heaviest observed subtree (GHOST) rule they peak which chain to mine.

Algorithm 1 The function that finds the "best" chain. The input is a block tree T.

1: **function** GHOST(T)
2: $B \leftarrow GenesisBlock$
3: **if** $children_T(B) = \emptyset$ **then**
4: **return** $\mathcal{C} = (GenesisBlock, ..., B)$
5: **else**
6: $B \leftarrow argmax_{c \in children_T(B)} |subtree_T(c)|$
7: **end if**
8: **go to** 3
9: **end function**

At every round players update their tree by adding valid blocks sent by other players. The same principle as Bitcoin applies, but now 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.

2.3 Security Properties of the Backbone protocols

In [9, 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 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.

Definition 3 (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 $\text{VIEW}^{H(\cdot)}_{\Pi,\mathcal{A},\mathcal{Z}}(\kappa, q, z)$, it holds that

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

Definition 4 (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 $\text{VIEW}^{H(\cdot)}_{\Pi,\mathcal{A},\mathcal{Z}}(\kappa, q, z)$, 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 [9, Theorems 9&10] the following were proved:

Theorem 5. *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 6. *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 $e^{-\Omega(\delta^2\ell)}$.*

2.4 Robust public transaction ledgers

In [9] 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 [9].

Definition 7. 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 build on top of the Bitcoin backbone protocol. Formally, in [9, Lemma 15&16] the following were proved:

Lemma 8 (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 9 (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 Chain Growth and Trees of blocks

In this section we introduce our new security property, called *chain growth*, and a new analysis framework based on tress of blocks.

3.1 Chain Growth

In addition to the two security properties of the Bitcoin backbone protocol mentioned in Section 2.3, and inspired from the comparative analysis of Bitcoin and GHOST, we define a new property called *chain growth*. This property aims at expressing the minimum rate at which the chains of honest players grow. It is motivated by an attacker that has objective to slow down the overall transaction processing time of the blockchain system. The common prefix and chain quality properties do not explicitly address this issue, and this can be seen from the fact that both properties can hold even if honest players' chains do not grow at all.

Definition 10. (Chain Growth Property) The chain growth property Q_{cg} with parameters $\tau \in \mathcal{R}$ (the "chain speed" coefficient) and $s \in \mathbb{N}$ states that for any round $r > s$, where honest party P has chain $\mathcal{C}_1^P$ at round r and chain $\mathcal{C}_2^P$ at round $r-s$ in $\text{VIEW}_{\Pi,\mathcal{A},\mathcal{Z}}^{H(\cdot)}(\kappa, q, z)$, it holds that $\min_P |\mathcal{C}_1^P| - \min_P |\mathcal{C}_2^P| \geq \tau \cdot s$.

Bitcoin. For the Bitcoin backbone protocol this property is satisfied with parameter τ equal to γ and with overwhelming probability in s. Since all honest players choose the longest chain they see, and successful rounds happen with rate γ, their chains will grow at least at this rate. The worst the adversary can do is not participate, so this is a tight bound.

Theorem 11. *The Bitcoin protocol satisfies the chain growth property with speed coefficient* $(1-\delta)\gamma$ *and probability at least* $1 - e^{-\Omega(\delta^2 s)}$*, for* $\delta \in (0, 1)$.

Proof. Let $r, s \in \mathbb{N}$ and $\text{base}(r)$ denote the minimum length chain that an honest player mines at round r. Suppose that at round $r-s$, $\text{base}(r) = l$. We are going to show that at round r, $\text{base}(r)$ is at least $l + (1-\delta)\gamma s$ with probability $1 - e^{-\Omega(\delta^2 s)}$.

It holds that if some round r' is successful: $\text{base}(r'+1) \geq \text{base}(r') + 1$, because the honest player that mined the new solution at round r' was mining a chain of size at least $\text{base}(r')$. Inductively if between rounds r and $r-s$ there are k successful rounds, $\text{base}(r) \geq \text{base}(r-s) + k$.

But notice that γ is a lower bound on successful rounds. From the Chernoff bound at least $(1-\delta)\gamma s$ such rounds will occur between rounds $r-s+1$ and r with probability $1 - e^{-\Omega(\delta^2 s)}$. Thus $\text{base}(r+s) \geq \text{base}(r) + (1-\delta)\gamma s$ with probability $1 - e^{-\Omega(\delta^2 s)}$. □

The importance of chain growth as a fundamental property of the backbone protocol that is of the same caliber as common prefix and chain quality can be seen in the fact that the liveness of the ledger essentially depends on it. We elaborate: in [9, Lemma 16] the liveness property was not proved in a black box manner given the chain quality and common prefix properties. Interestingly, by introducing the chain growth property as a prerequisite together with the other two, a simple black box proof can be derived. As expected, the confirmation time parameter u of the liveness property is tightly connected to the chain speed coefficient τ.

Lemma 12 (Liveness)**.** *Let protocol* Π *satisfy the chain quality and chain growth properties with overwhelming probability on* l, s *and parameters* μ, τ*. Further, assume oracle* Txgen *is unambiguous. Then protocol* Π *satisfies Liveness with wait time* $u = \frac{2}{\tau} \cdot \max(k, \frac{1}{1-\mu})$ *rounds and depth parameter* k *with overwhelming probability in* k.

Proof. We prove that assuming all honest players receive as input the transaction tx for at least u rounds, there exists an honest party at round r with chain $\mathcal{C}$ such that tx is included in $\mathcal{C}^{\lceil k}$. From the chain growth property after u rounds the chain of all honest players has grown by at least $\tau u (\geq 2k)$ blocks with overwhelming probability on k. From the chain quality property there exist

at least $\frac{\tau u}{2}(1-\mu)(\geq 1)$ honest blocks in the length-k suffix of $C^{\lceil k}$ with overwhelming probability on k. Thus tx is included in these blocks and the lemma follows with overwhelming probability on k. □

GHOST. In the GHOST backbone we will prove that the chain growth property is more nuanced and it is in fact possible for the adversary to mount a non-trivial attack against it. We defer the details of the analysis and attack for section 6.2.

3.2 Trees of blocks

We introduce next our new analysis framework for backbone protocols that is focusing on trees of blocks . In this model every player stores all blocks he hears on a tree starting from the *Genesis* (or v_{root}) block. This is the model where GHOST is normally 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 minimal and maximal knowledge of honest players regarding the block tree on every round.

Definition 13. $T_r^{\forall}$ (resp. $T_r^{\exists}$) is the tree formed by blocks s.t. $\forall$(resp. $\exists$)$p \in$ honest players: p has received block b at the beginning of round r. Similarly, T_r^{tot} is the tree that contains $T_r^{\exists}$ and also includes all blocks mined by honest players at round r. Also we denote by T_r^P the tree that is formed from the blocks that player P has received until the beginning of round r and by $T_r^*(b)$ the subtree of T_r^* rooted on b where $* \in \{\forall, \exists, \text{tot}, P\}$.

Blocks in $T_r^{\forall}$ have been received by all players, and at least one honest player has received the blocks in $T_r^{\exists}$. So for every honest player P it holds that:

$$T_r^{\forall} \subseteq T_r^P \subseteq T_r^{\exists} \subseteq T_r^{\text{tot}}$$

Intuitively, heavier trees represent more proof of work. But there are more than one ways to define what is a "heavy" tree. For example, in Bitcoin a heavy tree is a long one. But for GHOST a heavy tree is one with many nodes. To capture this abstraction we condition our definitions on a norm g defined on trees. This norm will be responsible for deciding what is heavy, and thus favored by the chain selection rule. We choose to omit g from the notation since it will always be clear from the context which norm we use.

Definition 14. For each round r of the protocol we define the following three functions on the nodes of T_r^{tot} under a norm g defined on forests (sets of trees).

- $|v|_{\text{old}}^r$: If $v \in T_r^{\exists}$ then $|v|_{\text{old}}^r = g(T_r^{\exists}(v))$, otherwise $|v|_{\text{old}}^r = 0$.
- $|v|_{\text{tot}}^r$: $|v|_{\text{tot}}^r = g(T_r^{\text{tot}}(v))$.
- $|v|_{\text{new}}^r$: Let F be the forest formed by blocks mined by honest players at round r that are descendants of v (possibly including v). Then $|v|_{\text{new}}^r = g(F)$.

Let $siblings(v)$ denote the set of nodes in $T_r^{\exists}$ that share the same parent with v. Then node v is **d-dominant** at round r w.r.t. $\text{f} \in \{\text{old}, \text{new}, \text{tot}\}$ iff

$$\text{dom}_{\text{f}}^r(v, d) \Leftrightarrow |v|_{\text{f}}^r \geq d \wedge \forall v' \in siblings(v) : |v|_{\text{f}}^r \geq |v'|_{\text{f}}^r + d$$

The Bitcoin protocol can be described using the notion of the d-dominant node. Let g be the length of the longest tree in the forest. 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. Interestingly GHOST can also be described this way by setting g to be the number of nodes of the forest. Thus we have a unified way for describing both protocols. Building upon this unified language we can describe the paths that fully informed honest players may choose to mine in a quite robust way, thus showcasing the power of this notation.

Definition 15. (Paths sets)

- Paths(T) is the maximal set of root-leaf paths of tree T
- HonestPaths(r, b) is the maximal subset of $Paths(T_r^{\exists}(b))$ s.t.
  $\forall p = v_0 v_1..v_k \in \text{HonestPaths}(r, b)\ \forall i \in \{1, .., k\}\ \text{dom}^r_{\text{old}}(v_i, 0)$

Having established the necessary nomenclature we introduce a new technical tool, the notion of *leading branch* rounds. Intuitively, a leading branch round can be thought of as a round that gives the opportunity for honest players to consent. The idea is that leading branch rounds will throw off balance the tree, and mining paths on the following round will become concentrated on one branch of the tree. The adversary can try to balance the tree, so that a fork will be created, but he has to pay for it by mining blocks on the weak branch.

A unique dominating path exists every time a leading branch round happens pointing to the subtree where honest players will mine in the next round (unless the adversary interferes). We define both leading branch rounds and the dominating path formally as follows.

Definition 16. ($LB(s, d)$ rounds and $\text{p}_{\text{lb}}(r, s)$) We call round r *Leading Branch with respect to round s and difference $d \geq 1$* if and only if at that round, d is the maximum value s.t. the following set is non-empty:

$$\{p = v_{root} v_1..v_k \,|p \in Paths(T_r^{\text{tot}}) \ \wedge \exists i : v_i \in \text{recent}(s) \wedge \\ \forall i\ (\text{parent}(v_i) \notin \text{recent}(s)) \Rightarrow \text{dom}^r_{\text{new}}(v_i, d)\}$$

The common prefix up to the first node that was computed at round s or afterwards of all paths in this set (for the maximum value of d), if it exists, is denoted by $\text{p}_{\text{lb}}(r, s)$.

Note that leading branch rounds as defined here, constitute a generalization of the uniquely successful rounds of [9]. Uniquely successful rounds are defined independently of the history, but this is not the case for leading branch rounds, as they depend on $T_r^{\exists}$. Observe that every round r that is uniquely successful, is also a $LB(r, 1)$ round both for Bitcoin and GHOST. Additionally leading branch rounds have the extra parameter s, that is related to how deep the imbalance on the tree is, in terms of rounds. For example suppose no fork exists on $T_r^{\exists}$ (i.e., $T_r^{\exists}$ is a chain) and honest miners have mined new blocks. Then this round is leading branch for every s. If a round is leading branch for s_1, then it will also be for all s_2 that are smaller than s_1. However, the reverse does not always hold. Suppose honest miners only mine two new blocks, on top of two sibling nodes. Then for s larger than the round that their common ancestor was mined, this is not a leading branch round, but for smaller s it is.

Notice that uniquely successful rounds happen less and less often as the expected number of rounds per block[6] $1/f$ decreases. By focusing on the difference of the number of new solutions on

[6] Expected number of rounds per block is approximately $\frac{1}{f}$. The random variable described follows a negative binomial distribution with parameter $(1-p)$ and thus the expectation is $\lceil \frac{1}{f} - \frac{p}{f} \rceil$, where p is the probability that a query on the hash oracle will be successful (which is very small compared to 1).

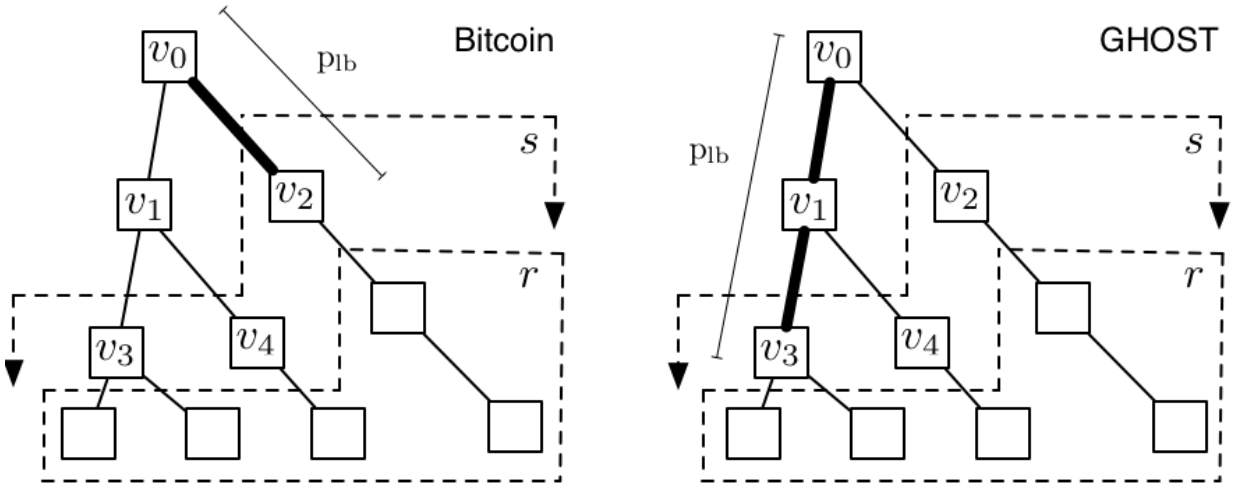

Figure 2: Illustration of the setting when round r is in $LB(s,1)$. In the left side (Bitcoin - the g norm measures maximum height), $|v_2|^r_{\text{new}} = 2$ while $|v_1|^r_{\text{new}} = 1$, hence v_2 is 1-dominant. In the right side (GHOST - the g norm measures number of nodes), $|v_2|^r_{\text{new}} = 2$ while $|v_1|^r_{\text{new}} = 3$, hence v_1 is 1-dominant and $|v_3|^r_{\text{new}} = 2$ while $|v_4|^r_{\text{new}} = 1$, hence v_3 is 1-dominant.

different branches, and not on their absolute number, we manage to describe a class of "good" events that happen with a non-negligible probability even for $1/f < 1$.

Remark 1. For Bitcoin, g is chosen to be the length of the longest tree in the forest and leading branch rounds are represented by $LB^{\max}$. For GHOST, g is chosen to be the total number of nodes on the forest, and leading branch rounds are represented by LB^{sum}.

Remark 2. For the Bitcoin and GHOST backbone protocols it holds that if, at round r, a block b is d_1-dominant w.r.t. to old and d_2-dominant w.r.t. to new and for any player P that mines some of the new blocks $T^P_r(b) = T^{\exists}_r(b)$, then b is $(d_1 + d_2)$-dominant w.r.t. to tot.

4 Bitcoin

4.1 A better bound for the common prefix property

In this section we present a better security bound than the one in [9] regarding the common prefix property of the Bitcoin backbone protocol. The bound of [9] is derived by the observation that (in our terminology) the adversary should produce a block for all rounds that are silent and leading branch. With this, it is shown that $\gamma_{\mathsf{u}} \geq \frac{f+\sqrt{f^2+4}}{2}\beta$ is sufficient for security; observe that in general the coefficient $\frac{f+\sqrt{f^2+4}}{2} > 1$ for any $f > 0$. Here we show that $\gamma \geq \beta$ is sufficient thus we eliminate entirely the dependence on f in the coefficient of β (also recall $\gamma \geq \gamma_{\mathsf{u}}$). This improvement in the bound has a significant impact in terms of provable security as shown in Figures 3,4.

Our main tool to derive this is a proof that *all* leading branch rounds have to be compensated by the adversary (and not just those that are silent). To show this we have to perform a more delicate analysis that requires some additional terminology. Next we introduce the notion of an m-Uniform round.

Definition 17. (m-Uniform rounds) We call a round m-Uniform if, at that round, m is the minimum value such that for all chains $\mathcal{C}_1, \mathcal{C}_2$ that any two honest parties initially invoke the *pow* algorithm

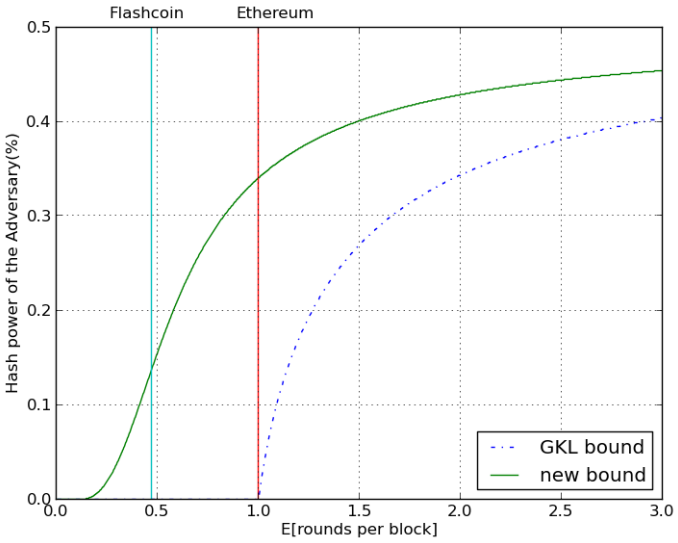

Figure 3: The level of provable security comparing the results of [9] and our improved results for Bitcoin. Under the curves the common prefix property provably holds. The respective block-rate values chosen for two altcoins are depicted on the graph.

with, it holds that $||\mathcal{C}_1| - |\mathcal{C}_2|| \leq m$.

Let base(r) denote the length of the shortest chain than an honest player at round r chooses to mine. From the definition of the m-uniform round it follows that on the next round, honest players will mine chains of size at least base(r) $+ m$. The size of these chains must also grow at least as much as the maximum number of solutions a single honest player has found at round r (recall that according to Definition 14 this is equal to $|v_{root}|^r_{\text{new}}$), because these solutions will be known to all players at round $r + 1$. More compactly:

Observation 18. For every m-uniform round r it holds that

$$base(r) + \max\{|v_{root}|^r_{\text{new}}, m\} \leq base(r + 1)$$

As it was discussed earlier, LB rounds are "bad" for the adversary, because they help honest players consent on a single blockchain in the following round. On the other hand, m-Uniform rounds are "good", since some honest players mine on shorter chains and thus waste their hash queries. Unfortunately for the adversary, this type of rounds does not happen naturally in the system and he must mine and publish blocks of his own to make a round non-uniform (m-uniform with $m > 0$). Independently of uniformity, the adversary must still compensate for all leading branch rounds as shown in the next lemma.

Lemma 19. *Suppose $\mathcal{C}_1$ and $\mathcal{C}_2$ are the chains of two honest parties at round r that diverge at round $s \leq r$. Also suppose that rounds $r_1, .., r_t$ are leading branch rounds such that $r_i \in LB^{\max}(s, d_i)$ and $r_i \in [s, .., r-1]$. Then, the adversary must have mined and published at least $\sum_{i=1}^{t} d_i$ different blocks until round r.*

Proof. For the m_i-uniform round r_i ($i \in \{1, .., t\}$), let $l_i = \text{base}(r_i)$ and $k_i = |v_{root}|^{r_i}_{\text{new}}$. For every such round, we prove that the adversary must have published at least d_i blocks and place them in specific positions in the respective chains, in order for the fork to be maintained. Formally we show the following.

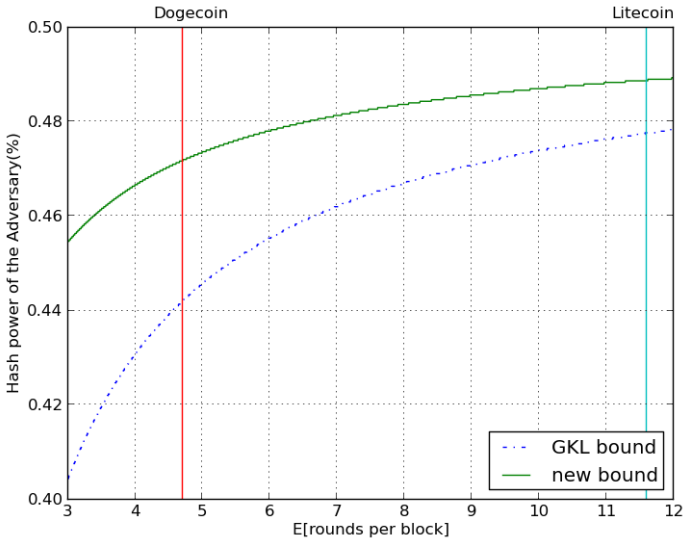

Figure 4: Similar to figure 3 but for larger values of $1/f$. Under the curves the common prefix property provably holds. The respective block-rate values chosen for two popular altcoins are depicted on the graph. Bitcoin is in the far right (recall from table 1 that for Bitcoin it holds $1/f \approx 47$).

Claim 1. *Let r be a $LB^{\max}(s,d)$ round that is m-uniform, with $s \leq r$, then:*

1. *if $m \geq 1$, there exists a chain $\mathcal{C}$ such that blocks at positions*

$$\text{base}(r)+1, ..., \text{base}(r)+m$$

are mined by the adversary.

2. *if $m < d$, at the end of round r and onwards and for all pairs of honest players' chains $\mathcal{C}_1, \mathcal{C}_2$ that diverge at round s, in each of the following positions there exists at least one adversarial block (in one of the two chains, as long as both chains have sufficient length):*

$$\text{base}(r)+m+(|v_{root}|^r_{\text{new}}-d)+1, ..., \text{base}(r)+|v_{root}|^r_{\text{new}}$$

Proof of Claim. The first point follows from the fact that all honest players mine a chain of size at least base(r). So for the round to be m-Uniform a chain of size at least base$(r)+m$ must exist. But honest players, at the start of round r, have mined blocks on chains of at most size base(r). Otherwise no honest player would choose to mine a chain with length base(r). So blocks at positions base$(r)+1, .., \text{base}(r)+m$ of the aforementioned chain must have been mined by the adversary.

For the second point, let $\mathcal{C}$ be a chain that an honest player extends by $|v_{root}|^r_{\text{new}}$ blocks at round r. By definition $\mathcal{C}$ by the end of the round has length at least $base(r)+|v_{root}|^r_{\text{new}}$. From the definition of the leading branch rounds we know that all honest players at round r, that extend a chain that diverges with $\mathcal{C}$ at round s, find at most $|v_{root}|^r_{new}-d$ solutions. Thus, by the end of the round, their chains have length at most base$(r)+m+|v_{root}|^r_{\text{new}}-d$. And so, all chains at the end of round r that have at least one honest block on the positions mentioned on the second point, do not diverge with $\mathcal{C}$ at round s (as they are longer than the upper bound we just established). And since no honest player is going to mine blocks in these positions in any following round, this

also holds for every round after r. It follows that any honest players' chain that diverges at round s with $\mathcal{C}$ has adversarial blocks at the positions mentioned.

Consider the chains $\mathcal{C}_1, \mathcal{C}_2$ of two honest players' at the end of round r and onwards that diverge at round s. For the sake of contradiction, assume that there is one position among those mentioned that both chains have blocks produced by honest players. In this case $\mathcal{C}_1, \mathcal{C}_2$ do not diverge with $\mathcal{C}$ at round s and thus they cannot diverge with each other at round s. This concludes the proof of the claim. ⊣

It remains to show that the blocks that the adversary must publish for every different leading branch round must be in distinct positions.

If $m_i \geq d_i$, from the previous claim, item 1, the adversary has published a chain where he has mined blocks at positions $l_i + 1, .., l_i + d_i$. On the other hand, if $m_i < d_i$ then, since $\mathcal{C}_1$ and $\mathcal{C}_2$ diverge at round s, and they have size greater or equal than $l_i + k_i$, the blocks at positions $l_i + m_i + k_i - d_i + 1, .., l_i + k_i$ cannot be both mined by honest players (due to the claim above, item 2). Moreover, there exists a chain where the adversary has mined blocks at positions $l_i + 1, .., l_i + m_i$ (due to claim above, item 1). Recall that $k_i \geq d_i$ hence these positions are disjoint and thus a total of $d_i = (k_i - (m_i + k_i - d_i + 1) + 1) + m_i$ blocks at least must have been mined and published by the adversary in the range $l_i + 1, .., l_i + k_i$.

Finally, from Observation 18 it holds that $l_i + \max(\{k_i, m_i\} \leq l_{i+1}$, and therefore all these blocks are on distinct positions on the chains they belong. Thus the lemma follows. □

Given the above core lemma we can now easily prove the improved bound for the common-prefix property following the same proof strategy as in [9]. Namely, it can be shown that the adversary cannot use very old solutions to compensate for recent leading branch rounds, and thus by suitably limiting his power he will be unable to produce enough solutions to compensate for every leading branch round, as it is required by the core lemma (proof in the Appendix).

Lemma 20. *Assume $\gamma \geq (1 + \delta)\beta$, for some real $\delta \in (0, 1)$. Suppose $\mathcal{C}_1$ and $\mathcal{C}_2$ are the chains of two honest parties at round r. Then, for any $s \leq r$, the probability that $\mathcal{C}_1$ and $\mathcal{C}_2$ diverge at round $r - s$ is at most $e^{-\Omega(\delta^3 s)}$.*

Theorem 21. *Assume $\gamma \geq (1 + \delta)\beta$, for some real $\delta \in (0, 1)$. Let S be the set of the chains of the honest parties at a given round of the backbone protocol. Then the probability that S does not satisfy the common-prefix property with parameter k is at most $e^{-\Omega(\delta^3 k)}$.*

5 GHOST

In this section, we prove that the GHOST backbone protocol is sufficient to construct a robust transaction ledger. Whenever notation from Definition 14 is used, it is assumed that $g(F)$ is the total number of nodes of the forest F.

5.1 Common Prefix and Chain Quality

In the previous section, it was shown that the effort that leading branch rounds impose on the adversary is cumulative. A similar idea is developed here but a different approach is needed. The reason that the previous analysis cannot be used for GHOST, is that the blocks that the adversary mines to compensate for leading branch rounds, are not uniquely associated with a specific height in the chain, as it was the case for Bitcoin. Moreover, in GHOST, honest players can choose to mine smaller chains than the ones that they were mining previously hence the length of the chain

is not monotonically increasing. To reflect these facts, we introduce a new notion, that of a path that all of its nodes are dominant up to a certain value.

Definition 22. ($\mathrm{p}_{\mathrm{dom}}(r,d)$) For $d > 0$, $\mathrm{p}_{\mathrm{dom}}(r,d)$ is the longest subpath $p = v_{\mathrm{root}}v_1 \dots v_k$ in T_r^{tot} s.t.

$$p \neq v_{\mathrm{root}} \wedge \forall i: \ \mathrm{dom}_{\mathrm{tot}}^r(v_i, d)$$

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

In the next lemma, we show that for any sequence of m leading branch rounds starting at some round r', no matter the strategy of the adversary, there will be at least one honest block at round r in $\mathrm{p}_{\mathrm{dom}}(r, m-k)$ where k is the number of adversarial blocks that have been released during rounds $[r', r-1]$. This establishes the robustness of $\mathrm{p}_{\mathrm{dom}}$ in the sense that only adversarial blocks can decrease it and they do so in a linear fashion at worst.

Lemma 23. *Let rounds $r_1, .., r_m$ be uniquely successful rounds from round r' until round r. If the adversary publishes $k < m$ blocks from round r' until round r, then there exist blocks $b_1, .., b_{m-k}$ mined by honest players at the uniquely successful rounds where (1) b_i is in $\mathrm{p}_{\mathrm{dom}}(r,i)$ and (2) if $i < j$ then b_i is a descendant of b_j.*

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 2. *Let round s be a uniquely successful round and b be the honest block mined at round s. If the adversary does not publish 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 publish any block it holds that $T_s^{\exists} = T_s^{\forall}$. Therefore, all nodes in the path from v_{root} to b are at least 0-dominant w.r.t. to old. For any uniquely successful round it holds that all nodes up to the newly mined block are 1-dominant w.r.t. new. Thus 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 publishes less than d blocks the same nodes continue to be at least 1-dominant in the following round.

Claim 3. *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 publishes (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 $\mathrm{p}_{\mathrm{dom}}(s-1,d) \subseteq \mathrm{p}_{\mathrm{dom}}(s,1)$.*

Proof of Claim. There are two cases. In the first case suppose the adversary publishes $k < d$ blocks. Then with these blocks the adversary can lower the dominance of nodes in $\mathrm{p}_{\mathrm{dom}}(s-1,d)$ by k. Thus $\mathrm{p}_{\mathrm{dom}}(s-1,d)$ will be a prefix of all the chains in $\mathrm{HonestPaths}(s, v_{\mathrm{root}})$. But because s is a uniquely successful round the dominance of all nodes in $\mathrm{p}_{\mathrm{dom}}(s-1,d)$ w.r.t. tot at round s will increase by one. Therefore $\mathrm{p}_{\mathrm{dom}}(s-1,d) \subseteq \mathrm{p}_{\mathrm{dom}}(s, d+1-k)$.

In the second case suppose the adversary publishes $k = d$ blocks. If he does not publish all of these blocks to reduce the dominance of nodes in path $\mathrm{p}_{\mathrm{dom}}(s-1,d)$, then $\mathrm{p}_{\mathrm{dom}}(s-1,d)$ will be a prefix of all the chains in $\mathrm{HonestPaths}(s, v_{\mathrm{root}})$ and as in the previous case, $\mathrm{p}_{\mathrm{dom}}(s-1,d) \subseteq \mathrm{p}_{\mathrm{dom}}(s, d+1-k)$.

Otherwise the adversary will reduce the dominance of at least one node in $\mathrm{p_{dom}}(s-1,d)$ to zero. If b is a descendant of the last node in $\mathrm{p_{dom}}(s-1,d)$, then all nodes in $\mathrm{p_{dom}}(s-1,d)$ will be 1-dominant w.r.t. tot and $\mathrm{p_{dom}}(s-1,d) \subseteq \mathrm{p_{dom}}(s,1) = \mathrm{p_{dom}}(s,d+1-d)$. If b is not a descendant of the last node in $\mathrm{p_{dom}}(s-1,d)$, 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 $\mathrm{p_{dom}}(s-1,d)$ at round s otherwise. Therefore, P at round s was mining a chain that belonged to $\mathrm{HonestPaths}(s, v_{\mathrm{root}})$ and thus all nodes in the chain are at least 0-dominant w.r.t. old. But because s is a uniquely successful round the dominance of all nodes in the chain will increase by one and $b \in \mathrm{p_{dom}}(s,1)$. $\dashv$

Let b_i denote one of the blocks mined by honest players 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 publish any block until round r_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 published by the adversary in the round interval $[r', r_{m-1}-1]$. We have two cases.

(First case) $k_1 = m-1$ and the adversary publishes 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 $b'_i \in \mathrm{p_{dom}}(r_{m-1}, i)$. If the adversary publishes $m-1-k_1$ new blocks before round $r_m - 1$, then from the first claim, $b_m \in \mathrm{p_{dom}}(r_m, 1)$. If the adversary publishes $k_2 < m-1-k_1$ before round $r_m - 1$, then from the second claim, at round $r_m - 1$, $b'_i \in \mathrm{p_{dom}}(r_m - 1, i - k_2)$ for i in $\{k_2+1, ..., m-1-k_1\}$.

Let k_3 be the number of blocks the adversary publishes at round $r_m - 1$. If $k_3 = m-1-k_1-k_2$ 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 $k_3 < m-1-k_1-k_2$ then again from the second claim at round $r_m - 1$, $b'_i \in \mathrm{p_{dom}}(r_m -, i - k_2 - k_3 + 1)$ 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 has published until round $r = r_m$, 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 publishes in the remaining rounds. The lemma follows easily from the second claim.

$\square$

In the next lemma we prove that after a fixed amount of consecutive rounds, one honest block mined on these rounds, is permanently inserted in the chain that every honest player chooses to mine thereafter with overwhelming probability on s.

Lemma 24. *Assume $\gamma \geq (1+\delta)\beta$, for some real $\delta \in (0,1)$. For any sequence of s consecutive rounds that happen before some round r, there exists a block mined by an honest player during these rounds that is contained in the chain which any honest player chooses to mine after round r with probability $1 - e^{-\Omega(\delta^2 s)}$.*

Proof. Let random variable $Z_{r'}$ denote the number of blocks the adversary produces during r' rounds, and random variable $X_{r'}$ denote the number of rounds that are uniquely successful during r' rounds. Then from the Chernoff bounds we have:

$$Pr[Z_{r'} \geq (1+\frac{\delta}{5})\beta r'] \leq e^{-\Omega(\delta^2 s)}, \text{for } \delta \in (0,1)$$
$$Pr[X_{r'} \leq (1-\frac{\delta}{4})\gamma r'] \leq e^{-\Omega(\delta^2 s)}, \text{for } \delta \in (0,1)$$

It follows that with probability at least $1 - e^{-\Omega(\delta^2 s)}$:

$$X_s > (1 + \frac{\delta}{2})Z_s$$

and thus from Lemma 23 at round s one honest block b will be in $\mathrm{p}_{\mathrm{dom}}(s, X_s - Z_s)$ with probability at least $1 - e^{-\Omega(\delta^2 s)}$.

Next we show that as long as for any round $r' > s$ the adversary has produced less blocks than the number of uniquely successful rounds $\mathrm{p}_{\mathrm{dom}}(s, X_s - Z_s) \subseteq \mathrm{p}_{\mathrm{dom}}(r', X_{r'} - Z_{r'})$.

Claim 4. *Let rounds $r_1, .., r_m$ be uniquely successful rounds in the round interval $[r' + 1, r]$ and $\mathrm{p}_{\mathrm{dom}}(r', d) \neq \perp$. If for all uniquely successful rounds r_i it holds that the adversary has published $k_i < i + d$ blocks and the adversary publishes at most $k < m + d$ blocks from round r' until round r, then $\mathrm{p}_{\mathrm{dom}}(r', d) \subseteq \mathrm{p}_{\mathrm{dom}}(r, m + d - k)$.*

Proof of Claim. As long as the nodes in $\mathrm{p}_{\mathrm{dom}}(r', d)$ are at least 1-dominant, all honest players will work on chains containing $\mathrm{p}_{\mathrm{dom}}(r', d)$ and thus uniquely successful rounds will increase their dominance. On the other hand the adversary can at worst reduce the dominance of these nodes by the number of blocks he publishes. But from the assumptions made in the statement the number of the blocks the adversary publishes is always less than the number of uniquely successful rounds plus d. Therefore in all rounds the nodes in $\mathrm{p}_{\mathrm{dom}}(r', d)$ are at least 1-dominant and the claim follows. $\dashv$

Notice that for all subsequent rounds r' after s it will hold with probability at least $1 - e^{-\Omega(\delta^2 s)}$ that

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

Thus b will stay in the chains of honest players permanently after round s, since $b \in \mathrm{p}_{\mathrm{dom}}(r', 1)$ for any $r' > s$, with probability $1 - e^{-\Omega(\delta^2 s)}$.

We can use this argument inductively for every round of the form $s \cdot k$ where $k \in \mathbb{N}$. Suppose that block b_k has been added permanently to the chains of honest players at round $s \cdot k$. Then for all uniquely successful rounds after $s \cdot k$, the path to the newly mined block contains b_k and thus Lemma 23 holds for the subtree under b_k. Everything stated in the proof of the base cases holds for the round interval $[s \cdot k, s \cdot (k + 1)]$ also. Therefore another block b_{k+1} will be added permanently to the chains of honest players at round $s \cdot (k + 1)$ and the lemma follows by induction.

□

From Lemma 24 it follows that the density in terms of rounds of honest blocks in any chain that an honest player chooses to mine is at least $\frac{1}{s}$ with probability $1 - e^{-\Omega(\delta^2 s)}$. Since in s rounds the adversary can compute only a limited number of blocks the chain quality property follows (proof in the Appendix).

Theorem 25. *Assume $\gamma \geq (1 + \delta)\beta$, for some real $\delta \in (0, 1)$. Suppose $\mathcal{C}$ is the chain of an honest party at round r. Then it holds that for any l consecutive blocks of $\mathcal{C}$, there exists at least one honest block with probability $1 - e^{-\Omega(\delta^2 l)}$.*

From Lemma 24 again it follows that all honest players at round r will share in the chains they choose to mine a block computed at worst at round $r - s$ with probability $1 - e^{-\Omega(\delta^2 s)}$. Since in s rounds all players can compute only a limited number of blocks the common prefix property follows. The proof is the same as the one that was presented in the Appendix for theorem 21.

Theorem 26. *Assume $\gamma \geq (1 + \delta)\beta$, for some real $\delta \in (0, 1)$. Let S be the set of the chains of the honest parties at a given round of the GHOST backbone protocol. Then the probability that S does not satisfy the common-prefix property with parameter k is at most $e^{-\Omega(\delta^2 k)}$.*

5.2 Chain Growth

In this section we prove that GHOST satisfies the Chain Growth property. However, in comparison with Bitcoin, the speed coefficient of GHOST is a lot weaker. This reflects the fact that honest players in GHOST may be lead by the adversary to adopt shorter chains and hence honest players' chains are not monotonically increasing (cf. Section 6.2 where we describe a chain growth attack against GHOST).

Theorem 27. *Assume $\gamma \geq (1+\delta)\beta$, for some real $\delta \in (0,1)$. The GHOST backbone protocol satisfies the chain growth property at least with parameters $\tau = \frac{1}{k(k+1)}$, $s = k(k+1)$ with probability at least $1 - e^{-\Omega(\delta^2 k)}$.*

Proof. We prove that after $k \cdot (k+1)$ rounds the block chain of every player will have grown by at least one block. Let C_1 be the chain of player p at round r and C_2 at round $r + k(k+1)$.

From theorem 26 it follows that the prefix $C_1^{\lceil k}$ of C_1 will be also a prefix for C_2 with probability $1-e^{-\Omega(\delta^2 k)}$. Also, from lemma 24 it follows that for all round intervals of the form $[r+ik, r+(i+1)k]$ for $i \in \{0, k\}$, there exists at least one block in chain $\mathcal{C}_2$ that was computed on this interval by an honest player with probability at least $1 - e^{-\Omega(\delta^2 k)}$. By an application of the union bound with probability $1 - e^{-\Omega(\delta^2 k)}$ there are a total of $k+1$ new blocks in $\mathcal{C}_2$ and $C_1^{\lceil k}$ is a prefix of C_2. Thus $|C_2| - |C_1| \geq 1 \geq \frac{1}{k(k+1)} k(k+1)$ and the GHOST protocol satisfies the chain growth property with probability $1 - e^{-\Omega(\delta^2 k)}$.

□

5.3 Robust public transaction ledger

It was proved on the previous subsections that the GHOST backbone protocol satisfies all three security properties: Common Prefix, Chain Quality and Chain Growth. As it was shown in [9] and further discussed on lemma 12, using arbitrary protocols that satisfy these properties one can implement in a black box manner a robust public transaction ledger through protocol Π_{PL}. Thus the GHOST backbone protocol can be used to implement a robust transaction ledger. The security parameters under which the ledger works are described in the next two lemmas.

Lemma 28 (Persistence)**.** *Suppose $\gamma \geq (1+\delta)\beta$, for some real $\delta \in (0,1)$. Protocol Π_{PL} satisfies Persistence with probability $1 - e^{-\Omega(\delta^2 k)}$, where k is the depth parameter.*

Lemma 29 (Liveness)**.** *Assume $\gamma \geq (1+\delta)\beta$, for some $\delta \in (0,1)$ and let $k \in \mathbb{N}$. Further, assume oracle* Txgen *is unambiguous. Then protocol Π_{PL} satisfies Liveness with wait time $u = 2k^2(k+1)$ rounds and depth parameter k with probability at least $1 - e^{-\Omega(\delta^2 k)}$.*

Theorem 30. *The GHOST backbone protocol can be used to implement a robust transaction ledger.*

6 Transaction Speed - Security Tradeoffs

In [8] an attack (selfish mining) against the chain quality property of Bitcoin was demonstrated. In [9] it was shown that (for the case of a rushing adversary[7]) it is optimal since it matches the bounds of the security theorem for chain quality. However little is known regarding optimal attacks on the common prefix and chain growth properties. For instance, it is known that a "51% attacker" can break the common prefix with an arbitrarily long fork. However long forks have been predicted

[7] As argued in [8] this is a plausible attack strategy, we refer to their paper for more details.

to be feasible even for attackers with below 50% of the hashing power in case f is large. In this section, we explore two attacks on these two properties in an experimental way (through computer simulations) providing some interesting insights on the optimality of the theoretical results that we have proven. Both of the attacks affect transaction speed in different ways. The first attack, targets security when f is large, and thus prohibits the increase of the block generation rate in order to increase the transaction speed. The second attack, targets chain growth and thus effectively it will increase the time that a certain transaction may appear in the blockchain.

6.1 Attack on Common Prefix

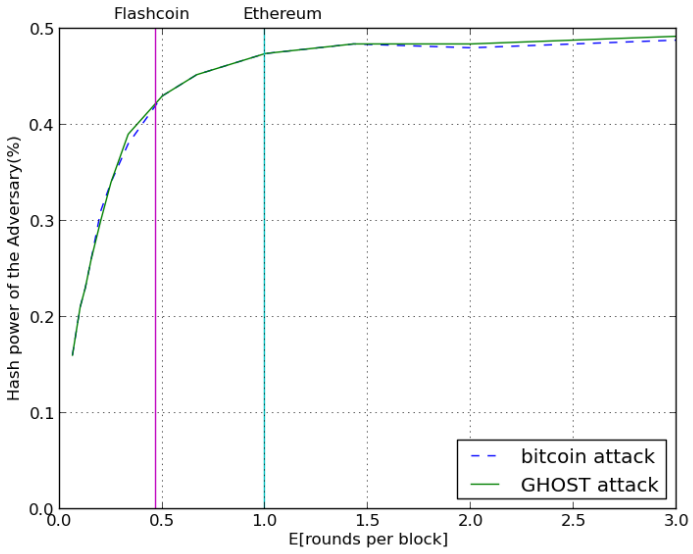

Figure 5: The level of insecurity in terms of the hashing power of the adversary as a function of $1/f$. Above the two (almost identical) curves our attack breaks common prefix with a fork that is 100 blocks deep with probability of success at least 1%. The respective block-rate values chosen for two altcoins are depicted on the graph.

In this subsection we do experimental analysis on attacks targeting the common prefix property of the Bitcoin and GHOST protocols. The two protocols seem quite robust against these attacks when $f < 1$. However, their security deteriorates as f grows bigger and taking advantage of these attacks an adversary can effectively cause deep forks to appear. Graphs on how various cryptocurrencies' (that use different parameterizations of the Bitcoin and GHOST backbone protocols) fare in terms of the attacks are also presented.

The idea of the attacks is the following: when a fork of depth 1 naturally happens, the adversary splits its hashing power, as well as the honest players' power, on the two branches. In our model this is possible because we consider the adversary to be rushing.

On Bitcoin, when an honest player in one of the two branches publishes a new solution, then the adversary also publishes one of its solutions (if he has any) on the other branch. If honest players extend both branches by the same length in the same round, then the adversary just reschedules the messages so again players are split in half. Otherwise, if possible, the adversary lengthens the chain that is behind by the same amount of blocks, to keep the fork running. Additionally, even if players modified the backbone protocol to use "flip a coin" in order to resolve ties, cf. [8], they

would have 0.5 probability to go in one of the two branches, so adding randomness does not seem to help against this attack.

The GHOST attack proceeds in the same way with two big differences. First, the adversary has to pay for the absolute difference of the total number of solutions released on the two branches by the honest players in each round (instead of the max that he paid for Bitcoin). Secondly, the solutions that are produced by the adversary are never invalid. He just mines the first nodes after the common prefix of the two branches and the blocks that are produced cannot be invalidated. In contrast, solutions that are produced for Bitcoin must always extend the head of any of the two diverging chains to be useful. Thus, the blocks used by the adversary must be recent.

Most interestingly these two attacks have almost the same effectiveness on both protocols as shown in figure 5. For $f < 1$ both protocols tolerate this type of attack and achieve an almost optimal level of security. But when f grows larger than 1, security deteriorates in a surprisingly similar rate. This result suggests that paying for the difference of the sum of new blocks in the two branches and paying for the difference between the maximum chains on the two branches with recent blocks seems to be equally hard for the adversary.

In the graphs we also present the specific choices made by various altcoins that were reported in table 1. It is interesting to point out that for the choice made in Ethereum[8] ($f = 1$) our provable security bound is around 35% while for Dogecoin and Litecoin our improved analysis brings the provable security bound to a relatively satisfactory level of over 47%. Extreme choices such as Flashcoin cannot be supported at all by the security analysis, while Bitcoin on the other end of the spectrum opts for the safest choice that enables a near optimal provable security bound of about 49%. We remark that the original proposal for GHOST for a 1 sec per block [14] yields an $1/f = 1/12$ which is in a completely precarious region of the speed-security domain (note it was subsequently amended to $f = 1$).

6.2 Attack on Chain Growth

Chain growth is closely related to transaction processing speed. Slow chain growth implies a low number of transactions per second. Also as proved in lemma 12, chain growth is closely related to the confirmation time of transactions.

In this subsection an attack on the chain growth of GHOST is presented and experimentally tested. This attack exploits the fact that in GHOST, thin and long trees may have the same or less weight than short and wide trees. 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 it can publish it and following the GHOST rule force the honest players to switch to a shorter main 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 as shown in Figure 6, that increases as the adversaries power increases. An interesting feature of the attack is that it gets better as f becomes smaller. A short description of the attack is given in Algorithm 2.

On the other hand we observe that the optimal attack for Bitcoin is quite trivial (see Section 3.1) and much less effective. Since GHOST under this attack is slower than Bitcoin under an optimal attack, we conclude that the chain growth parameter for GHOST should be smaller than that of Bitcoin's, and thus GHOST is sub-optimal in terms of the chain growth property (also note that our provable lower bound is also worse than that of Bitcoin, cf. Theorem 5.2). This conclusion

[8]Note that Ethereum is not yet in production stage and several variants of GHOST have been implemented. Our results refer to the original proposal of [14], but with 12 seconds block generation rate as discussed by Buterin in [5]. Our framework can be used to further explore design alterations of the original GHOST rule.

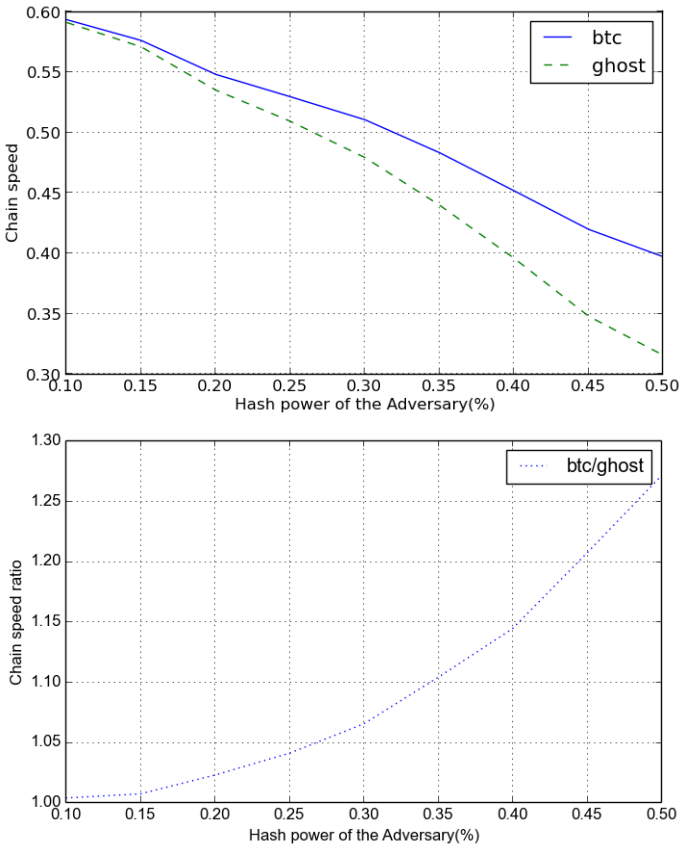

Figure 6: Chain speed from experimental analysis for $f = 1$. Note that as the hashing power of the adversary increases both Bitcoin and GHOST speed decrease. However, Bitcoin is clearly favorable to GHOST (upper graph) and in fact the ratio of Bitcoin to GHOST chain speed increases (lower graph).

debunks the suggestion of [14] that the difference of GHOST chain speed compared to Bitcoin is relatively small (cf. Figure 4 of [14] where it is suggested that the difference in speed between the two is small based on experiments without adversarial interference), and hence sheds light to a first noticeable shortcoming of the GHOST backbone in terms of chain growth.

Together with the attack on common prefix, we have attacks in the whole range of the spectrum for the GHOST backbone: for small values of f, chain growth can be made almost 10% less than that of Bitcoin (cf. Figure 6), while for bigger values of f security can be broken in terms of the common prefix (cf. Figure 5), for attackers with less that 35% of the total computation power.

7 Conclusion

In this paper we presented a new framework for analyzing backbone protocols based on trees and we showed its power by substantially improving the security bounds of the Bitcoin backbone, [9], and analyzing the GHOST backbone protocol (that we extract and formalize herein). Within this framework we presented a formal treatment of transaction processing speed of blockchain-based protocols focusing on the Bitcoin and GHOST backbone. With our chain growth definition we introduced a measure of speed, called the chain speed coefficient, and we showed that the

Algorithm 2 The algorithm of the adversary on the chain growth attack.

1: $\langle t_H, t_A \rangle \leftarrow \langle 0, 0 \rangle$ ▷ The weight of the competing trees.
2: Update the block tree
3: $\mathcal{C} \leftarrow argmin_{\mathcal{C} \in \text{HonestPaths}} |C|$
4: Mine head($\mathcal{C}$)
5: **if** |blocks mined| $= 0$ **then**
6: **go to** 1
7: **else**
8: $b \leftarrow$ newly mined block ▷ The head of the short tree.
9: Mine b
10: **end if**
11: **while** $t_H < 6$ **do**
12: Update the block tree
13: $\langle t_H, t_A \rangle \leftarrow \langle t_H +$ new honest blocks, $t_A +$ new adversarial blocks$\rangle$
14: **if** $(t_A > t_H)$ and (length of honest subtree ≥ 3) **then**
15: Broadcast $subtree(b)$
16: $\langle t_H, t_A \rangle \leftarrow \langle 0, 0 \rangle$
17: **go to** 1
18: **end if**
19: Mine b
20: **end while**

chain growth property (with a non-zero coefficient) is a fundamental security property of a robust transaction ledger.

Quantifying over all possible adversaries, transaction processing speed can be equated to the rate of blocks mined by honest players that are inserted on the common chain, since malicious blocks are not guaranteed to contain any honest transactions. We proved that as long as the common prefix property holds, this rate is proportional to the product $(1-\mu)\tau$, where μ is the chain quality parameter (as defined in [9]) and τ is our chain speed coefficient. Further we prove that a minimum transaction processing speed is guaranteed for both Bitcoin and GHOST.

Our formalization and study of the chain growth property yields a number of concerns about GHOST transaction processing speed compared to Bitcoin, contrary to some extensively cited claims to the contrary (see e.g. [4]). On the Bitcoin backbone, the chain speed coefficient τ is guaranteed to be at least γ (cf. Section 2 for the definition of this parameter), and as a result transaction processing speed is at least $\gamma - \beta$ (since, due to selfish mining, the chain quality parameter μ can be as big as β/γ and thus $\gamma(1 - \beta/\gamma) = \gamma - \beta$). However, regarding the GHOST backbone, the coefficient τ is not as large as γ. In fact, we have presented an attack that demonstrates that the GHOST chain speed coefficient is strictly smaller than that of Bitcoin's and in this way an adversary can achieve a significant reduction to GHOST transaction processing speed by mounting it. We note that there is still a gray area in determining the exact transaction processing speed of GHOST (as our lower bounds for both chain quality and the chain speed coefficient are not tight) and hence future work in determining GHOST's transaction processing speed should take into account both properties. Our work lays the theoretical foundation for such study.

References

[1] 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/`.

[2] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. Cryptology ePrint Archive, Report 2014/349, 2014. `http://eprint.iacr.org/`.

[3] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, May 2015.

[4] V. Buterin. Ethereum: A next-generation cryptocurrency and decentralized application platform, January 2014.

[5] V. Buterin. Toward a 12-second block time, July 2014. [Online; Posted: July 11th, 2014.].

[6] G. Caffyn. What is the bitcoin block size debate and why does it matter?, August 2015. [Online; posted 21 August 2015].

[7] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *P2P*, pages 1–10. IEEE, 2013.

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

[9] 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.

[10] 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/`.

[11] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 Internet Measurement Conference, IMC 2013, Barcelona, Spain, October 23-25, 2013*, pages 127–140, 2013.

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

[13] Y. B. Perez. Bitcoin in the headlines: Fork-load of drama, August 2015. [Online; posted 21 August 2015].

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

[15] B. Wiki. Block size limit controversy, August 2015. [Online; Accessed: 24 August 2015].

A Probability of Leading branch rounds

Lemma 31. *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 γ is at most:

$$(n-t)qp(1-p)^{q(n-t)-1} \geq ae^{-a-p} \geq \gamma$$

This is also a lower bound for the event that at least one honest party computes a solution in a round, and also that either for GHOST or bitcoin a leading branch round happens, since uniquely successful rounds are also leading branch rounds. So $\gamma = ae^{-a-p}$.

B Proofs

B.1 Lemma 20

Proof. We define three bad events, A, B and C, which we show to hold with probability exponentially small in s. We conclude the proof by showing that if none of these bad events happens, then there cannot exist $\mathcal{C}_1$ and $\mathcal{C}_2$ diverging at round $r - s$.

The bad event A occurs if, at some round $r' \geq r - s$, the adversary broadcasts a chain $\mathcal{C}$ with the following properties. (1) $\mathcal{C}$ is returned by the function maxvalid of an honest party; (2) the block head($\mathcal{C}$) was computed by the adversary before round $r - (1 + \frac{\delta}{8})s$.

We now give an upper bound on the probability that event A occurs. Let $r^* \leq r-(1+\frac{\delta}{8})s$ be the latest round at which a block of $\mathcal{C}$ was computed by an honest party (if none exists, then $r^* = 0$), and let ℓ denote the length of the chain up to that block. If any other block computed by an honest party exists among the blocks from length ℓ up to len($\mathcal{C}$), then such block was computed in rounds $r - (1 + \frac{\delta}{8})s$ up to r', and it follows that the probability that the adversary's block can extend it at round r' is negligible in $(\kappa - \log D)$. Therefore, we infer that with overwhelming probability the adversary has computed all the blocks from length ℓ to len($\mathcal{C}$), and done so during the rounds r^* to r'. Let Z denote the total number of solutions the adversary obtained in $r' - r^*$ rounds. Let also X denote the total number of successful rounds for the honest parties in $r' - r^*$ rounds. We have

$$Z \geq \text{len}(\mathcal{C}) - \ell \geq X.$$

The first inequality was argued above and the second one follows from [9, Lemma 5]. Finally, note that, by Lemma [9, Lemma 6], the event $Z \geq X$ has measure exponentially small in the number of rounds $r' - r^*$. Since that number satisfies $r' - r^* \geq \delta s/8$, we conclude that $\Pr[A] \leq e^{-\Omega(\delta^3 s)}$.

The second bad event occurs if the adversary has obtained a large number of solutions during $(1 + \frac{\delta}{8})s$ rounds. Specifically, let Z denote the number of successful calls to the oracle by the adversary, for a total of $(1 + \frac{\delta}{8})s$ rounds. Define B to be the event $Z \geq (1 + \frac{\delta}{9})(1 + \frac{\delta}{8})\beta s$. An application of Chernoff bounds gives

$$\Pr[Z \geq (1 + \tfrac{\delta}{9})(1 + \tfrac{\delta}{8})\beta s] \leq e^{-\Omega(\beta\delta^2 s)}.$$

The third bad event occurs when not enough leading branch rounds occur. Consider any number, say, s' of rounds, and denote by X' the number of them that were leading branch. We have

$$\Pr[X' \leq (1-\tfrac{\delta}{4})\gamma s'] \leq e^{-\Omega(\gamma\delta^2 s')}.$$

From now on we assume that none of the events A, B and C occurs. From lemma? , it is easy to see that the adversary has to compute at least $\sum_{i=1}^{k} d_i$ solutions, where $r_1, ..., r_k$ are $LB^{\max}(s, d_i)$ rounds such that $s \leq r_i \leq r$. Since a round is $LB^{\max}(s, d)$ with probability γ, from the negation of the third bad event we expect at least $(1-\frac{\delta}{4})\gamma s$ such rounds.

Note that, since A does not occur, the adversary may not use solutions computed before round $r-(1+\frac{\delta}{8})s$ with probability at least $1-e^{-\Omega(\delta^3 s)}$. The negation of the second bad event bounds the number of solutions the adversary can obtain. Thus from lemma 19 it has to hold with probability at least $1-e^{-\Omega(\delta^3 s)}$:

$$\begin{gathered}(1-\tfrac{\delta}{4})\gamma s \leq (1+\tfrac{\delta}{9})(1+\tfrac{\delta}{8})\beta s \Leftrightarrow \\ (1-\tfrac{\delta}{4})\gamma \leq (1+\tfrac{\delta}{9})(1+\tfrac{\delta}{8})\beta \Rightarrow \\ (1-\tfrac{\delta}{4})(1+\delta)\beta \leq (1+\tfrac{\delta}{9})(1+\tfrac{\delta}{8})\beta \Rightarrow \\ (1-\tfrac{\delta}{4})(1+\delta) \leq (1+\tfrac{\delta}{9})(1+\tfrac{\delta}{8})\end{gathered}$$

But the last inequality does not hold for $\delta \in (0,1)$. We conclude that if $A \cup B \cup C$ does not occur, then $\mathcal{C}_1$ and $\mathcal{C}_2$ cannot diverge at round $r-s$. Finally, an application of the union bound on $A \cup B \cup C$ implies that the adversary can successfully maintain such $\mathcal{C}_1$ and $\mathcal{C}_2$ with probability at most exponentially small in s and the statement of the lemma follows. □

B.2 Theorem 21

Proof. If there is only one chain in $\mathcal{S}$ then the property is satisfied trivially. Consider two chains $\mathcal{C}_1$ and $\mathcal{C}_2$ in $\mathcal{S}$ and the least integer k^* such that

$$\mathcal{C}_1^{\lceil k^* \rceil} \preceq \mathcal{C}_2 \text{ and } \mathcal{C}_2^{\lceil k^* \rceil} \preceq \mathcal{C}_1. \tag{1}$$

We need to show that the event $k^* \geq k$ happens with probability exponentially small in k.

Let r be the current round and let $r-s$ be the round at which the last common block of $\mathcal{C}_1$ and $\mathcal{C}_2$ was computed. The length of the chains cannot be greater than the number of solutions Y obtained from the oracle in s rounds. By the Chernoff bound,

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

It follows that, with probability $1-e^{-\delta^2 fs/3}$, $s > k^*/((1+\delta)f)$. Thus, if $k^* \geq k$, we have a sequence of $s = \Omega(k)$ consecutive rounds with chains $\mathcal{C}_1$ and $\mathcal{C}_2$ diverging, and the theorem follows from Lemma 20 □

B.3 Theorem 25

Proof. Let $r-s$ be the round that the last honest block b_0 in $\mathcal{C}$ was computed and l^* be the number of blocks from b_0 until the head of $\mathcal{C}$. The number of blocks in $\mathcal{C}$ after b_0 cannot be greater than the number of solutions Y obtained from the oracle in s rounds. By the Chernoff bound,

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

It follows that, with probability $1 - e^{-\delta^2 fs/3}$, $s > l^*/((1+\delta)f)$. Thus, if $l^* \geq l$, we have a sequence of $s = \Omega(l)$ consecutive rounds where no honest block in $\mathcal{C}$ has been computed during these rounds with probability $1 - e^{-\Omega(\delta^2 l)}$, which is a contradiction to Lemma 24.

The same argument can be used for the last honest block b_1 before b_0 with the same probability. In this way a sequence of blocks mined by honest players $b_0, b_1, ..., b_k$ is defined with the property that any l consecutive blocks of $\mathcal{C}$ contain at least one of these blocks. By applying the union-bound on the set of events where, for $i < k$, the sequence of blocks from b_{i+1} until b_i (or the head of $\mathcal{C}$ for i equal to zero or v_{root} for i equal to k) has length more than l, each happening with probability $e^{-\Omega(\delta^2 l)}$, the theorem follows. □

C The GHOST protocol

Algorithm 3 The *proof of work* function, parameterized by q, D and hash functions $H(\cdot), G(\cdot)$. The input is $(x, \mathcal{C}, T)$.

1: **function** pow$(x, \mathcal{C}, T)$
2: $\langle s', x', ctr' \rangle \leftarrow \text{head}(\mathcal{C})$
3: $s \leftarrow H(ctr', G(s', x'))$
4: $ctr \leftarrow 1$
5: $B \leftarrow \varepsilon$
6: $h \leftarrow G(s, x)$
7: **while** $(ctr \leq q)$ **do**
8: **if** $(H(ctr, h) < D)$ **then** ▷ Proof of work succeeded
9: $B \leftarrow \langle s, x, ctr \rangle$
10: $\mathcal{C} \leftarrow \mathcal{C}B$ ▷ Extend chain
11: $\langle s', x', ctr' \rangle \leftarrow \text{head}(\mathcal{C})$
12: $s \leftarrow H(ctr', G(s', x'))$
13: $h \leftarrow G(s, x)$
14: **end if**
15: $ctr \leftarrow ctr + 1$
16: **end while**
17: $T \leftarrow$ update(T, $\mathcal{C}$ as separate blocks) ▷ Add new blocks to the tree
18: **return** $\langle T, \mathcal{C} \rangle$
19: **end function**

Algorithm 4 The GHOST backbone protocol, parameterized by the *input contribution function* $I(\cdot)$ and the *reading function* $R(\cdot)$.

$T \leftarrow GenesisBlock$
$\mathcal{C} \leftarrow GenesisBlock$
$st \leftarrow \varepsilon$
$round \leftarrow 0$
while TRUE **do**
    $T_{new} \leftarrow \mathsf{update}(\mathsf{T}, \text{blocks found in } \textsc{Receive}())$
    $\tilde{\mathcal{C}} \leftarrow \mathsf{GHOST}(T_{new})$
    $\langle st, x \rangle \leftarrow I(st, \tilde{\mathcal{C}}, round, \textsc{Input}(), \textsc{Receive}())$ ▷ Determine the x-value.
    $\langle T_{\mathsf{new}}, \mathcal{C}_{\mathsf{new}} \rangle \leftarrow \mathsf{pow}(x, \tilde{\mathcal{C}}, T)$
    if $\mathcal{C} \neq \mathcal{C}_{\mathsf{new}}$ or $T \neq T_{\mathsf{new}}$ **then**
        $\mathcal{C} \leftarrow \mathcal{C}_{\mathsf{new}}$
        $T \leftarrow T_{\mathsf{new}}$
        $\textsc{Broadcast}(T \text{ as separate blocks})$
    end if
    $round \leftarrow round + 1$
    if $\textsc{Input}()$ contains $\textsc{Read}$ **then**
        write $R(\mathbf{x}_{\mathcal{C}})$ to $\textsc{Output}()$
    end if
end while