GHAST: Breaking Confirmation Delay Barrier in Nakamoto Consensus via Adaptive Weighted Blocks

Chenxing Li
Tsinghua University

Fan Long
University of Toronto

Guang Yang
Conflux Foundation

Abstract

Initiated from Nakamoto's Bitcoin system, blockchain technology has demonstrated great capability of building secure consensus among decentralized parties at Internet-scale, i.e., without relying on any centralized trusted party. Nowadays, blockchain systems find applications in various fields. But the performance is increasingly becoming a bottleneck, especially when permissionless participation is retained for full decentralization.

In this work, we present a new consensus protocol named GHAST (Greedy Heaviest Adaptive Sub-Tree) which organizes blocks in a Tree-Graph structure (i.e., a directed acyclic graph (DAG) with a tree embedded) that allows fast and concurrent block generation. GHAST protocol simultaneously achieves a logarithmically bounded liveness guarantee and low confirmation latency. More specifically, for maximum latency d and adversarial computing power bounded away from 50%, GHAST guarantees confirmation with confidence $\geq 1-\varepsilon$ after a time period of $O(d \cdot \log(1/\varepsilon))$. When there is no observable attack, GHAST only needs $3d$ time to achieve confirmation at the same confidence level as six-block-confirmation in Bitcoin, while it takes roughly $360d$ in Bitcoin.

1 Introduction

Blockchain systems like Bitcoin provide secure, decentralized, and consistent ledgers at Internet-scale. Such ledgers are initially designed for cryptocurrencies, but now have evolved to become a powerful abstraction that fuels innovations on many real-world applications across financial systems [3], supply chains [9], and health cares [4].

A novel aspect of blockchain systems is permissionless. It allows anyone to join or leave the system freely without getting approval from some centralized or distributed community. During the blockchain protocol execution, it is not necessary for any participant to be aware of others, once the protocol message from the other participants can be relayed on time. In order to prevent malicious behavior in a permissionless setting, the blockchain systems limit the rate for constructing new blocks using the idea of computational puzzles, which is called *proof-of-work*. To construct a valid block accepted by the blockchain protocol, the participants need to set a proper nonce in block to make the hash value of such block fall in a prescribed bit-string set. (e.g., the set contains bit-string with 70 leading zeros.) So all the participants need to try a large number of different nonces before finding a valid block.

The robustness of a blockchain system requires a majority of computing power is held by honest participants. So in the long term, the honest participants will generate more blocks than the attacker. Based on this fact, the blockchain protocol directs the participants to organize blocks and select a sequence of blocks as history. For example, Bitcoin adopts *Nakamoto consensus* protocol [17] which operates on a tree of blocks and selects the longest branch as its correct history. All the honest participants are required to append new blocks to the longest branch. Ideally, all the blocks generated by the honest participants will extend the longest chain in Nakamoto consensus. If we assume the attacker controls at most β computing power ($\beta < 1/2$) in total, for each block, we can compute the risk that such block is kicked out of history under the optimal attack strategy in the future. If the risk does not exceed a given threshold, we say a block is *confirmed*. Once a block is confirmed, we have high confidence for the transactions (e.g., payment message) carried in such block is recorded on the blockchain irreversible. A protocol that lacks consideration of all possible attack strategies may provide an incorrect way to estimate confirmation risk. And further, a confirmed block may be kicked out of the history frequently. It is regarded as a security flaw. So a rigorous security analysis handling all the possible attack cases is necessary for a blockchain protocol.

The network propagation delay brings issues in reaching consensus. Since a block can not be relayed to other participants instantly, the participants sometimes have an inconsistent view of the current block sets. The influence of network delay depends on the protocol parameters. A blockchain protocol can adjust the block generation interval by adjusting the difficulty of finding a valid block. If the block generation interval is much higher than the time propagating a message in network, with a high probability, no one will generate a new block when the participants have inconsistent local blocks. The blockchain protocol works in a synchronized network. For the opposite situation, there will be a considerable amount of blocks which generated when the participants have an inconsistent view and may cause the participants to diverge. For example, the participants may regard different branches as the longest branch. For a consensus protocol running in a low block generation interval, the protocol design must deal with the inconsistency view carefully.

A high block generation interval results in a bad performance. A higher block generation interval means fewer blocks are generated in a given time interval. So the consensus protocol has a low throughput. A high block generation interval also results in a high block confirmation delay. For example, Nakamoto consensus requires the block propagation delay d (i.e., the delay of one block propagating to all participants in the P2P network) must be significantly smaller than the block generation interval of $1/\lambda$. Otherwise, a large number of blocks will be generated in the scenario that another block is in propagation. These blocks will not contribute to the growth of the longest chain. Once the chain growth of the longest chain slowing down, it requires less cost for the attacker to construct a side chain competing with the confirmed history. Furthermore, the confirmation of a block has to wait for several subsequent blocks, since in a permissionless consensus system the agreement is only observable through mined blocks. Therefore Nakamoto consensus has to operate with very low block generation rate (e.g., 1 MB blocks per 10 minutes in Bitcoin) and suffer from unsatisfactory throughput and confirmation latency (e.g., 6 blocks or equivalently 60 minutes in Bitcoin).

Performance becomes one of the major obstacles that impede the adoption of blockchain techniques. To resolve the performance issues in Nakamoto consensus, several new protocols are proposed in the last five years. Some of them have a rigorous security analysis. Garay et al [7] first provide a rigorous security analysis for Nakamoto consensus in a synchronized network model. They prove two properties of Nakamoto consensus, the common prefix and the chain quality. Pass et al [19] consider the effect of network delay and provide an analysis in an asynchronized network model with a prior maximum network delay d. They prove an additional property, the chain-growth. Several subsequent works [5, 20, 25] built their security analysis on the top of the basic properties in Nakamoto consensus. These works achieve a good performance in throughput. However, since they build security on the top of Nakamoto consensus, they can not achieve a better confirmation latency than Nakamoto consensus.

In the same period, Sompolinsky et al [24] introduce GHOST (Greedy Heaviest-Observed Sub-Tree), which uses another way to select the branch. However, they only analysis the behavior of GHOST under some attack cases. Later, Kiayias et al [10] provide a security analysis for GHOST in a low block generation rate under a synchronized network model.

Unfortunately, Natoli et al [18] point out GHOST is vulnerable in a liveness attack when the block generation rate is high. The liveness attack is not aimed to re-ordering the confirmed blocks but tries to prevent from confirming the new blocks. They provide an attack strategy called the balance attack for a high block generation rate. We will introduce the details of this attack in Section 2.

We notice that a high block generation rate helps reduce confirmation delay. Given a time interval of T, let random variable X denote the ratio of newly generate blocks between malicious blocks and honest blocks. Since the proof-of-work protocols always assume the attacker has less computing power compared to the honest participants, the expectation of X is less than 1. The higher block generation rate, the lower variance X will have. So it is less likely for an attacker to generate more blocks than honest participants in a given time interval. The blocks can gain an advantage in subtree weight compared to the attacker's side chain quickly and achieve a lower confirmation delay. However, the existing protocols built on the top Nakamoto consensus can not break the barrier of confirmation delay in Nakamoto consensus and GHOST suffers a liveness issue in a high block generation rate.

1.1 Our contributions

This paper presents the GHAST (Greedy Heaviest Adaptive Sub-Tree) consensus protocol which achieves a nearly optimal confirmation delay in a normal case with rigorous security analysis. This protocol is designed based on the GHOST protocol with a high block generation rate. In order to resolve the liveness issue in GHOST protocol, GHAST slows down the block generation rate to defense the liveness attack. More precisely, when detecting a divergence of computing power, the block weight distribution is adaptively changed. Only a small fraction of blocks selected randomly are marked as "heavy blocks" and other blocks generated under this circumstance are valid but have zero weight. The block generation rate remains to keep a high throughput. In other words, only the heavy blocks are taken into consideration in the branch selection.

This work is the first design of a high-throughput BlockDAG consensus protocol (among all DAG-based consensus proposals including a bunch of GHOST-like protocols) that has a rigorous security analysis and liveness against an attacker with the ability to manipulate network delay. Furthermore, our protocol is also the first consensus protocol that provides both efficiency and robustness: 1) fast confirmation when there is no observable attack, i.e., the agreed history is immutable against covert attacks; and 2) polynomially bounded worst-case liveness when there is an active attack with 49% block generation power as well as the ability to arbitrarily manipulate the delay of every block within the maximum propagation delay bound d (recall that blocks exceeding this bound are counted as malicious).

Liveness guarantee We prove that GHAST guarantees security and liveness in the presence of an active attacker who has the power of manipulating communication delays of every block to every participant. Similar to the framework in the analysis of Nakamoto consensus, we have the following assumptions.

- The block generation rate of all the participants is λ.
- The adversary controls β computing powers among all the participants. ($\beta < 1/2$) In other words, the block generation rate of the adversary is $\beta\lambda$.
- There is a maximum latency d within which a block will be propagated to all honest nodes.

Once a block is received by all the honest participants, its order in history will be consistent among all the honest participants and become unchangeable after time $O(\log(\varepsilon))$, with probability $1 - \varepsilon$. More precisely, we have the following theorem.

Theorem 1.1 (Informal) *For every risk tolerance $\varepsilon > 0$ and fixed system parameters λ, d, β, let $\delta := 1 - \beta/(1-\beta)$, if $\lambda d \geq 5 + 0.8\log(1/\delta)$, there exist appropriate parameters such that GHAST guarantees that every block broadcast before time t is confirmed with confidence $\geq (1-\varepsilon)$ by time $t + d \cdot O\left(\frac{\log(1/\delta\varepsilon)}{\delta^3}\right)$.*

A formal version is given in theorem 4.13.

Low confirmation delay GHAST achieves fast confirmation time in the absence of an observable attack. Here an unobservable attack includes both cases of attacks that happened in the future and covertly withholding blocks without attempting to influence the current state (but withheld blocks may be released in the future). That is, as long as the attacker is not actively inuencing the Block-TG consensus system, transactions can be confirmed quickly and become immutable once confirmed even if under the fast conrmation rule. We provide a concrete method in estimating the block confirmation risk and runs an experiment. The system parameters in the experiment can tolerate liveness attacks from a powerful attacker that controls 40% of the network computation power. Conflux blockchain system running the GHAST protocol result in [14]shows that GHAST can obtain the same confidence as waiting for six blocks in Bitcoin in $3d$. While Bitcoin requires $360d$ and Prism requires $23d$.

1.2 Main techniques

GHOST protocol and Tree-Graph structure. The GHAST consensus protocol adopts the GHOST protocol proposed in [24] as the backbone of our protocol. We call the branch selected by GHOST protocol *pivot chain*.

Borrowing ideas from previous works [13, 22, 23], GHAST organizes the block in the Tree-Graph structure. Blocks in are Tree-Graph structure linked with two types of directed edges. Each block has one outgoing *parent edge* to indicate its parent block under GHOST protocol. And it may have multiple outgoing *reference edges* to show generation-before relationship between blocks. The parent edge and reference edges of a block are immutable. The reference edges also reflect the local Tree-Graph of the block's miner when generating such block.

Structured GHOST protocol. In structured GHOST protocol, only $1/\eta_{\mathsf{w}}$ of blocks are weighted blocks that would count in the chain selection process, where η_{w} is a protocol parameter. These blocks are selected randomly based on their hash value. During the chain selection process, all the un-weighted blocks are skimmed. It is equivalent to slows down block generation rate η_{w} times. Kiayias et al [10] prove that GHOST protocol has no liveness issue when the block generation rate is low enough. Their proof is based on a synchronized network model, which is different from our model. Our further analysis implies that this claim also holds in a partially synchronized network model.

Consensus with two strategies. The GHAST consensus protocol operates with two strategies, an optimistic strategy following the GHOST protocol and a conservative strategy following the structured GHOST protocol. We adopt an adaptive weight mechanism to incorporate two strategies into one framework. The blocks under the GHOST protocol have block weight one and the weighted blocks under structured GHOST protocol have block weight η_{w}. So the expected block weight does not vary while switching the strategies. In normal scenarios, the GHAST consensus protocol adopts the optimistic strategy. When a serious liveness attack happens, the GHAST consensus protocol switches to the conservative strategy. All the block headers include an immutable strategy bit to indicate the strategy it adopts to make the miners reach a consensus for its block weight.

Enforced strategy choices. We found that if an attacker can fill the strategy bit arbitrarily, the confirmation delay will be much worse than our expectations. So the GHAST consensus protocol determines the strategy bit of each block based on its "past graph". The *past graph* of a block refers to the set of all its reachable blocks following the parent edges and reference edges recursively. When an honest miner generates a new block, its current local Tree-Graph is the same as the past graph of this new block. If the past graph reflects an liveness attack is happening, the block should follow the conservative strategy. Otherwise, it should follow the optimistic strategy.

We define a concrete rule to decide the consensus strategy from the past graph. A block whose strategy bit is inconsistent with its past graph will be regarded as an invalid block and dropped by all the honest participants. So the consensus strategy choices are enforced by the consensus protocol. Since the strategy bit can be inferred from its past set, it can be omitted from the block header.

Notice that an attacker can still manipulate the strategy bit by ignoring some blocks in its past set. But its ability to delay block confirmation can be significantly reduced.

Detecting liveness attack. The GHAST consensus protocol provides a deterministic algorithm to detect if there is an active liveness attack given a past set and decide the consensus strategy. The GHAST consensus protocol detects liveness attack following one idea:

> *Whether there exists an old enough block in the branch selected by GHOST rule,*
> *its best child doesn't have a dominant advantage in subtree weight compared to its sibling blocks.*

Recalling that GHOST protocol selects the pivot chain by picking the child with maximum subtree weight recursively. In the Tree-Graph in an honest participant's view, if a block in the pivot chain doesn't have a child with a dominant advantage in subtree weight, other participants may have a different opinion in picking the next block in pivot chain. If such a block has been generated for a long time, we suspected that a liveness attack is happening.

Partially-synchronized clock Pass et al [19] mentions that the Bitcoin protocol can be used as a partially-synchronized clock. And their subsequent works Fruitchain [20] and Hybrid Consensus [21] show two examples that use Bitcoin protocol as a fundamental service. The GHAST protocol also runs a stand-alone blockchain following Bitcoin protocol, which we called *the timer chain*. Each block in Tree-Graph structure includes the hash value of the longest

branch leaf block in the timer chain. The height of the included timer block represents an imprecise timestamp. Given a local Tree-Graph and a block in this Tree-Graph, if the timestamp difference between the given block and maximum timestamp in the local Tree-Graph exceeds a threshold η_{b}, we regard such block as an old enough block in the Tree-Graph.

Notice that the GHAST consensus protocol only uses the timer chain to decide whether a block is old enough. The order of Tree-Graph blocks does not rely on their timestamp. The blocks confirmation in Tree-Graph does not need to wait for the confirmation of their timer block. In a normal scenario, a block in Tree-Graph is usually confirmed earlier than its timer block.

Embedding timer chain into Tree-Graph. Some consensus protocols have multiple proof-of-work tasks. For example, the GHAST consensus protocol has two proof-of-work tasks: mining a Tree-Graph block and mining a timer block. In a parallel chain protocol like OHIE [25] and Prism [2], each individual chain has a proof-of-work task. Usually, these protocols require that an attacker can not obtain majority computing power in each proof-of-work task. In order to prevent the attacker from concentrating its computing power on one task, a widely used trick makes the participants work on all the proof-of-work tasks simultaneously. It constructs a block that includes the components (or the digests) of all the tasks. When a block is successfully mined, its hash value decides the block type.

Following this trick, the timer block and Tree-Graph block have the uniform block format, each of which includes a parent edge, several reference edges, the hash value of the last timer block and other metadata such as transactions digest in the application. The GHAST consensus protocol also regards the timer block as a valid Tree-Graph block.

1.3 Related work

Nakamoto consensus. Nakamoto consensus [17] is the first blockchain protocol. In Nakamoto Consensus, each block has one predecessor block and all blocks form a tree rooted at the genesis block. Pass et al [19] build a round based analysis framework for Nakamoto consensus in an asynchronized network model with a prior known maximum network delay d. Given the adversary computing power threshold β, network delay d and block generation rate λ, they show several properties of Nakamoto consensus when $1 - \lambda d > \beta/(1 - \beta)$.

Some other blockchain systems like LiteCoin, Bitcoin Cash, Bitcoin Gold and Bitcoin SV tried to increase the throughput by tuning system parameters in Nakamoto consensus. However, Sompolinsky et al [24] give the tradeoff between increasing throughput of Nakamoto consensus and security threshold β. The Nakamoto consensus has two parameters related to throughput: the block generation rate λ and block size s. The throughput of Nakamoto consensus is bounded by $\lambda \cdot s$. In a network with limited bandwidth b, the propagation delay d is lower bounded by s/b. Since the previous analysis requires $1 - \lambda d > (1 + \delta)\beta/(1 - \beta)$, the throughput is upper bounded by $\frac{(2-\beta)\cdot s}{1-\beta}$.

FruitChain. FruitChain [20] organizes blocks hierarchically to decouple these functionalities. It packs transactions first into fruits (i.e., micro blocks) and then packs fruits into blocks (i.e., macro blocks). Both types of blocks are required solutions for proof-of-work puzzle. But only the macro blocks are maintained following Nakamoto consensus.

The mining rewards in FruitChain are mainly distributed via micro blocks. It mitigates some problems like selfish mining [6]. In a selfish mining attack, the attacker manipulates the longest chain by withholding its newly generated block accordingly to increase the ratio of its block in the blockchain. This makes the attacker receive more mining reward. In FruitChain, since the mining rewards are distributed according to micro blocks, it is no need to manipulate the macro blocks and the attacker can not apply this apply strategy over micro blocks because they are not maintained by Nakamoto consensus.

The drawback of FruitChain is that the block confirmation is built on the top of macro blocks, which follows Nakamoto consensus. The micro blocks become irreversible only if the macro block packing it is confirmed. So the confirmation delay of FruitChain is as worse as Nakamoto consensus.

Bitcoin-NG Bitcoin-NG [20] also organizes blocks hierarchically. In Bitcoin-NG, the key blocks (i.e., macro blocks) are organized following Nakamoto consensus. Once a miner generates a macro block, it is allowed to generate a sequence of light blocks (i.e., micro blocks) until the next miner generates a macro block. Each time a miner trying to generate a macro block, it should try to include all the micro blocks generated by the owner of the previous macro

block. Similar with FruitChain, the micro blocks are not taken into consideration in branch selection of macro blocks. Bitcoin-NG has the same drawback as FruitChain since its security is also built on the Nakamoto Consensus.

Hybrid consensus Hybrid consensus [21] extends the idea in Bitcoin-NG. In Bitcoin-NG, a leader is chosen periodically based on the mining of macro blocks. Hybrid consensus picks a small quorum from the miner of macro blocks. Unlike Bitcoin-NG, a miner is not included in the quorum at the time of its generation. Recalling that in Nakamoto consensus, the longest branch truncating the last k blocks has consistent property. So Hybrid consensus picks quorum from the truncated branch. As the blocks in the truncated branch become irreversible, the chosen quorum will not change. This is different with Bitcoin-NG. The quorum runs a PBFT protocol to commit transactions.

In Hybrid consensus, the security threshold β drops to $1/3$ to guarantee the attacker can not control more than 1/3 nodes in a selected quorum, which is the requirement of PBFT protocol. Hybrid also requires honestness has some stickiness, i.e., it takes a short while for an adversary to corrupt a node. So the adversarial can not corrupt the whole quorum instantly once a quorum is selected. However, such an assumption shows that the selected quorum is the single point of failure for the whole consensus protocol. If an attacker continues to DDoS attack the newly selected quorum, Hybrid consensus protocol will crash.

OHIE Another approach in increasing the throughput is running several parallel chains. In OHIE [25], the particpants mines on the hundreds of parallel chains simultaneously. When mining a block in OHIE, the miner needs to include the parent block hash of the current block in each chain. Once a valid proof-of-work puzzle is solved, the block hash determines which chain the new block belongs to. The parent blocks for each chain are selected following Nakamoto consensus. Each individual chain has a low block generation rate to match the security requirement in Nakamoto consensus. The parallel chain remains a low block generation rate for security and all the chains achieve a high throughput collaboratively.

However, such design increases the cost in metadata extremely. In order to achieve desirable performance, OHIE runs 640 parallel chains and generates 64 blocks per second. In security analysis of OHIE, a block will be confirmed in OHIE only if it is confirmed in the individual chain it belongs to. So its confirmation time is worse than the Nakamoto consensus.

Prism Prism [2] also runs parallel chains. Unlike OHIE, the parallel chains in Prism do not carry transactions. So Prism does not need to order the blocks in parallel chains. Prism has three types of blocks: transaction blocks that only pack transactions (like fruit in FruitChain), proposer blocks that pack transaction blocks and voter blocks that run in parallel chains. The voter blocks will vote for the proposer blocks and pick a leader block for each height. Prism orders the leader blocks according to their height. The leader block not necessarily appears in the longest branch. So the block confirmation in Prism does not depend on the confirmation of proposer blockchain.

The most clever point in Prism is that the confirmation of a leader block does not need to wait for its voters become irreversible. Though the delay for one voter block becomes irreversible is as worse as Nakamoto consensus, Prism claims that reverting a majority of voter chains at the same time is much more difficult than reverting one voter chain. So even if a few voter chains are reverted, as long as the leader block receives a majority votes, it is not reverted by the attacker. Prism is the first proof-of-work consensus protocol that breaks confirmation barrier in Nakamoto consensus. Our work has a better performance than Prism.

Prism still has some drawbacks. Similar to OHIE, parallel chains increase the amount of metadata in the protocol significantly. Prism only provides a security analysis in a synchronized network model, which is doubted unrealistic by [19].

GHOST Since the throughput in Nakamoto consensus is upper bounded by security threshold β, Sompolinsky et al [24] introduce GHOST, which uses another way to select the branch. [1]. Instead of measuring the length of branches, GHOST defines the subtree weight to measure the number of blocks in the subtree rooted at each block. For each block, GHOST regards its child block with maximum subtree weight as the best child and breaks ties by block hash. Started with the genesis block, GHOST visits the best child recursively to select the branch. GHOST claims that once all the

[1] Though the structured variants of Nakamoto consensus resolves the throughput issue, GHOST is proposed earlier than their work.

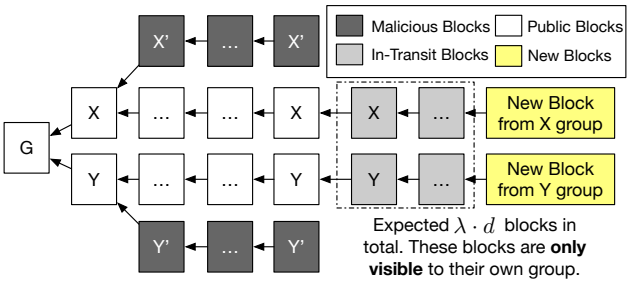

Figure 1: An example for balance attack.

honest nodes mining under the subtree of one block, the growth of its subtree weight will not be undermined on the decreasing of block generation interval. Thus GHOST claims it resolves the security issue of Nakamoto consensus in a high block generation rate (a low block generation interval). However, Sompolinsky et al only analysis the behavior of GHOST under some attack cases, lack of a rigorous security analysis with a practical result.

Kiayias et al [10] provides a security analysis for GHOST in a low block generation rate under a synchronized network model. When the block generation rate in GHOST is low enough to matches the requirement in Nakamoto consensus. GHOST has the same security properties as Nakamoto consensus and can have the same confirmation. Kiffer et al [11] try to provide a similar analysis under an asynchronized network model.

Liveness attack for GHOST Natoli et al [18] first point out GHOST is vulnerable facing a liveness attack when the block generation rate is high. The liveness attack is not aimed to re-ordering the confirmed blocks, but tries to prevent from confirming the new blocks.

They provide a liveness attack strategy called the balance attack. Suppose the total block generation rate is λ and the attacker is able to delay the message communication with time d. The attacker splits the honest miner into two groups with similar mining power. Figure 1 presents one example of such attacks. The example has the following settings: 1) the total block generation rate of honest participants is λ; 2) honest participants are divided into two groups with equal computation power (group X and group Y in Figure 1); 3) blocks will transmit instantly inside each group, but the propagation between these two groups has a delay of d. In Figure 1, each of the two groups extend their own subtree following the GHOST rule. Note that recent generated blocks within the time period of d are in-transit blocks (gray blocks in Figure 1), which are only visible by the group that generates it. Therefore each group will believe its own subtree is larger until one group generates sufficiently more blocks than the other to overcome the margin caused by the in-transit blocks. In normal scenarios, one of the two groups will get lucky to enable the blockchain to converge. However, an attacker can mine under two subtrees simultaneously to delay the convergence. The attacker can strategically withhold or release the mined blocks to maintain the balance of the two subtrees as shown in Figure 1. Previous work has shown that, if the margin caused by in-transit blocks is significant, i.e., $\lambda d > 1$, an attacker with little computation power can stall the consensus progress [25].

DAG-based structures To improve the throughput and the confirmation speed, researchers have explored several alternative structures to organize blocks. Inclusive blockchain [13] extends the Nakamoto consensus and GHOST to DAG and specifies a framework to include off-chain transactions. In PHANTOM [23], participating nodes first find an approximate k-cluster solution for its local block DAG to prune potentially malicious blocks. They then obtain a total order via a topological sort of the remaining blocks. Unfortunately, when the block generation rate is high, inclusive blockchain and PHANTOM are all vulnerable to liveness attacks. Unlike GHAST they cannot achieve both the security and the high performance.

Some protocols attempt to obtain partial orders instead of total orders for payment transactions. SPECTRE [22] produces a non-transitive partial order for all pairs of blocks in the DAG. Avalanche [1] connects raw transactions into a DAG and uses an iterative random sampling algorithm to determine the acceptance of each transaction. Unlike GHAST, it is very difficult to support smart contracts on these protocols without total orders.

Byzantine fault tolerance. ByzCoin [12] and Thunderella [21] propose to achieve consensus by combining the Nakamoto consensus with Byzantine fault tolerance (BFT) protocols. Algorand [8], HoneyBadger [16], and Stellar [15] replace the Nakamoto consensus entirely with BFT protocols. In practice, all these proposals run BFT protocols within a confined group of nodes, since BFT protocols only scale up to dozens of nodes. The confined group is often chosen based on their recent PoW computation power [12, 21], their stakes of the system [8], or external hierarchy of trusts [15, 16].

However, these approaches may create undesirable hierarchies among participants and compromise the decentralization of blockchain systems. Moreover, all of these approaches except Algorand are also vulnerable to DDoS attacks adaptively targeting those leader or committee nodes. Algorand is vulnerable to long range attacks — an attacker could use a set of old private keys that once hold the majority of coins to create an alternative transaction history that is indistinguishable from the real history for new nodes.

2 Model

We adapt round-based partial synchronous network model similar to [19]. A blockchain protocol is defined as a pair of algorithms $(\Pi^{\vec{\eta}}(1^\kappa), \mathcal{C})$ with security parameter κ. $\Pi^{\vec{\eta}}$ is parameterized by a list of protocol parameters $\vec{\eta}(\kappa)$ and we use Π when the context is clear. It maintains the local state $\mathcal{B}$ consists of a set of blocks and prepares new blocks to be resolved proof-of-work puzzle. $\mathcal{C}$ orders the blocks in $\mathcal{B}$ deterministically.

The model is directed by an environment $\mathcal{Z}(1^\kappa)$ with security parameter κ which interacts with an adversary $\mathcal{A}$ and a set of participant nodes $\mathcal{N}$ activated by $\mathcal{Z}$. $\mathcal{N}$ contains two types of nodes: the *honest nodes* which follow the blockchain protocol $(\Pi, \mathcal{C})$ and the *corrupted nodes* which are controlled by adversary $\mathcal{A}$. There is a random function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\kappa$ which can be accessed by participant nodes via four oracles $\mathsf{H}(x) := \mathcal{H}(x)$, $\mathsf{H.ver}(x, y) := [\mathcal{H}(x) = y]$, $\mathcal{H}^{\mathsf{weight}}(x)$ and $\mathcal{H}^{\mathsf{timer}}(x)$. ($\mathcal{H}^{\mathsf{weight}}$ and $\mathcal{H}^{\mathsf{timer}}$ are used to assign each block two random tags.) The output of $\mathcal{H}$ is interpreted as an integer in $[0, 2^\kappa - 1]$. Each node (honest or corrupted) is allowed to query oracle H once each round.

Round-based execution This protocol proceeds in round to model an atomic time steps (e.g. 10^{-12} seconds). For convenience in the security analysis, we re-order the actions of participants and divide them into four phases:

- *Phase 1*: $\mathcal{A}$ *corrupts* and *uncorrupts* arbitrary nodes in $\mathcal{N}$. It means $\mathcal{A}$ can switch the corrupted nodes set adaptively between rounds.
- *Phase 2*: $\mathcal{A}$ delivers blocks to each node. $\mathcal{Z}$ delivers other messages (e.g. the contents to be recorded on blockchain) to each node.
- *Phase 3(a)*: For honest nodes, they maintain the local state with input delivered by $\mathcal{A}$ and $\mathcal{Z}$ in phase 2, organize a block to be solved proof-of-work puzzle following Π and try to solve the puzzle by querying oracle H. If an honest node constructs a valid block, it delivers to $\mathcal{A}$ and incorporates the new block into its local state.
- *Phase 3(b)*: For corrupted nodes, $\mathcal{A}$ gets access to their local state and takes control of access for oracle H, meaning that adversary is allowed to query oracle H with quota the number of corrupted nodes.

Let random variable $\mathsf{View}^{(\Pi,\mathcal{C})}(\mathcal{Z}, \mathcal{A}, \kappa)$ denote the joint view of all the participant nodes and the adversary in all rounds.

Block and graph A *block* $\mathbf{b}$ is a tuple of $(h_{-1}, \vec{h}, \mathsf{m}, s, h)$, where h_{-1} is a hash value (κ-bits string) of a previous block (a pointer to this block), $\vec{h}$ is a list of hash values to some other blocks, m represents the contents and metadata carried by such block and s is a nonce. h is the hash of block $\mathbf{b}$ satisfying $\mathcal{H}(h_{-1}, \vec{h}, \mathsf{m}, s)$. We use $\mathbf{b}.\mathsf{digest}$ to denote the hash h. The blocks corresponding to h_{-1} or the hash values in $\vec{h}$ are called direct dependency blocks of block $\mathbf{b}$. Following the dependency relation recursively, the other reached blocks are called indirect dependency blocks.

In order to limit the generation rate of blocks, $\Pi^{\vec{\eta}}$ only accepts the block whose hash is smaller than $2^\kappa/\eta_{\mathsf{d}}$, where η_{d} is the *puzzle difficulty* parameter in $\vec{\eta}$. In phase 3(a), an honest node following protocol $\Pi^{\vec{\eta}}$ checks the validity of incoming blocks and incorporate the valid blocks into local state $\mathcal{B}$. After that, it prepares a new block in format of $(h_{-1}, \vec{h}, \mathsf{m}, \perp, \perp)$ directed by protocol $\Pi^{\vec{\eta}}$. The proof-of-work puzzle refers finding an appropriate nonce s with

$\mathcal{H}(h_{-1}, \vec{h}, \mathsf{m}, s) < 2^{\kappa}/\eta_{\mathsf{d}}$ to make this block valid. The only way to solve puzzle is querying $\mathcal{H}$ with random s via oracle H. When an appropriate s is found, we say a block is *generated*.

Protocol Π regards a block as valid if its first component h_{-1} is not $\perp$, its hash value is consistent with other components and it solves the proof-of-work puzzle. The *genesis block* $\mathbf{g} := (\perp, \perp, \perp, \perp, \mathcal{H}(\perp, \perp, \perp, \perp))$ is a special valid block which does not satisfy these properties.

A set of valid blocks $\mathbf{B}$ will be regarded as a *valid graph* if for each block $\mathbf{b} \in \mathbf{B}$, its direct and indirect dependency blocks are in $\mathbf{B}$. Since the hash value is unpredictable before querying $\mathcal{H}$, we can simply claim $\mathcal{H}$ must output the hash of block $\mathbf{b}$ later than all its direct and indirect dependency blocks. So there should be no cycle in $\mathbf{B}$. The local state $\mathcal{B}$ is a valid graph $\mathbf{B}$ and we use $\mathcal{B}$ and $\mathbf{B}$ interchangeably. The following part only focuses on the valid blocks and valid graphs. So we omit term "valid" for succinctness.

The blocks constructed by honest nodes are called *honest blocks* and the blocks constructed by adversary are called *malicious blocks*. In our execution model, honest nodes are not informed whether a block is honest or malicious.

Adversary restriction We discuss the model with a restricted environment and adversary.

Definition 2.1 (Admissible environment) *We say that the tuple $(m(\cdot), \beta, d(\cdot), \mathcal{A}, \mathcal{Z})$ is admissible w.r.t. $(\Pi, \mathcal{C})$ if $\beta < 1/2$, $\mathcal{A}$ and $\mathcal{Z}$ are non-uniform probabilistic polynomial-time algorithms, $m(\cdot)$ and $d(\cdot)$ are polynomial functions, and for every $\kappa \in \mathbb{N}$*

- *$\mathcal{Z}$ activates $m = m(\kappa)$ participant nodes;*
- *$\mathcal{A}$ does not modify the contents of delivered message;*
- *$\mathcal{A}$ always corrupts $\beta \cdot m(\kappa)$ corrupted nodes at the same time.* [2]
- *For any block $\mathbf{b}$, if it appears in local state of one honest node in round r, $\mathcal{A}$ is responsible to make sure all the honest nodes incorporate $\mathbf{b}$ to the local states at and after phase 2 of round $r + d(\kappa)$.*

Metrics Here we discuss the aim of our protocol $(\Pi, \mathcal{C})$. Recalling that random variable $\mathsf{View}^{(\Pi,\mathcal{C})}(\mathcal{Z}, \mathcal{A}, \kappa)$ denote the joint view of all the participant nodes in all rounds. The randomness is from the oracle $\mathcal{H}(\cdot)$ and random coins in $\mathcal{Z}, \mathcal{A}$ and participant nodes. Let $\mathcal{U}_r$ collect all the local states of honest nodes in round r (in each phase). Similar with [2], the protocol executes for a finite round r_{max} polynomial in κ.

Our study focus on the finality of block history. The *history of block* $\mathbf{b}$ in local state $\mathcal{B}$ refers the prefix [3] of $\mathcal{C}(\mathcal{B})$ ended at block $\mathbf{b}$, which is denoted by $\mathrm{Prefix}(\mathcal{C}(\mathcal{B}), \mathbf{b})$. If $\mathbf{b} \notin \mathcal{C}(\mathcal{B})$, $\mathrm{Prefix}(\mathcal{C}(\mathcal{B}), \mathbf{b}) = \perp$. Block $\mathbf{b}$ is finalized (or confirmed) if all the honest nodes have consistent history of block $\mathbf{b}$ remain unchanged. Formally, we define $(\varepsilon, \mathcal{A}, \mathcal{Z}, r_0, \kappa)$-finalized as follows.

Definition 2.2 (Finalization) *Let $\mathbf{B}_r$ denote the joint local state of all the honest nodes at round r (after phase 2) in $\mathsf{View}^{(\Pi,\mathcal{C})}(\mathcal{Z}, \mathcal{A}, \kappa)$. Round r_{con} is $(\varepsilon, \mathcal{A}, \mathcal{Z}, r_0, \kappa)$-finalized w.r.t. protocol $(\Pi, \mathcal{C})$ iff*

$$\Pr_{\mathsf{View}^{(\Pi,\mathcal{C})}(\mathcal{Z},\mathcal{A},\kappa)}\left[\forall \mathbf{b} \in \mathbf{B}_{r_{\mathsf{con}}}, \left|\bigcup_{\substack{r \in \{r_0, \cdots, r_{\mathsf{max}}\} \\ \mathcal{B} \in \mathcal{U}_r}} \mathrm{Prefix}(\mathcal{C}(\mathcal{B}), \mathbf{b})\right| = 1\right] \geq 1 - \varepsilon - \mathsf{negl}(\kappa)$$

Since the local state of each honest node is a random variable, the finality is defined over a round r_{con} other than a block $\mathbf{b}$.

Definition 2.3 (Latency) *If there exists r_ε such that for any $r_{\mathsf{con}} \leq r_{\mathsf{max}} - r_\varepsilon$, round r_{con} is $(\varepsilon, \mathcal{A}, \mathcal{Z}, r_{\mathsf{con}} + r_\varepsilon, \kappa)$-finalized, then we say $\mathsf{View}^{(\Pi,\mathcal{C})}(\mathcal{Z}, \mathcal{A}, \kappa)$ has the ε-latency r_ε.*

[2] We assume $\beta \cdot m(\kappa)$ always be an integer here. This setting also handle the case that adversary $\mathcal{A}$ corrupts nodes less than $\lfloor \beta \cdot m(\kappa) \rfloor$, because adversary $\mathcal{A}$ can make some corrupted nodes act like an honest node.

[3] In this paper, a *prefix* of a list could equal to the list itself.

3 Protocol

3.1 Rephrase GHOST protocol in our framework

GHOST proposed in [24] takes a set of blocks in a tree structure as input and outputs a sequence of blocks. Each block in this tree has a non-negative *subtree weight*. For every block $\mathbf{b}$, the subtree weight of $\mathbf{b}$ refers to the total weights of all blocks in the subtree rooted at $\mathbf{b}$. The GHOST starts from the root of the tree and repeatedly proceeds to the child block with maximum subtree weight until reaching a leaf node block. Then the path of blocks will be the chain output by GHOST.

Now, we formalize the GHOST [24] with our notations. Each block under GHOST doesn't have the component of block hash values list. They can be represent by $\mathbf{b} = (h_{-1}, \perp, \mathsf{m}, s, h)$. The genesis block $\mathbf{g}$ is the root of tree in GHOST. For any other valid block $\mathbf{b}$, h_{-1} should be the digest of a valid predecessor block $\mathbf{b}_{-1}$, which is called *parent block* of $\mathbf{b}$. Since two different blocks never have the same digest with negligible exception, we denote the parent block by $\mathbf{b}.\mathsf{parent}$. (For genesis block, $\mathbf{g}.\mathsf{parent} = \perp$.) Started at any block, following the parent block recursively gives a chain of blocks ended as the genesis block. Every two consecutive blocks in this chain have a parent/child relation. It is called *chain of block* $\mathbf{b}$ and defined as

$$\mathrm{Chain}(\mathbf{b}) := \begin{cases} \mathbf{b} & \mathbf{b} = \mathbf{g} \\ \mathrm{Chain}(\mathbf{b}.\mathsf{parent}) \circ \mathbf{b} & \text{otherwise} \end{cases} \tag{1}$$

In a graph $\mathbf{B}$, each block has exactly one outgoing edge except the genesis block with no outgoing edge and there is no cycle because of unpredictable of digest computation. So all the blocks organize in a tree rooted at $\mathbf{g}$. We use $\mathrm{SubT}(\mathbf{B}, \mathbf{b})$ to denote the subtree rooted at block $\mathbf{b}$ in $\mathbf{B}$.

$$\mathrm{SubT}(\mathbf{B}, \mathbf{b}) := \{\mathbf{b}' \in \mathbf{B} : \mathbf{b}' \in \mathrm{Chain}(\mathbf{b})\}. \tag{2}$$

The subtree weight for each block $\mathbf{b} \in \mathbf{B}$ refers the total block weight of blocks $\mathrm{SubT}(\mathbf{B}, \mathbf{b})$. In GHOST protocol, all the blocks have the same weight 1 ($\forall \mathbf{b} \in \mathbf{B}, \mathbf{b}.\mathsf{weight} = 1$) [4] and the subtree weight is formulated as

$$\mathrm{SubTW}(\mathbf{B}, \mathbf{b}) = \sum_{\mathbf{b}' \in \mathrm{SubT}(\mathbf{B}, \mathbf{b})} \mathbf{b}'.\mathsf{weight}. \tag{3}$$

The *children of block* $\mathbf{b}$ refers all the blocks $\mathbf{b}'$ which regard $\mathbf{b}$ as its parent block. In addition, we filters out the blocks with weight 0. This rule is not activated in GHOST protocol since all the block have weight 1. But the following design will introduce zero-weight block.

$$\mathrm{Chldn}(\mathbf{B}, \mathbf{b}) := \{\mathbf{b}' \in \mathbf{B} : \mathbf{b}'.\mathsf{parent} = \mathbf{b} \wedge \mathbf{b}'.\mathsf{weight} > 0\}. \tag{4}$$

Among all the children block, GHOST chooses the one with the largest subtree weight and break tie by choosing the block with minimum block hash. Formally, it can be described by

$$\mathrm{BestChild}(\mathbf{B}, \mathbf{b}) := \underset{\mathbf{b}' \in \mathrm{Chldn}(\mathbf{B}, \mathbf{b})}{\arg\max} \mathrm{SubTW}(\mathbf{B}, \mathbf{b}'). \tag{5}$$

(Note: 1. When $\mathrm{Chldn}(\mathbf{B}, \mathbf{b}) = \emptyset$, let $\mathrm{BestChild}(\mathbf{B}, \mathbf{b}) = \perp$; 2. When there are multiple children having maximum subtree weight, this function breaks tie by returning the block with minimum block digest. [5])

[4] A blockchain system uses puzzle difficulty as block weight. Since our model has a static puzzle difficulty, we simply set block weight be 1.

[5] The original work [24] didn't mention how to break ties when two subtree have the same weight. However, breaking tie with block digest is a common setting in previous work. [19]

Started with the genesis block $\mathbf{g}$, GHOST recursively choose the best child until reaching a block without children. All reached blocks organize a chain called *pivot chain* of graph $\mathbf{B}$. $\mathrm{Pivot}(\mathbf{B})$ is defined formally in figure 1.

Input : A graph $\mathbf{B}$
Output: A sequence of blocks $\mathbf{L}$
Initialize $\mathbf{L}$ with empty list
Initialize $\mathbf{b}$ with genesis block $\mathbf{g}$
$\mathbf{L} \longleftarrow \mathbf{L} \circ \mathbf{b}$
while $\mathrm{Child}(\mathbf{B}, \mathbf{b}) \neq \emptyset$ **do**
  $\mathbf{b} \longleftarrow \mathrm{BestChild}(\mathbf{B}, \mathbf{b})$
  $\mathbf{L} \longleftarrow \mathbf{L} \circ \mathbf{b}$
return $\mathbf{L}$

Figure 2: The definition of $\mathrm{Pivot}(\mathbf{B})$.

Now we describe the protocol $(\Pi^{\vec{\eta}}_{\mathsf{GHOST}}, \mathcal{C}_{\mathsf{GHOST}})$. $\Pi^{\vec{\eta}}$ first initiates the local state of all participant nodes with genesis block $\mathbf{g}$. In phase 3(a) of each round, upon receiving the blocks delivered by adversary $\mathcal{A}$, $\Pi^{\vec{\eta}}_{\mathsf{GHOST}}$ directs the honest nodes check their validity and append them into $\mathbf{B}$ while making sure $\mathbf{B}$ be valid. Then each honest node computes the pivot chain by $\mathrm{Pivot}(\mathbf{B})$ over their local state $\mathbf{B}$, sets h_{-1} be the digest of last block in $\mathrm{Pivot}(\mathbf{B})$, prepares block $\mathbf{b}_{\mathsf{new}} = (h_{-1}, \perp, \mathsf{m}, \perp, \perp)$ and tries to solve proof-of-work puzzle by querying $\mathcal{H}(h_{-1}, \perp, \mathsf{m}, s)$ with random s. After that, the model enters phase 4. In ordering the graph $\mathbf{B}$, order algorithm $\mathcal{C}_{\mathsf{GHOST}}(\mathcal{B})$ simply returns the pivot chain of local state $\mathrm{Pivot}(\mathbf{B})$.

3.2 Tree-graph structure

We adopt the ideas from previous works [22, 23] which allow each block refers multiple predecessor blocks and organize blocks in the structure of directed acyclic graph instead of tree. A valid blocks can be represented by $\mathbf{b} = (h_{-1}, \vec{h}, \mathsf{m}, s, h)$, in which $\vec{h}$ contains a list of digests (pointers) of other valid blocks. The blocks pointed by $\vec{h}$ are called *reference blocks* of $\mathbf{b}$. (Genesis block $\mathbf{g}$ should have empty $\vec{h}$). Started with block $\mathbf{b}$, by following the reference blocks repeatedly, we can reach all the direct and indirect dependency blocks of $\mathbf{b}$. These blocks (not including block $\mathbf{b}$) organize a valid graph, which is called the *past graph* of block $\mathbf{b}$ and denoted by $\mathbf{b}.\mathsf{past}$.

The parent block digest h_{-1} of all blocks organizes blocks in a tree structure and the reference block digests $\vec{h}$ organize blocks in a directed acyclic graph structure. So we call it Tree-graph structure and denote the protocol by $(\Pi^{\vec{\eta}}_{\mathsf{TG}}, \mathcal{C}_{\mathsf{TG}})$. Compared with $\Pi^{\vec{\eta}}_{\mathsf{GHOST}}$, $\Pi^{\vec{\eta}}_{\mathsf{TG}}$ has an additional requirement for block validity. It requires a valid block $\mathbf{b}$ chooses the pivot chain tip of $\mathbf{b}.\mathsf{past}$ as its parent block. So the chain of block $\mathbf{b}$ is consistent with the pivot chain in graph $\mathbf{b}.\mathsf{past}$. (a.k.a. $\mathrm{Pivot}(\mathbf{b}.\mathsf{past}) \circ \mathbf{b} = \mathrm{Chain}(\mathbf{b})$.) In organizing new blocks $\mathbf{b}_{\mathsf{new}} = (h_{-1}, \vec{h}, \mathsf{m}, \perp, \perp)$ to be solved proof-of-work puzzle, $\Pi^{\vec{\eta}}_{\mathsf{TG}}$ prepares h_{-1} and m in the same way as $\Pi^{\vec{\eta}}_{\mathsf{GHOST}}$, and includes the digests of tip blocks in graph $\mathcal{B}$ into $\vec{h}$ to make sure $\mathbf{b}_{\mathsf{new}}.\mathsf{past} = \mathcal{B}$.

The ordering algorithm $\mathcal{C}_{\mathsf{TG}}$ is defined formally in figure 3. It initializes a list $\mathbf{L}$ with only genesis block $\mathbf{g}$ and visits the blocks in the pivot chain sequentially. In each round, when the algorithm $\mathcal{C}_{\mathsf{TG}}$ reaches block $\mathbf{b}_{\mathsf{next}}$ with parent block $\mathbf{b}$, it collects all the blocks in $\mathbf{b}.\mathsf{past}$ but not appended to $\mathbf{L}$ in the last round (not in $\mathbf{b}.\mathsf{past} \cup \{\mathbf{b}\}$), topological sorts them with a deterministic function $\mathrm{TopoSort}(\cdot)$ and appends the result to $\mathbf{L}$. Next the algorithm appends $\mathbf{b}_{\mathsf{next}}$ to $\mathbf{L}$ and continues to visit the next block. The detailed implementation of function $\mathrm{TopoSort}(\cdot)$ is not necessary in subsequent analysis.

3.3 GHAST

Now we introduce the Ghast protocol $(\Pi^{\vec{\eta}}_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})$. It is the same as $(\Pi^{\vec{\eta}}_{\mathsf{TG}}, \mathcal{C}_{\mathsf{TG}})$ except the definition of block weight. In $(\Pi^{\vec{\eta}}_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})$, the protocol may assign different block weight to each block in order to solve the liveness issues in Ghost. [6] The following design requires four more parameters in $\vec{\eta}$: η_{w}, η_{a}, η_{t} and η_{b}.

[6] This design is totally different from the mechanism called *difficulty adjustment* in Bitcoin. Difficulty adjustment mechanism changes the block weights and puzzle difficulty in react to the change of computing power (the frequency in querying oracle $\mathcal{H}$). In the safety analysis model with

Input : A valid graph $\mathbf{B}$ (An alias of local state $\mathcal{B}$)
Output: A sequence of blocks $\mathbf{L}$

$\mathbf{P} \longleftarrow \mathrm{Pivot}(\mathbf{B})$
$\mathbf{b} \longleftarrow$ Pull the first block from $\mathbf{P}$
Assert $\mathbf{b}$ is genesis block $\mathbf{g}$
$\mathbf{L} \longleftarrow \mathbf{b}$
while $\mathbf{P}$ *is not empty list* **do**
  $\mathbf{b}_{\mathsf{next}} \longleftarrow$ Pull the first block from $\mathbf{P}$
  $\mathbf{L} \longleftarrow \mathbf{L} \circ \mathrm{TopoSort}(\mathbf{b}_{\mathsf{next}}.\mathsf{past} \backslash (\{\mathbf{b}\} \cup \mathbf{b}.\mathsf{past}))$
  $\mathbf{L} \longleftarrow \mathbf{L} \circ \mathbf{b}_{\mathsf{next}}$
  $\mathbf{b} \longleftarrow \mathbf{b}_{\mathsf{next}}$
return $\mathbf{L}$

Figure 3: The definition of $\mathcal{C}_{\mathsf{TG}}(\mathcal{B})$.

3.3.1 Assign block weight accordingly

Previous work [25] figures out an adversary with a low computing power in ratio (e.g. $\beta = 0.2$) has the ability to defer the finality of a block when the block generation interval (m/η_{d} rounds in average) is much smaller than the maximum message delay (d rounds). In order to solve this problem, we propose a structured GHOST protocol which slows down the weighted block generation rate. We assign each block $\mathbf{b} = (h_{-1}, \vec{h}, \mathsf{m}, s, h)$ a random tag computed by oracle $\mathcal{H}^{\mathsf{weight}}(h) := \mathcal{H}(\mathsf{weight}, h)$ where weight represents a fixed bit-string to make $\mathcal{H}^{\mathsf{weight}}(h)$ be independent with the outputs from $\mathcal{H}$ in solving proof-of-work puzzle. Let $\eta_{\mathsf{w}} > 0$ be a parameter which specifies the ratio in slowing down generation rate. We set $\mathbf{b}.\mathsf{weight} = \eta_{\mathsf{w}}$ when $\mathcal{H}^{\mathsf{weight}}(h) < 2^{\kappa}/\eta_{\mathsf{w}}$ and $\mathbf{b}.\mathsf{weight} = 0$ otherwise. It means that a valid block will be assigned with weight η_{w} with probability $1/\eta_{\mathsf{w}}$. Thus the generation rate of the blocks with non-zero weight reduces η_{w} times.

When the block generation rate is low enough to solve issues in Ghost, the confirmation latency is as worse as nakamoto consensus. So the protocol switch between an optimistic consensus strategy and a conservative strategy accordingly. In the optimistic strategy, all the blocks have the same weight 1. When the protocol detects a serious attack happens, it switches to the conservative strategy in which only $1/\eta_{\mathsf{w}}$ of blocks is assigned weight η_{w}.

In a partial synchronous network, it is impossible to make all the honest nodes switch the settings simultaneously. So the protocol determines the consensus strategy for each block individually and incorporates the blocks generated in two strategies into one Tree-Graph. We set a deterministic function $\mathrm{Adapt}(\mathbf{B})$ which is responsible for detecting the presence of a liveness in a local state $\mathbf{B}$. It takes the past-set of a block as input and outputs opt or con to indicate the consensus strategy for this block. For the block $\mathbf{b}$ with $\mathrm{Adapt}(\mathbf{b}.\mathsf{past}) = \mathsf{opt}$, $\mathbf{b}.\mathsf{weight} = 1$. If $\mathrm{Adapt}(\mathbf{b}.\mathsf{past}) = \mathsf{con}$, the block weight equals to 0 or η_{w} depending on $\mathcal{H}^{\mathsf{weight}}(h)$.

3.3.2 Detect liveness attack

Function $\mathrm{Adapt}(\cdot)$ is parameterized by a positive integer η_{a} in $\vec{\eta}$. We define a concept called $\eta_{\mathsf{a}}-$*dominant child*. In a given graph $\mathbf{B}$, when block $\mathbf{b}$ has a child block $\mathbf{b}'$ whose subtree weight is at least η_{a} larger than the subtree weight of the other blocks, we say block $\mathbf{b}'$ is $\eta_{\mathsf{a}}-$dominant child of block $\mathbf{b}$ in graph $\mathbf{B}$. Specially, if block $\mathbf{b}$ has only one child block, the subtree weight of its $\eta_{\mathsf{a}}-$dominant child block should be at least η_{a}. Function $\mathrm{Adapt}(\cdot)$ starts with the genesis block $\mathbf{g}$, visits the $\eta_{\mathsf{a}}-$dominant child repeatedly until reaching a block without $\eta_{\mathsf{a}}-$dominant child. Let block $\mathbf{b}_{\mathsf{c}}$ be the last visited block.

For another graph $\mathbf{B}'$, if the total block weight of symmetric difference between graph $\mathbf{B}$ and graph $\mathbf{B}'$ is less than η_{a}, the pivot chain of $\mathbf{B}$ and graph $\mathbf{B}'$ must have the common prefix ended at $\mathbf{b}_{\mathsf{c}}$. Intuitively, for another honest node whose local state is not much different from $\mathbf{B}$, it should also agree that block $\mathbf{b}_{\mathsf{c}}$ is in pivot chain.

After getting the chain ended at $\mathbf{b}_{\mathsf{c}}$, function $\mathrm{Adapt}(\cdot)$ accesses a *block age speculation* function $\mathrm{Old}(\mathbf{B}, \mathbf{b})$. (It is defined in 3.3.3.) It conjectures whether a block $\mathbf{b}$ is old enough (has been generated for a sufficient long time) at the time point that an honest node has local graph $\mathbf{B}$. If $\mathrm{Old}(\mathbf{B}, \mathbf{b}_{\mathsf{c}})$ output False, we have relative high confidence

constant computing power, Bitcoin and some other protocols like Ghost and Prism [2] doesn't adjust the puzzle difficulty and block weight. But Ghast protocol may assign different block weight to each block.

that honest nodes have almost the same pivot chain and there is no liveness issue. So $\mathrm{Adapt}(\mathbf{B})$ outputs opt when $\mathrm{Old}(\mathbf{B}, \mathbf{b}_\mathsf{c}) = \mathsf{False}$ and outputs con otherwise.

The block age speculation function is required the following two properties:

1. If a block is old enough, all the blocks in its past set should also be old enough. Formally, for any block $\mathbf{b}, \mathbf{b}'$ and graph $\mathbf{B}$ with $\mathrm{Old}(\mathbf{B}, \mathbf{b}) = \mathsf{True}$ and $\mathbf{b}' \in \mathbf{b}.\mathsf{past}$, it should be $\mathrm{Old}(\mathbf{B}, \mathbf{b}') = \mathsf{True}$.

2. Once a block becomes old enough, it will always be old enough in the future. Formally, for any graph $\mathbf{B}, \mathbf{B}'$ and block $\mathbf{b}$ with $\mathrm{Old}(\mathbf{B}, \mathbf{b}) = \mathsf{True}$ and $\mathbf{B} \subseteq \mathbf{B}'$, it will be $\mathrm{Old}(\mathbf{B}', \mathbf{b}) = \mathsf{True}$.

Now we give a formal definition for function $\mathrm{Adapt}(\cdot)$. Let $\mathrm{SibSubTW}(\mathbf{B}, \mathbf{b})$ returns the maximum subtree weight of the siblings of $\mathbf{b}$ (the other children of its parent block).

$$\mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}) := \max_{\mathbf{b}' \in \mathrm{Chldn}(\mathbf{B}, \mathbf{b}.\mathsf{parent}) \setminus \{\mathbf{b}\}} \mathrm{SubTW}(\mathbf{B}, \mathbf{b}'). \tag{6}$$

(If $\mathrm{Chldn}(\mathbf{B}, \mathbf{b}.\mathsf{parent}) \setminus \{\mathbf{b}\}$ is empty set, $\mathrm{SibSubTW}(\mathbf{B}, \mathbf{b})$ returns 0.)

Notice that $\mathbf{b}$ is the $\eta_\mathsf{a}-$dominant child iff $\mathrm{SubTW}(\mathbf{B}, \mathbf{b}) - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}) \geq \eta_\mathsf{a}$. The function $\mathrm{Adapt}(\cdot)$ can be expressed in an equivalent form:

Definition 3.1 *Function* $\mathrm{Adapt}(\mathbf{B})$ *outputs* con *if one of the following two conditions satisfied:*

- $\exists \mathbf{b} \in \mathrm{PivotChain}(\mathbf{B}), \mathrm{Old}(\mathbf{B}, \mathbf{b}.\mathsf{parent}) = \mathsf{True} \wedge \mathrm{SubTW}(\mathbf{B}, \mathbf{b}) - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}) < \eta_\mathsf{a}$.
- *For the last block* $\mathbf{b}'$ *in* $\mathrm{PivotChain}(\mathbf{B})$, $\mathrm{Old}(\mathbf{B}, \mathbf{b}') = \mathsf{True}$

For other cases, $\mathrm{Adapt}(\mathbf{B})$ *outputs* opt. *Specially, if* $\mathbf{B} = \emptyset$, $\mathrm{Adapt}(\mathbf{B})$ *outputs* opt. *The function* $\mathrm{Old}(\mathbf{B}, \mathbf{b})$ *is defined in definition 3.2.*

3.3.3 Block age speculation function

Block age speculation function $\mathrm{Old}(\mathbf{B}, \mathbf{b})$ parameterized by $\eta_\mathsf{t}, \eta_\mathsf{b}$ conjectures if a block has been generated for a long enough time when an honest node see graph $\mathbf{B}$. Let graph $\mathbf{B}$ be the local state of an honest node at round r. Informally, we want to achieve two properties with some integers $r_2 < r_1$:

- If block $\mathbf{b}$ is generated by an honest node, $\mathrm{Old}(\mathbf{B}, \mathbf{b})$ outputs True if $\mathbf{b}$ is generated before round $r - r_1$ and outputs False if $\mathbf{b}$ is generated later than $r - r_2$ (with negligible exception).
- If block $\mathbf{b}$ is generated by adversary, $\mathrm{Old}(\mathbf{B}, \mathbf{b})$ outputs True if $\mathbf{b}$ is generated before round $r - r_1$ (with negligible exception).

Note that we allow $\mathrm{Old}(\mathbf{B}, \mathbf{b})$ to falsely think a newly generated malicious block has been withheld for a long time. Because in some attack cases, an honest node can not distinguish between a newly generated block and an old block. For example, the adversary generates block $\mathbf{b}_1$ when the blockchain system just launched, withholds block $\mathbf{b}_1$ for a long time, and then generates another block $\mathbf{b}_2$ with the same parent blocks and direct dependency blocks. Then block $\mathbf{b}_1$ and block $\mathbf{b}_2$ have the same past set and difference block age.

We start with an idea that the protocol runs another blockchain under nakamoto consensus [17] as *timer chain*. The previous work [19] shows that the longest branch in nakamoto consensus excluding last x blocks will be agreed by all the honest nodes (except with exponentially small probability in x), and the growth rate on chain length has a lower bound and an upper bound. So the timer chain excluding last several blocks grows steadily and never be reverted. A block in tree graph structure can indicate its earliest possible generation time by including a block hash in timer chain. Counting the block height difference between the included block and the newest block in the timer chain, we can speculate the age of a block.

The timer chain is constructed by picking a sequence of blocks in tree-graph structure. Formally, we assign each block $\mathbf{b} = (h_{-1}, \vec{h}, \mathsf{m}, s, h)$ another random tag computed by oracle $\mathcal{H}^{\mathsf{timer}}(h) := \mathcal{H}(\mathsf{timer}, h)$. The blocks with $\mathcal{H}^{\mathsf{timer}}(h) < \eta_\mathsf{t} \cdot 2^\kappa$ are called *timer blocks*. Let $\mathrm{Timer}(\mathbf{B})$ denote all the timer blocks in $\mathbf{B}$. For each timer block $\mathbf{b}$, we defines its height in timer chain as

$$\mathrm{TimerHeight}(\mathbf{b}) := \max_{\mathbf{b}' \in \mathrm{Timer}(\mathbf{b}.\mathsf{past})} \mathrm{TimerHeight}(\mathbf{b}') + 1. \tag{7}$$

Specially, if $\mathbf{b}.\mathsf{past}$ has no timer block, let $\mathrm{TimerHeight}(\mathbf{b}) = 1$. For a graph $\mathbf{B}$, $\mathrm{MaxTM}(\mathbf{B})$ returns maximum timer height.

$$\mathrm{MaxTH}(\mathbf{B}) := \max_{\mathbf{b}' \in \mathrm{Timer}(\mathbf{B})} \mathrm{TimerHeight}(\mathbf{b}'). \tag{8}$$

Definition 3.2 (Block age speculation function) *Given graph* $\mathbf{B}$ *and block* $\mathbf{b}$*,* $\mathrm{Old}(\mathbf{B}, \mathbf{b})$ *outputs* True *iff*

$$\mathrm{MaxTH}(\mathbf{B}) - \mathrm{TimerHeight}(\mathbf{b}) \geq \eta_{\mathsf{b}}.$$

Note that function $\mathrm{Old}(\mathbf{B}, \mathbf{b})$ is also well-defined even if $\mathbf{b} \notin \mathbf{B}$,

3.3.4 Summary for the GHAST protocol

Formally, the protocol $(\Pi_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})$ is the same as $(\Pi_{\mathsf{TG}}, \mathcal{C}_{\mathsf{TG}})$ excepts the way in assigning block weights. Given function $\mathrm{Adapt}(\mathbf{B})$ defined in definition 3.1, the block weights in the GHAST protocol is defined as

$$\mathbf{b}.\mathsf{weight} := \begin{cases} 1 & \mathrm{Adapt}(\mathbf{B}) = \mathsf{opt} \\ \eta_{\mathsf{w}} & \mathrm{Adapt}(\mathbf{B}) = \mathsf{con} \wedge \mathcal{H}^{\mathsf{weight}}(\mathbf{b}.\mathsf{digest}) < 2^{\kappa}/\eta_{\mathsf{w}} \\ 0 & \mathrm{Adapt}(\mathbf{B}) = \mathsf{con} \wedge \mathcal{H}^{\mathsf{weight}}(\mathbf{b}.\mathsf{digest}) \geq 2^{\kappa}/\eta_{\mathsf{w}} \end{cases} \tag{9}$$

4 Security

Now we analysis the security of Ghast protocol. Formally, given $\vec{\eta} := (\eta_{\mathsf{d}}, \eta_{\mathsf{w}}, \eta_{\mathsf{a}}, \eta_{\mathsf{t}}, \eta_{\mathsf{b}})$, we analysis the finalization for $(m, \beta, d, \mathcal{A}, \mathcal{Z})$ which is admissible w.r.t. $(\Pi^{\vec{\eta}}_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})$.

4.1 Skeleton of proofs

We let the adversary $\mathcal{A}$ maintains an additional *adversary state* $\mathcal{S}$ for security analysis only. The adversary state is updated only on the following four types of events:

- *Honest block generation (denoted by* $\mathsf{hGenRls}$*).* An honest node constructs a valid block $\mathbf{b}$, incorporates it into local state and sends it to adversary $\mathcal{A}$.
- *Malicious block generation (denoted by* mGen*).* The adversary constructs a valid block $\mathbf{b}$.
- *Malicious block release (denoted by* mRls*).* For a block $\mathbf{b}$ constructed by the adversary, the first time it appears in the local state of an honest node.
- *Block received by all honest nodes (denoted by* Arvl*).* If a block $\mathbf{b}$ is incorporated to the local state of an honest node at round r, an event happens at round $r + d$ which guarantees block $\mathbf{b}$ appears the local state of all the honest nodes after this time.

We use a tuple $e := (r, \mathbf{b}, t)$ with $t \in \{\mathsf{hGenRls}, \mathsf{mGen}, \mathsf{mRls}, \mathsf{Arvl}\}$ to denote a type t event happens at round r corresponding to block $\mathbf{b}$. A function $\psi(\mathcal{S}_{-1}, e) = \mathcal{S}$ directs adversary updates $\mathcal{S}_{-1}$ to $\mathcal{S}$ when event e happens.

We define a *global potential value* $\tilde{P}(\mathcal{S}, \mathbf{B})$ to quantify the adversary's power in changing the history of blocks in $\mathbf{B}$. We show that the history of blocks in $\mathbf{B}$ will be consistent among all the honest participants become unchangeable unless $\tilde{P}(\mathcal{S}, \mathbf{B})$ larger than a threshold in the future in theorem 4.7. In other words, for any graph $\mathbf{B}$, the finalization of blocks in graph $\mathbf{B}$ can be reduced to the upper bound for $\tilde{P}(\mathcal{S}, \mathbf{B})$.

We impose an *event value* on every events to upper bound its influence on global potential value. (Theorem 4.9.) The sum of event values naturally implies an upper bound for the global potential value. So the finalization property can be derived from the random variable for the sum of event values.

We will introduce the intuitions in designing adversary state update function $\psi(\mathcal{S}_{-1}, e)$, potential function $\tilde{P}(\mathcal{S}, \mathbf{B})$, and event value in section 4.2. Then we will define these functions formally in section 4.3. It is inconsistent with the intuition in some details since the intuitions are illustrative. Section 4.4 proves that our design achieves our aim in proof skeleton.

4.2 Intuitions

Common pivot chain Recalling that the ordering algorithm $\mathcal{C}_{\mathsf{GHAST}}(\mathcal{B})$ is the same as $\mathcal{C}_{\mathsf{TG}}(\mathcal{B})$ except the definition of block weight. From the definition of $\mathcal{C}_{\mathsf{TG}}(\mathcal{B})$ in figure 3, the block order is only determined by the pivot chain $\mathrm{Pivot}(\mathcal{B})$. (We use $\mathbf{B}$ and $\mathcal{B}$ interchangeably.) Given two graphs $\mathbf{B}_1, \mathbf{B}_2$, if block $\mathbf{b}$ appears in the pivot chain of graph $\mathbf{B}_1$ and graph $\mathbf{B}_2$, block $\mathbf{b}$ and the blocks in its past-set $\mathbf{b}.\mathsf{past}$ must have the same history in graph $\mathbf{B}_1$ and $\mathbf{B}_2$. Because the execution of $\mathcal{C}_{\mathsf{TG}}(\mathbf{B}_1)$ and $\mathcal{C}_{\mathsf{TG}}(\mathbf{B}_2)$ are the same before block $\mathbf{b}$ appended to $\mathbf{L}$. So the history of a block will remain unchanged if it is kept in the pivot chain.

A pivot chain block $\mathbf{b}$ will be kept in the pivot chain only if its subtree weight is always no less than the its sibling blocks. We notice that when all the honest nodes are trying to construct block in the subtree of block $\mathbf{b}$, the subtree weight of $\mathbf{b}$ will increase faster than all its siblings in expectation, no matter what adversary $\mathcal{A}$ does. To study this case, we define *common pivot chain* which refers the intersection of the local state pivot chains of all honest nodes. As long as a block $\mathbf{b}$ lies on the common pivot chain, all the newly generated honest blocks will fall into the subtree of $\mathbf{b}$ and contribute to the subtree weight of $\mathbf{b}$. So block $\mathbf{b}$ will accumulate subtree weight advantages compared to siblings blocks. Intuitively, if a block $\mathbf{b}$ can stay in a common pivot chain for a sufficient long time, block $\mathbf{b}$ will be finalized.

Special status Some known attack can make the honest nodes disagree on choosing the best child for an old pivot chain block. In section 3.1, we try to handle this attack by switching to a conservative setting. Here, we want all the honest nodes switch to the conservative setting when the last block of common pivot chain is old enough. Formally, let P be the common pivot chain, $\mathbf{b}_{\mathsf{tip}}$ be the last block of the common pivot chain, $\mathcal{B}$ be the local state of an honest node. We want $\mathrm{Adapt}(\mathcal{B}) = \mathsf{con}$ when $\mathrm{Old}(\mathcal{B}, \mathbf{b}_{\mathsf{tip}}) = \mathsf{True}$.

Let $\mathbf{b}_{\mathsf{tipc}}$ be the next block in $\mathrm{Pivot}(\mathcal{B})$. ($\mathbf{b}_{\mathsf{tipc}} = \perp$ if $\mathbf{b}_{\mathsf{tipc}}$ is the last block in $\mathrm{Pivot}(\mathcal{B})$.) By definition 3.1, when $\mathrm{Old}(\mathcal{B}, \mathbf{b}_{\mathsf{tip}}) = \mathsf{True}$, $\mathrm{Adapt}(\mathcal{B}) = \mathsf{con}$ if $\mathrm{SubTW}(\mathcal{B}, \mathbf{b}_{\mathsf{tipc}}) - \mathrm{SibSubTW}(\mathcal{B}, \mathbf{b}_{\mathsf{tipc}}) < \eta_{\mathsf{a}}$ or $\mathbf{b}_{\mathsf{tipc}} = \perp$. For the case $\mathbf{b}_{\mathsf{tipc}} \neq \perp$, since $\mathbf{b}_{\mathsf{tipc}}$ is not in the pivot chain, there exists local state $\mathcal{B}'$ of another honest node satisfying $\mathrm{SubTW}(\mathcal{B}', \mathbf{b}_{\mathsf{tipc}}) - \mathrm{SibSubTW}(\mathcal{B}', \mathbf{b}_{\mathsf{tipc}}) \leq 0$. Let $\mathbf{T}$ include blocks which are in subtree of $\mathbf{b}_{\mathsf{tip}}$ and have been received by only a part of honest nodes. Let w be the total block weight in $\mathbf{T}$. The total block weight of symmetric difference between $\mathrm{SubTW}(\mathcal{B}, \mathbf{b}_{\mathsf{tip}})$ and $\mathrm{SubTW}(\mathcal{B}', \mathbf{b}_{\mathsf{tip}})$ is at most w. If $w < \eta_{\mathsf{a}}$, we have

$$\mathrm{SubTW}(\mathcal{B}, \mathbf{b}_{\mathsf{tipc}}) - \mathrm{SibSubTW}(\mathcal{B}, \mathbf{b}_{\mathsf{tipc}}) \leq w + \mathrm{SubTW}(\mathcal{B}', \mathbf{b}_{\mathsf{tipc}}) - \mathrm{SibSubTW}(\mathcal{B}', \mathbf{b}_{\mathsf{tipc}}) < \eta_{\mathsf{a}}.$$

So we can guarantee $\mathrm{Adapt}(\mathcal{B}) = \mathsf{con}$ if $\mathrm{Old}(\mathcal{B}, \mathbf{b}_{\mathsf{tip}}) = \mathsf{True}$ and $w < \eta_{\mathsf{a}}$. For the case $w \geq \eta_{\mathsf{a}}$, we can't provide such a guarantee. We define such case as *special status*.

Let w_{h} and w_{m} denote the total weight of honest blocks and malicious blocks in $\mathbf{T}$. So $w = w_{\mathsf{h}} + w_{\mathsf{m}}$. w_{h} is upper bounded by the total weight of honest blocks generated in the past d rounds. All honest nodes generate $(1-\beta)\cdot n \cdot \eta_{\mathsf{d}}/2^{\kappa}$ block weight per round in average. So we can find an threshold ξ such that $w_{\mathsf{h}} \geq \xi$ with a low probability. So if the adversary wants to trigger special status, it needs to release blocks in subtree of $\mathbf{b}_{\mathsf{tip}}$ with total weight at least $\eta_{\mathsf{a}} - \xi$ in consecutive d rounds.

Block potential value For each block $\mathbf{b}$ in the common pivot chain, we introduce a *block potential value* $P(\mathcal{S}, \mathbf{b})$ to quantify the ability of adversary in changing or choosing the next common pivot chain block $\mathbf{b}$. Given round number r, if there exists a block $\mathbf{b}_r$ in the common pivot chain and $\mathbf{b}_r.\mathsf{past}$ contains all the blocks released no later than r. If we want to keep $\mathbf{b}_r$ in the common pivot chain, for each block in $\mathrm{Chain}(\mathbf{b}_r.\mathsf{parent})$, the adversary should not be able to change its next common pivot chain block. We set global potential value $\tilde{P}_r(\mathcal{S})$ the maximum block potential values of blocks in $\mathrm{Chain}(\mathbf{b}_r.\mathsf{parent})$.

Event value Recalling that we impose event values to upper bounds event influence on global potential value. The random variable for the sum of event values is relevant to block finalization.

Noticing that solving the proof-of-work puzzle is essentially a Markov process. Given $r_1 \leq r_2$, whether a hGenRls event or mGen event happens in round r_2 is independent with events before round r_1 and adversary strategies. Since block generation time cannot be manipulated, the attackers strategy becomes almost transparent when only looking at block generation events. Therefore, we only impose non-zero event values on block generation events to skim complicate adversary strategies. Usually, hGenRls events have negative values and mGen events have positive values.

The event values also depend on the background when such event happens. (For example, whether such event happens in special status.)

4.3 Concepts

Notations We define some tool functions and notations used in the following discussion. For any set of blocks $\mathbf{T}$, $\mathrm{TotalW}(\mathbf{T}) := \sum_{\mathbf{b}\in\mathbf{T}} \mathbf{b}.\mathsf{weight}$ returns the total weight of blocks in $\mathbf{T}$. For any two blocks $\mathbf{b}_1, \mathbf{b}_2$ in graph $\mathbf{B}$, $\mathbf{b}_1 \in \mathrm{Chain}(\mathbf{b}_2)$ is equivalent to $\mathbf{b}_2 \in \mathrm{SubT}(\mathbf{B}, \mathbf{b}_1)$ according to notations in section 3.1. We denote their relation by $\mathbf{b}_1 \preceq \mathbf{b}_2$. If furthermore there is $\mathbf{b}_1 \neq \mathbf{b}_2$, then we write $\mathbf{b}_1 \prec \mathbf{b}_2$. If $\mathbf{C}$ is a chain of blocks in which the adjacent blocks have parent/child relation, we use $\mathrm{Tip}(\mathbf{C})$ to denote the last block in $\mathbf{C}$, and use $\mathrm{Next}(\mathbf{C}, \mathbf{b})$ to denote the next element (child block) of block $\mathbf{b}$ in $\mathbf{C}$. Specially, if $\mathbf{b} = \mathrm{Tip}(\mathbf{C})$ or $\mathbf{b} \notin \mathbf{C}$, $\mathrm{Next}(\mathbf{C}, \mathbf{b})$ returns $\perp$. We emphasize that the function $\mathrm{SubT}(\mathbf{B}, \mathbf{b})$ and $\mathrm{SubTW}(\mathbf{B}, \mathbf{b})$ are well-defined when $\mathbf{B}$ is only a subset of valid blocks other than a valid graph. For event $e = (r, \mathbf{b}, t)$, we use $e.\mathsf{block}$ to denote block $\mathbf{b}$.

4.3.1 Adversary State

The adversary state $\mathcal{S}$ is a tuple of

$$(\mathbf{B}^{\mathsf{gen}}, \mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{B}^{\Delta}, \mathbf{M}, \mathbf{f}, \mathbf{C}, \mathbf{S}, v).$$

$\mathbf{B}^{\mathsf{gen}}, \mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}^{\mathsf{min}}$ are three graphs, which include all the generated blocks, all the released blocks (the blocks whose mRls or hGenRls event has happened) and all the blocks released d rounds before (the blocks whose Arvl event has happened). $\mathbf{B}^{\Delta} := \mathbf{B}^{\mathsf{max}} \backslash \mathbf{B}^{\mathsf{min}}$ denotes the blocks which may be received by only a part of honest nodes. $\mathbf{M}$ is a set of malicious blocks in $\mathbf{B}^{\mathsf{gen}}$, so we can distinguish the honest block and malicious blocks. $\mathbf{S}$ and v are a set of blocks and a real number relevant to special status. $\mathbf{f}$ is the flag block. It equals to $\perp$ to represents there is no flag block. $\mathbf{C}$ is a variant of common pivot chain. It will be formally defined later.

For convenient, symbols $\mathbf{B}^{\mathsf{gen}}$, $\mathbf{B}^{\mathsf{max}}$, $\mathbf{B}^{\mathsf{min}}$, $\mathbf{B}^{\Delta}$, $\mathbf{M}$, $\mathbf{f}$, $\mathbf{C}$, $\mathbf{S}$, and v denote corresponding components of $\mathcal{S}$. The symbols with subscript denote components of adversary state with the same subscript in the context. For example, $\mathbf{B}^{\mathsf{max}}_{-1}$ denotes the first component of $\mathcal{S}_{-1}$. For event $e = (r, \mathbf{b}, t)$, we use $e.\mathsf{round}$, $e.\mathsf{block}$ and $e.\mathsf{type}$ to denote its three components.

Traverse function $\psi(\mathcal{S}_{-1}, e) = \mathcal{S}$ updates $\mathcal{S}_{-1}$ to $\mathcal{S}$ when an event e happens. ψ is parameterized by two non-negative integers $s_{\mathsf{m}}, s_{\mathsf{h}}$, which is for security analysis only. s_{m} and s_{h} satisfy $2s_{\mathsf{h}} + 2s_{\mathsf{m}} \leq \eta_{\mathsf{w}}$. . The following part introduces how the traverse function maintain each component of the adversary state.

The first five sets $\mathbf{B}^{\mathsf{gen}}$, $\mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}^{\mathsf{min}}$ are initiated with a set with genesis block $\{\mathbf{g}\}$. $\mathbf{M}$ is initiated with an empty set. Upon event e happens, $\mathbf{B}^{\mathsf{gen}}_{-1}$, $\mathbf{B}^{\mathsf{max}}_{-1}$, $\mathbf{B}^{\mathsf{min}}_{-1}$ and $\mathbf{M}_{-1}$ are updated to $\mathbf{B}^{\mathsf{gen}}$, $\mathbf{B}^{\mathsf{max}}$, $\mathbf{B}^{\mathsf{min}}$ and $\mathbf{M}$ according to $e.\mathsf{type}$.

- hGenRls event: add $e.\mathsf{block}$ to $\mathbf{B}^{\mathsf{gen}}_{-1}$ and $\mathbf{B}^{\mathsf{max}}_{-1}$
- mGen event: add $e.\mathsf{block}$ to $\mathbf{B}^{\mathsf{gen}}_{-1}$ and $\mathbf{M}_{-1}$
- mRls event: add $e.\mathsf{block}$ to $\mathbf{B}^{\mathsf{max}}_{-1}$
- Arvl event: add $e.\mathsf{block}$ to $\mathbf{B}^{\mathsf{min}}_{-1}$

Then we set $\mathbf{B}^{\Delta} = \mathbf{B}^{\mathsf{max}} \backslash \mathbf{B}^{\mathsf{min}}$.

$\mathbf{B}^{\mathsf{max}}$ includes all the blocks appears in the local state of an honest node, and the blocks in $\mathbf{B}^{\mathsf{min}}$ must appear in the local state of all the honest nodes in an admissible environment (by definition 2.1). Thus we can claim an honest node local state $\mathcal{B}$ must always be $\mathbf{B}^{\mathsf{min}} \subseteq \mathcal{B} \subseteq \mathbf{B}^{\mathsf{max}}$. Since all the blocks are added to $\mathbf{B}^{\mathsf{gen}}$ when it is generated, there must be $\mathbf{B}^{\mathsf{max}} \subseteq \mathbf{B}^{\mathsf{gen}}$.

Claim 4.1 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, and any honest node local state $\mathcal{B}$ at the same time, there must be $\mathbf{B}^{\mathsf{min}} \subseteq \mathcal{B} \subseteq \mathbf{B}^{\mathsf{max}} \subseteq \mathbf{B}^{\mathsf{gen}}$.*

Speical status The special status function $\mathrm{Spe}(\mathbf{B}^{\Delta}, \mathbf{C})$ outputs True or False to indicate whether $\mathcal{S}$ is in special status. $\mathrm{Spe}(\mathcal{S})$ represents $\mathrm{Spe}(\mathbf{B}^{\Delta}, \mathbf{C})$ in case we don't care about which components are accessed. The special status is defined as follows.

Definition 4.2 (Special Status) *Given* $\mathbf{B}^{\Delta}$ *and chain* $\mathbf{C}$, $\mathrm{Spe}(\mathbf{B}^{\Delta}, \mathbf{C})$ *returns* True *if one of the following three conditions satisfied:*

1. $\mathrm{SubT}\big(\mathbf{B}^{\Delta} \cap \mathbf{M}, \mathrm{Tip}(\mathbf{C})\big) \geq s_{\mathsf{m}}$
2. $|\{\mathbf{b} \in \mathbf{B}^{\Delta} \backslash \mathbf{M} \mid \mathbf{b}.\mathsf{weight} = 1\}| \geq s_{\mathsf{h}}$
3. $|\{\mathbf{b} \in \mathbf{B}^{\Delta} \backslash \mathbf{M} \mid \mathbf{b}.\mathsf{weight} = \eta_{\mathsf{w}}\}| \geq 3$

Flag block The flag block $\mathbf{f}$ is a block in $\mathbf{B}^{\mathsf{max}}$ or equals to $\perp$ representing no flag block. It is initiated with $\perp$. Given adversary state $\mathcal{S}_{-1}$ and event e, traverse function updates $\mathbf{f}_{-1}$ by the following rules:

1. If $e.\mathsf{block}.\mathsf{weight} = \eta_{\mathsf{w}}$, $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$, and $\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{M}_{-1}$ has no block with block weight η_{w}, then $\mathbf{f} = e.\mathsf{block}$.
2. If $e.\mathsf{block}.\mathsf{weight} = \eta_{\mathsf{w}}$, $e.\mathsf{type} = \mathsf{hGenRls}$ and $\mathbf{f}_{-1} \neq \perp$, then $\mathbf{f} = \perp$.
3. If $e.\mathsf{block} = \mathbf{f}_{-1} \neq \perp$ and $e.\mathsf{type} = \mathsf{Arvl}$, then $\mathbf{f} = \perp$.
4. For other cases, let $\mathbf{f} = \mathbf{f}_{-1}$.

The definition shows that a block can only become a flag block when it is generated, and it is no longer a flag block when its Arvl event happens. So a flag block $\mathbf{f}$ must be in $\mathbf{B}^{\Delta}$.

Claim 4.3 *For any adversary state* $\mathcal{S}$ *appearing in the execution of ghast protocol, if* $\mathbf{f} \neq \perp$, $\mathbf{f}$ *must be an honest block in* $\mathbf{B}^{\Delta}$ *with block weight* η_{w}.

Variant of common pivot chain The chain $\mathbf{C}$ is defined on $\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}$, $\mathbf{C}_{-1}$ and $\mathbf{f}$. [7] First, we define function $\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b})$ which provides a lower bound for subtree weight difference between block $\mathbf{b}$ and its maximum sibling blocks in an honest node local state, with the assumption that the flag block (if exists) has been received by all the honest nodes. $\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b})$ can be represented by $\mathrm{Adv}(\mathcal{S}, \mathbf{b})$ for simplicity.

$$\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}) := \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}, \mathbf{b}\big) - \mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}).$$

Here we regard $\{\perp\}$ as an empty set because $\perp$ presents "no such a block".

Lemma 4.4 *For any two graphs* $\mathbf{B}^{\mathsf{min}} \subseteq \mathbf{B}^{\mathsf{max}}$, *let* $\mathbf{f} \in \mathbf{B}^{\Delta}$ *or* $\mathbf{f} = \perp$. *For any block* $\mathbf{b}$, *there exists at most one block* $\mathbf{b}' \in \mathrm{Chldn}(\mathbf{B}^{\mathsf{max}}, \mathbf{b})$ *with* $\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}') > 0$.

The previous lemma shows that every blocks will have at most one child block $\mathbf{b}'$ with $\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}') > 0$. It is proved in appendix A.2. Based on this property, we specifies the rule in updating $\mathbf{C}_{-1}$. $\mathbf{C}$ is initiated with a genesis block $\mathbf{g}$. Given $\mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}^{\mathsf{min}}$, $\mathbf{C}_{-1}$ is updated in the following steps:

1. Started with the genesis block, we recursively visit its child block $\mathbf{b}$ in graph $\mathbf{B}^{\mathsf{max}}$ which satisfies $\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}) > 0$, until we reach a block without such child block. These blocks organize the chain $\mathbf{C}$.
2. If there is a block $\mathbf{b}' \in \mathbf{C}$ satisfying $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{b}'$ and $\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}') \leq s_{\mathsf{m}} + s_{\mathsf{h}}$, we cut off the suffix started with $\mathbf{b}'$ from $\mathbf{C}$.

So we have the following claims for chain $\mathbf{C}$.

Claim 4.5 *For any adversary state* $\mathcal{S}_{-1}$ *appearing in the blockchain protocol execution, and let* $\mathcal{S} = \psi(\mathcal{S}_{-1}, e)$ *for some event* e. *Let block* $\mathbf{b}$ *be the last block of* $\mathbf{C}$ *and block* $\mathbf{b}_{-1}$ *be the last block of* $\mathbf{C}_{-1}$. *We have*

1. *For any block* $\mathbf{b}' \in \mathbf{C}$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}') > 0$. *Specially, if* $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{b}'$, *then* $\mathrm{Adv}(\mathcal{S}, \mathbf{b}') > s_{\mathsf{m}} + s_{\mathsf{h}}$.
2. *For any block* $\mathbf{b}' \in \mathrm{Chldn}(\mathbf{B}^{\mathsf{max}}, \mathrm{Tip}(\mathbf{C}))$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}') \leq s_{\mathsf{m}} + s_{\mathsf{h}}$. *Specially, if* $\mathbf{b}' \preceq \mathrm{Tip}(\mathbf{C}_{-1})$, *then* $\mathrm{Adv}(\mathcal{S}, \mathbf{b}') \leq 0$.
3. *For any block* $\mathbf{b}_{\mathsf{c}}$, *if all the block* $\mathbf{b}'$ *in* $\mathrm{Chain}(\mathbf{b}_{\mathsf{c}})$ *satisfying* $\mathrm{Adv}(\mathcal{S}, \mathbf{b}') > s_{\mathsf{m}} + s_{\mathsf{h}}$, *then* $\mathbf{b}_{\mathsf{c}} \in \mathbf{C}$.

[7] Notice that $\mathbf{f}$ depends on $\mathrm{Spe}(\mathcal{S}_{-1})$ other than $\mathrm{Spe}(\mathcal{S})$. So we don't have a recursive dependency here.

The set and value related to special status The set of blocks $\mathbf{S}$ and value v are initialized with empty set and 0. $\mathbf{S}$ records all the malicious blocks have been taken into consider in special status. It is defined on $\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{C}, \mathbf{M}$ and $\mathbf{S}_{-1}$. Recalling that the special status consider the blocks in $\mathbf{T} := \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C})\big)$. Upon the events happens, we set $\mathbf{S} = \mathbf{S}_{-1} \cup (\mathbf{T} \cap \mathbf{M})$. The value v traces the total block weight in $\mathbf{S}$ and increases by up to s_{m} in each update. Formally, v is initiated by 0 and let $v := v_{-1} + \min\{s_{\mathsf{m}}, \mathrm{TotalW}(\mathbf{S} \backslash \mathbf{S}_{-1})\}$. We summarize several properties that are immediately induced from the definition.

Claim 4.6 *For any adversary state $\mathcal{S}_{-1}$ appearing in the blockchain protocol execution, and let $\mathcal{S} = \psi(\mathcal{S}_{-1}, e)$ for some event e. Let $\mathbf{T} := \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C})\big)$, we have the following claims*

1. $\mathbf{S}_{-1} \subseteq \mathbf{S}$
2. $0 \leq v - v_{-1} \leq s_{\mathsf{m}}$
3. $v - v_{-1} \leq \mathrm{TotalW}(\mathbf{S}) - \mathrm{TotalW}(\mathbf{S}_{-1})$
4. $\mathbf{S} \backslash \mathbf{S}_{-1} \subseteq \mathbf{T} \cap \mathbf{M} \subseteq \mathbf{S}$.

4.3.2 Potential value

Block potential value Now we provide a formal definition for *block potential value* over the adversary state defined above. The block potential value $P(\mathcal{S}, \mathbf{b})$ is defined for a single block $\mathbf{b}$ over some adversary state $\mathcal{S}$. Formally, the potential value $P(\mathcal{S}, \mathbf{b})$ is $\perp$ by default and it is an integer when $\mathbf{b}$ is old enough in $\mathbf{B}^{\mathsf{min}}$ and agreed by all honest nodes, i.e. $\mathrm{Old}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}) = \mathsf{True}$ and $\mathbf{b} \in \mathbf{C}$. When not being $\perp$, the potential value $P(\mathcal{S}, \mathbf{b})$ is the summation of three components $P_{\mathsf{with}}, P_{\mathsf{adv}}, P_{\mathsf{sp}}$ defined as follows.

- $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b})$ is the total weight of blocks withheld by adversary under the subtree of $\mathbf{b}$:

$$P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) := \mathrm{SubTW}(\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}}, \mathbf{b})$$

- $P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c})$ roughly corresponds to the volatility of the pivot child of $\mathbf{b}$. Let $\mathbf{c} := \mathrm{Next}(\mathbf{C}, \mathbf{b})$, $\mathbf{N} := \{\mathbf{b}' \in \mathbf{B}^{\mathsf{max}} \backslash \mathbf{M} \mid \mathbf{b}'.\mathsf{weight} = 1\}$.

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) := \begin{cases} s_{\mathsf{h}} + s_{\mathsf{m}} - \mathrm{Adv}(\mathcal{S}, \mathbf{c}) - \min\{\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{N}), s_{\mathsf{h}}\} & \mathbf{c} \neq \perp \\ 0 & \mathbf{c} = \perp \end{cases}$$

- $P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) + v$ measures the total cost for triggering the special status. Let $\mathbf{c} = \mathrm{Next}(\mathbf{C}, \mathbf{b})$.

$$P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) := \begin{cases} \mathrm{TotalW}((\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M}) \backslash \mathbf{S}) & \mathbf{c} \neq \perp \\ 0 & \mathbf{c} = \perp \end{cases}$$

And the potential value for block $\mathbf{b}$ is defined as follows

$$P(\mathcal{S}, \mathbf{b}) := \begin{cases} P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) + P_{\mathsf{adv}}(\mathcal{S}, \mathrm{Next}(\mathbf{C}, \mathbf{b})) + P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) & \mathbf{b} \in \mathbf{C} \wedge \mathrm{Old}(\mathbf{B}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True} \\ \perp & \text{Otherwise} \end{cases}$$

Global potential value The global potential function $\tilde{P}(\mathcal{S}, \mathbf{B})$ is defined on the adversary state $\mathcal{S}$ and a graph $\mathbf{B}$. Given adversary state $\mathcal{S}$ and graph $\mathbf{B}$, let $\mathbf{C}' := \{\mathbf{b} \in \mathbf{C} | \mathbf{B} \nsubseteq \mathbf{b}.\mathsf{past}\}$. It returns the maximum block potential values in $\mathbf{C}'$.

$$\tilde{P}(\mathcal{S}, \mathbf{B}) := \max_{\mathbf{b} \in \mathbf{C} \wedge \mathbf{B} \nsubseteq \mathbf{b}.\mathsf{past}} P(\mathcal{S}, \mathbf{b}). \tag{10}$$

Specially, if $\mathbf{C}'$ is empty set or all the blocks in $\mathbf{C}'$ have a block potential value $\perp$, $\tilde{P}(\mathcal{S}, \mathbf{B})$ returns 0.

4.3.3 Event value

In order to upper bound the global potential value $\tilde{P}(\mathcal{S}, \mathbf{B})$, we investigate how an event influences the potential and prove its upper bound by a case-by-case analysis. More specifically, given an event e happens when the adversary state is $\mathcal{S}_{-1}$, we introduce the *event value* of e with respect to the adversary state $\mathcal{S}_{-1}$. Formally, the event value of e is denoted by $\Delta(\mathcal{S}_{-1}, e)$ and defined as follows:

- If $e.\mathsf{type} \in \{\mathsf{mRls}, \mathsf{Arvl}\}$, $\Delta(\mathcal{S}_{-1}, e) := 0$.
- If $e.\mathsf{type} = \mathsf{mGen}$, $\Delta(\mathcal{S}_{-1}, e) := e.\mathsf{block.weight}$.
- If $e.\mathsf{type} = \mathsf{hGenRls}$ and $e.\mathsf{block.weight} = 0$, $\Delta(\mathcal{S}_{-1}, e) := 0$
- If $e.\mathsf{type} = \mathsf{hGenRls}$ and $e.\mathsf{block.weight} = 1$, then

$$\Delta(\mathcal{S}_{-1}, e) := \begin{cases} -1 & \mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False} \\ 0 & \mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{True} \end{cases}$$

- If $e.\mathsf{type} = \mathsf{hGenRls}$ and $e.\mathsf{block.weight} = \eta_{\mathsf{w}}$, then $\Delta(\mathcal{S}_{-1}, e)$ is defined as follows
  - If $\mathbf{B}^{\Delta}_{-1}$ doesn't have an honest block with block weight η_{w} and $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$, $e.\mathsf{block.weight} := 2s_{\mathsf{h}} + 2s_{\mathsf{m}} - \eta_{\mathsf{w}}$. (If $\mathbf{f} = e.\mathsf{block}$, it must be in this case.)
  - If $\mathbf{B}^{\Delta}_{-1}$ doesn't have an honest block with block weight η_{w} and $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{True}$, $e.\mathsf{block.weight} := 0$.
  - If $\mathbf{B}^{\Delta}_{-1}$ has an honest block with block weight η_{w} and $\mathbf{f}_{-1} = \perp$, $e.\mathsf{block.weight} := 0$.
  - If $\mathbf{B}^{\Delta}_{-1}$ has an honest block with block weight η_{w} and $\mathbf{f}_{-1} \neq \perp$, $e.\mathsf{block.weight} := \eta_{\mathsf{w}} + s_{\mathsf{m}}$. (If $\mathbf{f}_{-1} \neq \perp$ and $\mathbf{f} = \perp$, it must be in this case.)

  (Note that a flag block must be an honest block with block weight η_{w} and it must belong to $\mathbf{B}^{\Delta}_{-1}$. So $\mathbf{B}^{\Delta}_{-1}$ doesn't have an honest block with block weight η_{w} only if $\mathbf{f}_{-1} = \perp$.)

4.4 Properties

Notations Let e_n denote the n^{th} event since the ghast protocol launched and $\mathcal{S}_{n-1}$ denote the adversary state when event e_n happens. The traverse function $\psi(\mathcal{S}_{n-1}, e_n) = \mathcal{S}_n$ updates $\mathcal{S}_{n-1}$ to $\mathcal{S}_n$. Variable $N(r)$ denotes the number of events happens before round r. In other words, the last event before round r is $e_{N(r)}$.

Let $\delta := 1 - \beta/(1-\beta)$ and $\lambda := m(d+1)/\eta_{\mathsf{d}}$. So $1-\delta$ represents the ratio of computing power between the honest participants and the adversary, λ approximately represents the expectation of block generated in a maximum network delay.

All the notations in the previous sub-sections are inherited here.

The proof of theorems are in the appendix.

First, we shows the sufficient conditions for block finalization. Theorem 4.7 shows that the history of blocks in $\mathbf{B}^{\mathsf{min}}_{N(r_0)}$ will be consistent among all the honest participants and remain unchanged as long as $\tilde{P}(\mathcal{S}, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) < -\eta_{\mathsf{w}}$ holds and all the blocks $\mathbf{b}$ satisfying $\mathbf{B}^{\mathsf{min}}_{N(r_0)} \nsubseteq \mathbf{b}.\mathsf{past}$ must be old enough. This implies that the analysis for block finalization can be reduced to the analysis of potential value and "not old enough" blocks.

Theorem 4.7 *In execution of ghast protocol, for any r_0 and r_1, we have*

$$\forall \tilde{\mathbf{b}} \in \mathbf{B}^{\mathsf{min}}_{N(r_0)}, \left| \bigcup_{\substack{r \in \{r_1, r_1+1, \cdots, r_{\mathsf{max}}\} \\ \mathcal{B} \in \mathcal{U}_r}} \mathrm{Prefix}(\mathcal{C}_{\mathsf{GHAST}}(\mathcal{B}), \tilde{\mathbf{b}}) \right| = 1$$

as long as both of the following conditions satisfied for any $n > N(r_1)$,

- $\tilde{P}(\mathcal{S}_n, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) < -\eta_{\mathsf{w}}$

- *For any block* $\mathbf{b} \in \mathbf{B}_n^{\mathsf{gen}}$ *with* $\mathbf{B}_{N(r_0)}^{\mathsf{min}} \nsubseteq \mathbf{b}.\mathsf{past}$*, it will be* $\mathrm{Old}(\mathbf{B}_n^{\mathsf{min}}, \mathbf{b}) = \mathsf{True}$.

(Note that $\mathcal{U}_r$ *collects all the local states of honest nodes in round* r*.)*

We study the difference of the addition of potential value and special value between two adjacent adversary state. For the block potential value, we have the following theorem. This theorem shows that when the adversary state is updated from $\mathcal{S}_{-1}$ to $\mathcal{S}$, we provide an upper bound for block potential values except one special case: $P(\mathcal{S}_{-1}, \mathbf{b}) = \perp$, $P(\mathcal{S}, \mathbf{b}) \neq \perp$ and $\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}) = \mathsf{False}$.

Theorem 4.8 *For any adversary state* $\mathcal{S}$ *appearing in the execution of ghast protocol, let* $\mathcal{S}_{-1}$ *be the last adversary state and* e *be the event updates* $\mathcal{S}_{-1}$ *to* $\mathcal{S}$*.*

For any block $\mathbf{b}$ *with* $P(\mathcal{S}_{-1}, \mathbf{b}) \neq \perp$ *and* $P(\mathcal{S}, \mathbf{b}) \neq \perp$*, we have*

$$(P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e).$$

For any block $\mathbf{b}$ *with* $P(\mathcal{S}_{-1}, \mathbf{b}) = \perp$*,* $P(\mathcal{S}, \mathbf{b}) \neq \perp$ *and* $\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True}$*, we have*

$$(P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e).$$

Since the global potential value takes the maximum block potential over a block set, theorem 4.8 derives a similar property for the global potential value in theorem 4.9.

Theorem 4.9 *For any adversary state* $\mathcal{S}$ *appearing in the execution of ghast protocol, let* $\mathcal{S}_{-1}$ *be the last adversary state and* e *be the event updates* $\mathcal{S}_{-1}$ *to* $\mathcal{S}$*. For any graph* $\mathbf{B}$ *we have the following inequality*

$$(\tilde{P}(\mathcal{S}, \mathbf{B}) + v) - (\tilde{P}(\mathcal{S}_{-1}, \mathbf{B}) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e)$$

as long as $\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{g}) = \mathsf{True}$ *for genesis block* $\mathbf{g}$ *and the following holds for every block* $\mathbf{b} \in \{\mathbf{b}' \in \mathbf{C} | \mathbf{B} \nsubseteq \mathbf{b}'.\mathsf{past}\}$*,*

$$\mathrm{Old}(\mathbf{B}^{\mathsf{min}}, \mathbf{b}) = \mathsf{False} \vee \mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True} \vee \tilde{P}(\mathcal{S}, \mathbf{B}) \neq P(\mathcal{S}, \mathbf{b})$$

This theorem shows that when an event e happens, the value of $\tilde{P}(\mathcal{S}_{-1}, \mathbf{B}) + v_{-1}$ grows at most the event value $\Delta(\mathcal{S}_{-1}, e)$. The exceptional case is that a block $\mathbf{b}$ just becomes old enough at $\mathcal{S}$ and at the same time $\tilde{P}(\mathcal{S}, \mathbf{b})$ determines $\tilde{P}(\mathcal{S}, \mathbf{B})$. So for any event index n, let n' denote the last event triggering the exceptional case. The global potential value $\tilde{P}(\mathcal{S}_n, \mathbf{B})$ is upper bounded by the summation of two terms: 1) the global potential value after event $e_{n'}$, a.k.a. $\tilde{P}(\mathcal{S}_{n'}, \mathbf{B})$; 2) The summation of event values (removing the influence from special value), a.k.a. $\sum_{i=n'+1}^{n} (\Delta(\mathcal{S}_{i-1}, e_i) - (v_i - v_{i-1}))$. For the first term, theorem 4.10 provides an upper bound. For the second term, theorem 4.11 show that the summation of tends to negative infinity in a long enough time.

So if we want to prove $\tilde{P}(\mathcal{S}_n, \mathbf{B}_{N(r_0)}^{\mathsf{min}}) < -\eta_{\mathsf{w}}$ for all the sufficient large n, (the first sufficient condition for block finalization given in theorem 4.7), we only need to show that the exceptional case of theorem 4.9 never happens after a time point. Recalling that $\tilde{P}(\mathcal{S}_n, \mathbf{B}_{N(r_0)}^{\mathsf{min}})$ takes the maximum block potential weight among blocks with $\mathbf{b}.\mathsf{past} \nsubseteq \mathbf{B}_{N(r_0)}^{\mathsf{min}}$. Since theorem 4.12 proves that for all the blocks satisfying $\mathbf{b}.\mathsf{past} \nsubseteq \mathbf{B}_{N(r_0)}^{\mathsf{min}}$ must be old enough after a time point, this guarantee that the exceptional case will never happen for $\mathbf{B}_{N(r_0)}^{\mathsf{min}}$ after that. This is also the second sufficient condition for block finalization.

Theorem 4.10 *Given* $(m, \beta, d, \mathcal{A}, \mathcal{Z})$ *which is admissible w.r.t.* $(\Pi_{\mathsf{GHAST}}^{\vec{\eta}}, \mathcal{C}_{\mathsf{GHAST}})$*. Let*

$$w(\varepsilon) := 4\lambda \cdot \max\left\{ \frac{140}{\delta^2} \cdot \log\left(\frac{9000}{\varepsilon\delta^2}\right), \frac{8(\eta_{\mathsf{b}} + 4)}{\delta} \right\}.$$

Let $\tilde{\mathbf{B}}_r$ *denote all the blocks with* $\mathrm{Old}(\mathbf{B}_{N(r)}^{\mathsf{min}}, \mathbf{b}) = \mathsf{False}$*. When* $\lambda \geq 0.8\log(500/\delta)$*,* $\eta_{\mathsf{t}} = 2\lambda/\delta$ *and* $\eta_{\mathsf{w}} = 30\lambda/\delta$*, for any* $r_2 \geq 0$ *and* $\varepsilon > 0$*, we have*

$$\Pr\left[\exists N(r_2) < n \leq N(r_2 + 1), \exists \mathbf{b} \in \tilde{\mathbf{B}}_{r_2}, P(\mathcal{S}_n, \mathbf{b}) \geq w(\varepsilon)\right] \leq \varepsilon.$$

Theorem 4.11 *Given $(m, \beta, d, \mathcal{A}, \mathcal{Z})$ which is admissible w.r.t. $(\Pi^{\vec{\eta}}_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})$. When $\lambda \geq 0.8\log(500/\delta)$ and $\eta_{\mathsf{w}} = 30\lambda/\delta$, for any round $r_1 < r_2$, $\rho > 0$ and $\varepsilon > 0$, if*

$$\frac{r_2 - r_1}{d+1} \geq \max\left\{(3+\rho/\eta_{\mathsf{w}}) \cdot \frac{600}{\delta^2}, \log\left(\frac{4}{\varepsilon}\right) \cdot \frac{3000}{\delta^3}, \log\left(\frac{404}{\varepsilon}\right) \cdot \frac{200}{\delta}\right\},$$

we have

$$\Pr\left[\forall r_2, \sum_{i=N(r_1)+1}^{N(r_2)} \Delta(\mathcal{S}_{i-1}, e_i) - \left(v_{N(r_2)} - v_{N(r_1)}\right) \geq -\rho \middle| \mathsf{View}_{r_1}\right] \leq \varepsilon.$$

(Let View_r denote the joint view in $\mathsf{View}^{(\Pi_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})}(\mathcal{Z}, \mathcal{A}, \kappa)$ before round r.)

Theorem 4.12 *Given $(m, \beta, d, \mathcal{A}, \mathcal{Z})$ which is admissible w.r.t. $(\Pi^{\vec{\eta}}_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})$. When $\beta \geq 0.1$, $\eta_{\mathsf{t}} \geq 2\lambda/\delta$ and r_Δ satisfying*

$$r_\Delta \geq \frac{\eta_{\mathsf{t}}\eta_{\mathsf{d}}}{m} \cdot \max\left\{\frac{128}{\delta^2} \cdot \log\left(\frac{8400}{\varepsilon\delta^2}\right), \frac{8(\eta_{\mathsf{b}}+2)}{\delta}\right\}$$

for any r, we have

$$\Pr\left[\exists n \geq N(r + r_\Delta), \exists \mathbf{b} \in \mathbf{B}^{\mathsf{gen}}_n, \mathbf{B}^{\mathsf{min}}_{N(r)} \nsubseteq \mathbf{b}.\mathsf{past} \wedge \mathrm{Old}(\mathbf{B}^{\mathsf{min}}_n, \mathbf{b}) = \mathsf{False}\right] \leq \varepsilon.$$

Summarize all the previous proofs, finally we gives the finalization properties of the GHAST protocol under any possible attack strategies.

Theorem 4.13 *Given $(m, \beta, d, \mathcal{A}, \mathcal{Z})$ which is admissible w.r.t. $(\Pi^{\vec{\eta}}_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})$. When $\beta \geq 0.1$, $\lambda \geq 0.8\log(500/\delta)$, $\eta_{\mathsf{t}} = 2\lambda/\delta$ and $\eta_{\mathsf{w}} = 30\lambda/\delta$, $\mathsf{View}^{(\Pi_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})}(\mathcal{Z}, \mathcal{A}, \kappa)$ has the ε latency*

$$d \cdot O\left(\max\left\{\frac{\log\left(\frac{1}{\varepsilon\delta}\right)}{\delta^3} + \frac{\eta_{\mathsf{b}}}{\delta^2}\right\}\right) \textit{ rounds.}$$

Proof. Given round r_0, let n denote the largest event index with

$$(\tilde{P}(\mathcal{S}_n, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) + v_n) - (\tilde{P}(\mathcal{S}_{n-1}, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) + v_{n-1}) > \Delta(\mathcal{S}_{n-1}, e_n).$$

We define

$$w_1(\varepsilon) := 4\lambda \cdot \max\left\{\frac{140}{\delta^2} \cdot \log\left(\frac{9000}{\varepsilon\delta^2}\right), \frac{8(\eta_{\mathsf{b}}+4)}{\delta}\right\}.$$

$$w_2(\varepsilon) := \max\left\{(4 + w_1(\varepsilon)/\eta_{\mathsf{w}}) \cdot \frac{600}{\delta^2}, \log\left(\frac{4}{\varepsilon}\right) \cdot \frac{3000}{\delta^3}, \log\left(\frac{404}{\varepsilon}\right) \cdot \frac{200}{\delta}\right\}.$$

$$w_3(\varepsilon) := \frac{\eta_{\mathsf{t}}\eta_{\mathsf{d}}}{m} \cdot \max\left\{\frac{128}{\delta^2} \cdot \log(\frac{8400}{\varepsilon\delta^2}), \frac{8(\eta_{\mathsf{b}}+2)}{\delta}\right\}.$$

According to theorem 4.9, since $(\tilde{P}(\mathcal{S}_n, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) + v_n) - (\tilde{P}(\mathcal{S}_{n-1}, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) + v_{n-1}) > \Delta(\mathcal{S}_{n-1}, e_n)$, there must exists block $\mathbf{b}$ with $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_{n-1}, \mathbf{b}) = \mathsf{False}$, $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_n, \mathbf{b}) = \mathsf{True}$ and $\tilde{P}(\mathcal{S}_n, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) = P(\mathcal{S}_n, \mathbf{b})$. According to theorem 4.10, since $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_{n-1}, \mathbf{b}) = \mathsf{False}$, with probability $1 - \varepsilon$, $P(\mathcal{S}_n, \mathbf{b}) \leq w(\varepsilon)$.

According to the choice of n, for any $n' > n$, we claim $\tilde{P}(\mathcal{S}_{n'}, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) \leq w(\varepsilon) + \sum_{i=n+1}^{n'} \Delta(\mathcal{S}_{i-1}, e_i) - (v_{n'} - v_n)$. So $\tilde{P}(\mathcal{S}_{n'}, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) < -\eta_{\mathsf{w}}$ if

$$\sum_{i=n+1}^{n'} \Delta(\mathcal{S}_{i-1}, e_i) - (v_{n'} - v_n) < -\eta_{\mathsf{w}} - w(\varepsilon). \tag{11}$$

According to theorem 4.11, with probability $1-\varepsilon$, for any event n' which is $w_2(\varepsilon)$ rounds later than event n, inequality 11 holds.

Let round r_{young} denote the largest round that exists $\mathbf{b}$ such that $\mathbf{B}^{\mathsf{min}}_{N(r_0)} \nsubseteq \mathbf{b}.\mathsf{past}$ and $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_{N(r_{\mathsf{young}})}, \mathbf{b}) = \mathsf{False}$. According to theorem 4.12, with probability at least $1-\varepsilon$,

$$r_{\mathsf{young}} \leq r_0 + w_3(\varepsilon).$$

Notice that $n < N(r_{\mathsf{young}}+1)$ by definition, so we claim for any $n' > N(r_0 + w_3(\varepsilon) + w_2(\varepsilon))$, there will be

- $P(\mathcal{S}_{n'}, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) < -\eta_{\mathsf{w}}$
- For any block $\mathbf{b} \in \mathbf{B}^{\mathsf{gen}}_{n'}$ with $\mathbf{B}^{\mathsf{min}}_{N(r_0)} \nsubseteq \mathbf{b}.\mathsf{past}$, it will be $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_{n'}, \mathbf{b}) = \mathsf{True}$.

According to theorem 4.7, the history of blocks in $\mathbf{B}^{\mathsf{min}}_{N(r_0)}$ will not be changed after round $r_0 + w_2(\varepsilon) + w_3(\varepsilon)$.

In all, the blocks released before round $r_0 - d$ will be confirmed after round $r_0 + w_2(\varepsilon) + w_3(\varepsilon)$ with probability $1-3\varepsilon$. Notice that $w_2(\varepsilon) + w_3(\varepsilon) + d = d \cdot O\left(\max\left\{\frac{\log(1/(\varepsilon\delta))}{\delta^3}, \frac{\eta_{\mathsf{b}}}{\delta^2}\right\}\right)$, we have proved this theorem. □

5 Confirmation Policy

In order to show the low confirmation delay of GHAST protocol, we provide a concrete method in estimating the block confirmation risk.

Consider a block $\mathbf{b}'$ in Tree-Graph structure and suppose $\mathbf{b}'$ is in the past set of a pivot block $\mathbf{b}$. Then the finalization of $\mathbf{b}'$ can be reduced to block $\mathbf{b}$. Given a local state $\mathcal{B}$ and block $\mathbf{b} \in \mathrm{Pivot}(\mathcal{B})$, if block $\mathbf{b}$ has more subtree weight than one of its siblings, block $\mathbf{b}$ may be kicked out of the pivot chain and the history of block $\mathbf{b}'$ may change. So in this section, we study the probability that block $\mathbf{b}$ is kicked out of pivot chain in the future under the assumption that the blocks in $\mathrm{Chain}(\mathbf{b}.\mathsf{parent})$ are always the common pivot chain.

For simplicity, we ignore the network delay and assume all the blocks are delivered to all the honest nodes at once. But we still allow the attacker to withhold blocks. The confirmation policy considering the network delay can have a similar idea, except assuming an honest participant can not see the newly generated honest block in past time $2d$.

The confirmation rule consists of two parts. First, we estimate the confirmation risk under an assumption that for the first θ blocks $\mathbf{b}'$ generated later that block $\mathbf{b}.\mathsf{parent}$, if $\mathbf{b}'$ is in the subtree of block $\mathbf{b}'.\mathsf{parent}$, there will be $\mathrm{Adapted}(\mathbf{b}') = \mathsf{False}$. θ can be an arbitrary positive integer. Second, we estimate the probability that such assumption is break. By the union bound, we get the confirmation risk finally.

5.1 Confirmation risk under an assumption

Here, we define two variables m and n for a Tree-Graph $\mathbf{B}$. m and n shadow the same symbols defined in execution model and security analysis.

$$m := \text{An } \textit{upper bound} \text{ for the number of honest blocks generated later than } \mathbf{b}.\mathsf{parent} \tag{12}$$

$$n := \text{A } \textit{lower bound} \text{ for subtree weight advantage compared between } \mathbf{b} \text{ and its siblings} \tag{13}$$

(When computing the subtree weight of $\mathbf{b}$'s sliblings, only honest blocks are taken into account.)

The confirmation risk is computed conditioned on given m and n. In estimation for value m, notice that the blocks in $\mathbf{b}.\mathsf{parent}.\mathsf{past}$ must be generated earlier than $P(\mathbf{b})$, we can just count the number of blocks in $\mathbf{b} \backslash \mathbf{b}.\mathsf{parent}.\mathsf{past}$ as the value of m. In estimation for value n, we try to distinguish the malicious blocks as many as possible. Then we compare the subtree weight of $\mathbf{b}$ and its maximum siblings for the normal case. If we fail to distinguish malicious block, we will get a more conservative confirmation risk estimation.

Let K denote the total weight of malicious block in the subtree of block $\mathbf{b}.\mathsf{parent}$, T denote the number of blocks generated after the generation of $\mathbf{b}.\mathsf{parent}$. Although K and T are fixed in the view of execution model, but a participants can not know the exact number of how many malicious blocks generated in a given time interval. So we regard K and T as random variables and discuss their probability distribution conditioned on m, n.

Let random variable X_0 denote the current difference between subtree weight of block $\mathbf{b}$ and the maximum subtree weight of $\mathbf{b}$'s sibling blocks (including the blocks withheld by the attacker). Then it will be $X_0 \geq n - K$. Let X_i denote the difference after i blocks generation. So block $\mathbf{b}$ will be kicked out of pivot chain only if $X_i < 0$. Notice that $X_{i+1} - X_i$ is independent with K and T because their randomness are from different time interval. So we can regard K and T as fixed value in discussing $X_{i+1} - X_i$.

We have assume that for the first θ blocks $\mathbf{b}'$ generated later than $\mathbf{b}$.parent, if block $\mathbf{b}'$ is in the subtree of $\mathbf{b}$.parent, its block weight must be 1. [8] Let $\tilde{\theta} = \max\{\theta - T, 0\}$. When $i \leq \tilde{\theta}$, each time an honest node generate a block, X_i will increase by 1. Each time an malicious node generates a block, X_i will decrease by at most one. We have the following claim:

$$X_i - X_{i-1} \geq \begin{cases} 1 & \text{The } i\text{-th block is generated by honest nodes.} \\ -1 & \text{The } i\text{-th block is generated by the attacker.} \end{cases}$$

When $i > \tilde{\theta}$, the i-th block could be a normal block or an adapted block. Since the block weight never exceed η_{w}, we have $-\eta_{\mathsf{w}} \leq X_{i+1} - X_i \leq \eta_{\mathsf{w}}$. Since the normal block and adapted block have the same block weight expectation, similarly, we have

$$\mathbb{E}[X_i - X_{i-1} | X_{i-1}, \cdots, X_0] \geq \begin{cases} 1 & \text{The } i\text{-th block is generated by honest nodes.} \\ -1 & \text{The } i\text{-th block is generated by the attacker.} \end{cases}$$

Since the adversary shares β computing power, we have $\mathbb{E}[X_i - X_{i-1} | X_{i-1}, \cdots, X_0] = 1 - 2\beta$. Thus

$$\forall s > 0, \mathbb{E}[e^{s(X_{i-1} - X_i)} | X_{i-1}, \cdots, X_0] \leq \begin{cases} e^{s\beta} + e^{-s(1-\beta)} & i \leq \tilde{\theta} \\ \frac{1}{\eta_{\mathsf{w}}} \cdot \left(e^{s\eta_{\mathsf{w}}\beta} + e^{-s\eta_{\mathsf{w}}(1-\beta)}\right) & i > \tilde{\theta} \end{cases}$$

We use $g_1(s)$ to denote $e^{s\beta} + e^{-s(1-\beta)}$ and $g_2(s)$ to denote $\frac{1}{h} \cdot \left(e^{sh\beta} + e^{-sh(1-\beta)}\right)$. We have

$$\forall s > 0, \mathbb{E}[e^{s(X_0 - X_i)} | X_{i-1}, \cdots, X_0] \leq g_1(s)^{\min\{i, \tilde{\theta}\}} \cdot g_2(s)^{\max\{0, i - \tilde{\theta}\}}.$$

According to Markov's inequality,

$$\forall c > 0, \forall i \geq 0, \Pr[X_0 - X_i \geq c] \leq \min_s g_1(s)^{\min\{i, \tilde{\theta}\}} \cdot g_2(s)^{\max\{0, i - \tilde{\theta}\}} \cdot e^{-tc}.$$

Recalling $X_0 \geq n - K$, applying the union bound,

$$\Pr[\exists i \geq 0, X_i \leq 0] \leq \min\left\{1, \sum_{i=1}^{+\infty} \min_s \left(g_1(s)^{\min\{i, \tilde{\theta}\}} \cdot g_2(s)^{\max\{0, i - \tilde{\theta}\}} \cdot e^{s(K-n)}\right)\right\}.$$

Let random variable K' denote the number of malicious blocks generated from the creation of $\mathbf{b}$.parent to the current time. So $T \leq m + K'$. During the time interval that honest nodes generates $m + 1$ blocks, the number of malicious blocks follows the negative binomial distribution with $m + 1$ successes and $1 - \beta$ success probability. Thus

$$\forall k > 0, \Pr[K' \geq k] \leq I_{1-\beta}(k, m + 1). \quad (I \text{ is regularized beta function.}).$$

When $K' + m \leq \theta$, all the K' malicious blocks will be normal blocks with weight one except the blocks not in subtree of $\mathbf{b}$.parent. Thus $K \leq K'$.

Let $p(K, T)$ equal the right hand side of inequality (14). Notice that $p(K, T)$ is non-decreasing in terms of K and T and $p(n, T) = 1$. For any given $t \leq \theta$, the confirmation risk will be

[8] But we don't discuss the probability conditioned on the assumption. If we want to learn the probability of event E, the probability under the assumption A means $\Pr[E \wedge A]$ and the probability conditioned on the assumption A means that $\Pr[E|A]$.

$$\begin{aligned}\mathbb{E}_{K,T}[p(K,T)] \leq & \Pr[T > t] + \mathbb{E}_K[p(K,t)] \\ \leq & \Pr[K' + m > t] + p(0,t) + \sum_{k=1}^{n} \Pr[K \geq k] \cdot (p(k,t) - p(k-1,t)) \\ \leq & \Pr[K' + m > t \vee K' + m > \theta] + p(0,t) + \sum_{k=0}^{n-1} \Pr[K' \geq k] \cdot p(k,t) \\ \leq & I_{1-\beta}(t - m + 1, m + 1) + p(0,t) + \sum_{k=0}^{n-1} I_{1-\beta}(k, m+1) \cdot p(k,t)\end{aligned}$$

5.2 Risk for breaking the assumption

The previous computation assumes the GHAST weight adaption is not triggered under the subtree of $\mathbf{b}.\mathsf{parent}$ during the generation of θ blocks. In this subsection, we provide a concrete method to estimate the risk that the attacker breaks this assumption. The θ^{th} block generation time after $\mathbf{b}.\mathsf{parent}$ is called the *deadline*.

If a block $\mathbf{c}$ in subtree of $\mathbf{b}.\mathsf{parent}$ satisfying $\mathrm{Adapted}(\mathbf{c}) = \mathsf{True}$. There must exists block $\mathbf{a} \in \mathrm{Chain}(\mathbf{c})$ such that

1. $\mathrm{MaxTH}(\mathbf{c}.\mathsf{past}) - \mathrm{TimerHeight}(\mathbf{a}.\mathsf{parent}) > \eta_{\mathsf{b}}$.
2. $\mathrm{SubTW}(\mathbf{c}.\mathsf{past}, \mathbf{a}) - \mathrm{SibSubTW}(\mathbf{c}.\mathsf{past}, \mathbf{a}) \leq \eta_{\mathsf{a}}$.

Let event $E_1(\mathbf{a})$ and $E_2(\mathbf{a})$ denote that there exists block $\mathbf{c}$ generated before the deadline in subtree of $\mathbf{b}.\mathsf{parent}$ satisfying the first and the second condition respectively. Then the risk that the assumption is broken is no more than

$$\Pr[\exists \mathbf{a}, E_1(\mathbf{a}) \wedge E_2(\mathbf{a})] \leq \min\{\Pr[\exists \mathbf{a}, E_1(\mathbf{a})], \Pr[\exists \mathbf{a}, E_2(\mathbf{a})]\}. \tag{14}$$

Now, we will discuss how to compute $\Pr[E_1(\mathbf{a})]$ and $\Pr[E_2(\mathbf{a})]$ for fixed block $\mathbf{a}$.

Let Z denote the maximum timer height among all the generated blocks at the creation of $\mathbf{b}.\mathsf{parent}$ (including the blocks withheld by the adversary). A necessary condition for $E_1(\mathbf{c})$ is that there are $\eta_{\mathsf{b}} - (Z - \mathrm{TimerHeight}(\mathbf{a}.\mathsf{parent}))$ timer blocks among the first θ blocks after the generation of $\mathbf{b}.\mathsf{parent}$. Thus

$$\Pr[E_1(\mathbf{a})] \leq \Pr[\mathrm{B}(\theta, 1/\eta_{\mathsf{t}}) \leq \eta_{\mathsf{b}} - (Z - \mathrm{TimerHeight}(\mathbf{a}.\mathsf{parent})] \qquad \mathrm{B} \text{ denotes the binomial distribution} \tag{15}$$

The value of $Z - \mathrm{MaxTH}(\mathbf{b}.\mathsf{parent})$ is upper bounded by the number of consecutive malicious blocks in the timer chain, which can be estimated by chain-quality property in Nakamoto consensus proved by Pass et al [19]. We will formally reason it in the later version.

In the estimation for $\Pr[E_2(\mathbf{a})]$, we only consider the block $\mathbf{a}$ with $\mathbf{a} \in \mathrm{Chain}(\mathbf{b}.\mathsf{parent})$. Since $\mathbf{c}$ is in the subtree of $\mathbf{b}.\mathsf{parent}$, it will be $\mathbf{b}.\mathsf{parent}.\mathsf{past} \subseteq \mathbf{c}.\mathsf{past}$. Thus $\mathrm{SubTW}(\mathbf{c}.\mathsf{past}, \mathbf{a}) \geq \mathrm{SubTW}(\mathbf{b}.\mathsf{parent}.\mathsf{past}, \mathbf{a})$. Let $w := \mathrm{SubTW}(\mathbf{b}.\mathsf{parent}.\mathsf{past}, \mathbf{a})$ is a known value. So $E_2(\mathbf{a})$ happens only if

$$\mathrm{SibSubTW}(\mathbf{c}.\mathsf{past}, \mathbf{a}) \geq w - \eta_{\mathsf{a}}.$$

We takes two parameters as input,

$m :=$ An *upper bound* for the number of honest blocks generated later than $\mathbf{a}.\mathsf{parent}$ and before $\mathbf{b}.\mathsf{parent}$

$l :=$ An *upper bound* for the total weight of honest blocks contributes to the subtree of sliblings blocks of $\mathbf{a}$

So the malicious blocks must generate at least $w - \eta_{\mathsf{a}} - l$ block weights between the creation of $\mathbf{a}.\mathsf{parent}$ and the deadline. The probability estimation for this events compose of two steps: 1) the number N of blocks that the adversary generates in this time interval; 2) If the adversary is allowed to generated $N = n$ blocks and switch its consensus strategy adaptively, how many block weights it can generated. We can choose proper n and claim that if the

adversary generates $w - \eta_{\mathsf{a}} - l$ block weights during the time interval, the adversary must be either generates more than n blocks during the time interval or generates $w - \eta_{\mathsf{a}} - l$ block weights in the first n blocks.

Let N_1 be the number of malicious blocks generated between the creation of $\mathbf{a}.\mathsf{parent}$ and $\mathbf{b}.\mathsf{parent}$ and N_2 be the number of malicious blocks generated between the creation of $\mathbf{b}.\mathsf{parent}$ and the deadline. So $N = N_1 + N_2$. N_1 is upper bounded by the negative binomial distribution with $m+1$ successes and β success probability, N_2 is a binomial distribution with θ trials and β success probability. By the union bound, we have

$$\forall n_1, n_2, \Pr[N \geq n_1 + n_2] \leq \Pr_{N_1 \sim \mathrm{NB}(m+1,1-\beta)}[N_1 \geq n_1] + \Pr_{N_2 \sim \mathrm{B}(\theta,\beta)}[N_2 \geq n_2].$$

If the adversary is allowed to generate n blocks during the this interval. Let Y_i denote the total block weight after the generation of i^{th} block. Then we have $Y_0 = 0$ and

$$\mathbb{E}[Y_i - Y_{i-1} | Y_{i-1}, \cdots, Y_0] = 1.$$

Notice that $Y_i \leq \eta_{\mathsf{w}}$, thus

$$\forall t > 0, \mathbb{E}[e^{t(Y_i - Y_{i-1})} | Y_{i-1}, \cdots, Y_0] \leq \frac{\eta_{\mathsf{w}} - 1 + e^{t\eta_{\mathsf{w}}}}{\eta_{\mathsf{w}}}.$$

So we have

$$\Pr[Y_n \geq \rho] \leq \min_t \left(\left(\frac{\eta_{\mathsf{w}} - 1 + e^{t\eta_{\mathsf{w}}}}{\eta_{\mathsf{w}}} \right)^n \cdot e^{-t\rho} \right).$$

So an upper bound for probability of $E_2(\mathbf{a})$ is $\Pr[N \geq n] + \Pr[Y_n \geq w - \eta_{\mathsf{a}} - l]$.

Taking the union bound over all the possible $\mathbf{a}$ gives the risk that an attacker generated an adaptive weighted block before the deadline. In order to achieve a better result, we compute the probability of $E_1(\mathbf{a}) \wedge E_2(\mathbf{a})$ in batch. Formally, we split $\mathrm{Chain}(\mathbf{b}.\mathsf{parent})$ into several slices. For each slice $\mathbf{S}$, we compute the probability that

$$\Pr[\exists \mathbf{c} \in \mathbf{S}, E_1(\mathbf{c}) \wedge E_2(\mathbf{c})] \leq \min\{\Pr[\exists \mathbf{c} \in \mathbf{S}, E_1(\mathbf{c})], \Pr[\exists \mathbf{c} \in \mathbf{S}, E_2(\mathbf{c})]\}.$$

Let $\mathbf{c}'$ be the oldest block in $\mathbf{S}$, then $\Pr[\exists \mathbf{c} \in \mathbf{S}, E_1(\mathbf{c})] = \Pr[E_1(\mathbf{c}')]$. In estimation of $\Pr[\exists \mathbf{c} \in \mathbf{S}, E_2(\mathbf{c})]$, we pick the maximum m, l and minimum w among all the blocks $\mathbf{a}$ in the slice. (Notice that m, l, w are defined on given block $\mathbf{a}$ in the estimation of $\Pr[E_2(\mathbf{a})]$.) This gives a better result.

6 Implementation

A separated work [14] implements this protocol in the Conflux blockchain system. Under the experiment with 20Mbps network bandwidth limit per node, Conflux chooses the block generation rate of 4 blocks per second and the block size limit of 300K. Under this parameter, Conflux achieves a block throughput of 9.6Mbps. The experiments shows that all the blocks can be propagated to 99% full nodes in 15 seconds, which implies $\lambda = 60$. (λ is defined in section 4.4.) Conflux chooses the consensus protocol parameters $\vec{\eta}$ to tolerate liveness attacks from a powerful attacker that controls 40% of the network computation power. (a.k.a. $\beta = 0.4$, $\delta = 1 - \beta/(1-\beta) = 1/3$.) We choose the parameter $\eta_{\mathsf{w}} = 600$, [9] $\eta_{\mathsf{t}} = 2\lambda/\delta = 360$ and $\eta_{\mathsf{a}} = 3\eta_{\mathsf{w}}$.

Waiting for six blocks in Bitcoin has the confirmation risk 2×10^{-5} with $\beta = 0.1$ adversary. To obtain the same confidence as waiting for six blocks in Bitcoin in a short time, we set $\eta_{\mathsf{b}} = 160$ to make the risk output by the computation policy small enough. In the global view, a block can be confirmed in less than three times of network delay.

[9] Theorem 4.13 requires $\eta_{\mathsf{w}} = 30\lambda/\delta$. This requirement derives from lemma B.9. With the concrete parameter $\lambda = 60$ and $\delta = 1/3$, we find a smaller solution $\eta_{\mathsf{w}} = 600$ which achieves the same results as lemma B.9 in complexity.

7 Conclusion

In this work, we design a novel consensus protocol that achieves security and low confirmation delay in a normal scenario. The protocol executes two consensus strategies to achieve efficiency in normal cases and switch to a conservative strategy in defending a liveness attack. We provide a rigorous security analysis to show such design can resolve all the possible liveness issues in GHOST protocol. Any block will become $\varepsilon-$finalized after a logarithm time after its release. A separated work [14] implements this protocol and shows that the block can be confirmed in less than $3d$ in the global view.

References

[1] Snowflake to Avalanche: A Novel Metastable Consensus Protocol Family for Cryptocurrencies. https://ipfs.io/ipfs/QmUy4jh5mGNZvLkjies1RWM4YuvJh5o2FYopNPVYwrRVGV.

[2] BAGARIA, V., KANNAN, S., TSE, D., FANTI, G., AND VISWANATH, P. Prism: Deconstructing the blockchain to approach physical limits. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019), pp. 585–602.

[3] DELOITTE. 5 blockchain technology use cases in financial services. https://www2.deloitte.com/nl/nl/pages/financial-services/articles/5-blockchain-use-cases-in-financial-services.html.

[4] DELOITTE. Blockchain: Opportunities for health care. https://www2.deloitte.com/us/en/pages/public-sector/articles/blockchain-opportunities-for-health-care.html, 2018.

[5] EYAL, I., GENCER, A. E., SIRER, E. G., AND VAN RENESSE, R. Bitcoin-ng: A scalable blockchain protocol. In *NSDI* (2016), pp. 45–59.

[6] EYAL, I., AND SIRER, E. G. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security* (2014), Springer, pp. 436–454.

[7] GARAY, J., KIAYIAS, A., AND LEONARDOS, N. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2015), Springer, pp. 281–310.

[8] GILAD, Y., HEMO, R., MICALI, S., VLACHOS, G., AND ZELDOVICH, N. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles* (2017), ACM, pp. 51–68.

[9] IBM. Blockchain for supply chain. https://www.ibm.com/blockchain/supply-chain/.

[10] KIAYIAS, A., AND PANAGIOTAKOS, G. On trees, chains and fast transactions in the blockchain. In *International Conference on Cryptology and Information Security in Latin America* (2017), Springer, pp. 327–351.

[11] KIFFER, L., RAJARAMAN, R., AND SHELAT, A. A better method to analyze blockchain consistency. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (2018), pp. 729–744.

[12] KOGIAS, E. K., JOVANOVIC, P., GAILLY, N., KHOFFI, I., GASSER, L., AND FORD, B. Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th USENIX Security Symposium (USENIX Security 16)* (2016), pp. 279–296.

[13] LEWENBERG, Y., SOMPOLINSKY, Y., AND ZOHAR, A. Inclusive block chain protocols. In *International Conference on Financial Cryptography and Data Security* (2015), Springer, pp. 528–547.

[14] LI, C., LI, P., ZHOU, D., YANG, Z., WU, M., XU, W., LONG, F., AND YAO, A. A decentralized blockchain with high throughput and fast confirmation. In *USENIX Annul Technical Conference* (2020), USENIX.

[15] MAZIERES, D. The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation* (2015).

[16] MILLER, A., XIA, Y., CROMAN, K., SHI, E., AND SONG, D. The honey badger of bft protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), ACM, pp. 31–42.

[17] NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf.

[18] NATOLI, C., AND GRAMOLI, V. The balance attack against proof-of-work blockchains: The r3 testbed as an example. *arXiv preprint arXiv:1612.09426* (2016).

[19] PASS, R., SEEMAN, L., AND SHELAT, A. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2017), Springer, pp. 643–673.

[20] PASS, R., AND SHI, E. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing* (2017), ACM, pp. 315–324.

[21] PASS, R., AND SHI, E. Hybrid consensus: Efficient consensus in the permissionless model. In *LIPIcs-Leibniz International Proceedings in Informatics* (2017), vol. 91, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[22] SOMPOLINSKY, Y., LEWENBERG, Y., AND ZOHAR, A. Spectre: Serialization of proof-of-work events: confirming transactions via recursive elections, 2016.

[23] SOMPOLINSKY, Y., AND ZOHAR, A. Phantom, a scalable blockdag protocol. https://eprint.iacr.org/2018/104.pdf.

[24] SOMPOLINSKY, Y., AND ZOHAR, A. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security* (2015), Springer, pp. 507–527.

[25] YU, H., NIKOLIC, I., HOU, R., AND SAXENA, P. Ohie: Blockchain scaling made simple. *arXiv preprint arXiv:1811.12628* (2018).

A Potential Values

A.1 Preparation

First we remind the most frequently used notations in this section. Functions $\mathrm{Chain}(\cdot)$,$\mathrm{SubT}(\cdot)$, $\mathrm{SubTW}(\cdot)$ are $\mathrm{SibSubTW}(\cdot)$ are defined in section 3. Functions $\mathrm{TotalW}(\cdot)$, $\mathrm{Tip}(\cdot)$ and Next are defined in the beginning of section 4.3. There are four types of events hGenRls, mGen, mRls and Arvl defined in section 4.3.1. Recalling that we use symbols $\mathbf{B}^{\mathsf{gen}}$, $\mathbf{B}^{\mathsf{max}}$, $\mathbf{B}^{\mathsf{min}}$, $\mathbf{M}$, $\mathbf{f}$, $\mathbf{C}$, $\mathbf{S}$, v to denote corresponding components of adversary state $\mathcal{S}$ in the context and use the symbols with subscript "$_{-1}$" to denote components of $\mathcal{S}_{-1}$. These symbols will never be used as ephemeral symbols. $\mathbf{b}_1 \preceq \mathbf{b}_2$ represents the relation $\mathbf{b}_1 \in \mathrm{Chain}(\mathbf{b}_2)$. If furthermore there is $\mathbf{b}_1 \neq \mathbf{b}_2$, we write $\mathbf{b}_1 \prec \mathbf{b}_2$. Since the local state $\mathcal{B}$ of a participant node is essentially a graph $\mathbf{B}$, we use $\mathbf{B}$ to denote the local state here.

We claim the necessary condition and sufficient condition for a block $\mathbf{b} \in \mathbf{B}$ be in the pivot chain $\mathrm{Pivot}(\mathbf{B})$.

Lemma A.1 *For any graph* $\mathbf{B}$ *and a block* $\mathbf{b}$ *in graph* $\mathbf{B}$*, let* $\mathbf{g}$ *be the genesis block, the sufficient condition and necessary condition of* $\mathbf{b} \in \mathrm{Pivot}(\mathbf{B})$.

- *Sufficient condition:* $\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b}) \backslash \{\mathbf{g}\}$, $\mathrm{SubTW}(\mathbf{B}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}') > 0$
- *Necessary condition:* $\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b})$, $\mathrm{SubTW}(\mathbf{B}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}') \geq 0$

Proof. Recalling that the pivot chain is a list of blocks which starts with the genesis block $\mathbf{g}$ and recursively expands the best child (defined in equation 5) of last block into it. So the necessary and sufficient condition for $\mathbf{b} \in \mathrm{Pivot}(\mathbf{B})$ is $\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b}) \backslash \{\mathbf{g}\}, \mathbf{b}' = \mathrm{BestChild}(\mathbf{B}, \mathbf{b}'.\mathsf{parent})$. ($\mathbf{g}$ is the genesis block.) The definition of best child in eq. (5) shows that the sufficient condition and necessary condition for $\mathbf{b}' = \mathrm{BestChild}(\mathbf{B}, \mathbf{b}'.\mathsf{parent})$

- Sufficient condition: $\mathrm{SubTW}(\mathbf{B}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}') > 0$
- Necessary condition: $\mathrm{SubTW}(\mathbf{B}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}') \geq 0$

So we have the necessary condition and sufficient condition for pivot chain.

- Sufficient condition: $\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b}) \backslash \{\mathbf{g}\}$, $\mathrm{SubTW}(\mathbf{B}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}') > 0$
- Necessary condition: $\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b}) \backslash \{\mathbf{g}\}$, $\mathrm{SubTW}(\mathbf{B}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}') \geq 0$

Specially, the genesis block doesn't have any sibling blocks, so $\mathrm{SibSubTW}(\mathbf{B}, \mathbf{g}) = 0$ always holds. Thus $\mathrm{SubTW}(\mathbf{B}, \mathbf{g}) - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{g}) \geq 0$ can be also a necessary condition. □

We claim some properties which derives from the definitions directly. We do not explicitly refer the previous two claims in use since they are intuitive.

By the definition of $\mathrm{SubT}(\mathbf{B}, \mathbf{b})$, $\mathrm{SubTW}(\mathbf{B}, \mathbf{b})$ and $\mathrm{SibSubTW}(\mathbf{B}, \mathbf{b})$ in eq. (2,3,6), when the graph $\mathbf{B}$ includes more blocks, the outputs of these three functions will be non-decreasing. Formally, we have the following claim.

Claim A.2 *For any set of blocks* $\mathbf{B}$ *and a block* $\mathbf{b}$*, we have* $\mathrm{SubT}(\mathbf{B}, \mathbf{b}) \subseteq \mathbf{B}$.

For any set of blocks $\mathbf{B}_1$ *and* $\mathbf{B}_2$ *with* $\mathbf{B}_1 \subseteq \mathbf{B}_2$ *and a block* $\mathbf{b}$*, we have* $\mathrm{SubT}(\mathbf{B}_2, \mathbf{b}) = \mathrm{SubT}(\mathbf{B}_1, \mathbf{b}) \cup \mathrm{SubT}(\mathbf{B}_2 \backslash \mathbf{B}_1, \mathbf{b})$, $\mathrm{SubTW}(\mathbf{B}_2, \mathbf{b}) = \mathrm{SubTW}(\mathbf{B}_1, \mathbf{b}) + \mathrm{SubTW}(\mathbf{B}_2 \backslash \mathbf{B}_1, \mathbf{b})$ *and* $\mathrm{SibSubTW}(\mathbf{B}_1, \mathbf{b}) \leq \mathrm{SibSubTW}(\mathbf{B}_2, \mathbf{b})$.

For any set of blocks $\mathbf{B}$ *and blocks* $\mathbf{b}_1, \mathbf{b}_2$ *with* $\mathbf{b}_1 \preceq \mathbf{b}_2$*, we have* $\mathrm{SubT}(\mathbf{B}, \mathbf{b}_2) \subseteq \mathrm{SubT}(\mathbf{B}, \mathbf{b}_1)$ *and* $\mathrm{SubTW}(\mathbf{B}, \mathbf{b}_2) \leq \mathrm{SubTW}(\mathbf{B}, \mathbf{b}_1)$.

Recalling that protocol Π_{GHAST} is the same as Π_{TG} except the block weight. In Π_{TG}, the validity of a block requires $\mathrm{Pivot}(\mathbf{b}.\mathsf{past}) \circ \mathbf{b} = \mathrm{Chain}(\mathbf{b})$. So the ghast protocol inherits such property.

Claim A.3 *For any block* $\mathbf{b}$*, we have* $\mathrm{Pivot}(\mathbf{b}.\mathsf{past}) \circ \mathbf{b} = \mathrm{Chain}(\mathbf{b})$.

Claim A.4 *Let* $\mathbf{P}$ *be a list of blocks in which any two consecutive blocks are in parent/child relation. (The outputs of* $\mathrm{Chain}(\mathbf{b}')$ *for some block* $\mathbf{b}'$ *and the chain* $\mathbf{C}$ *in the adversary state satisfy such requirement.)*

For any $\mathbf{b} \in \mathbf{P} \backslash \{\mathrm{Tip}(\mathbf{P})\}$*, let* $\mathbf{b}_1 := \mathrm{Next}(\mathbf{P}, \mathbf{b})$*, it will be* $\mathbf{b}_1 \neq \perp$ *and* $\mathbf{b}_1.\mathsf{parent} = \mathbf{b}$.

A.2 Properties for concepts

Before touching the potential value, we prove some important properties for concepts first.

Proof of Lemma 4.4. For any block $\mathbf{b} \in \mathbf{B}^{\mathsf{max}}$ and its children blocks $\mathbf{b}_1, \mathbf{b}_2 \in \mathrm{Chldn}(\mathbf{B}^{\mathsf{max}}, \mathbf{b})$ ($\mathbf{b}_1 \neq \mathbf{b}_2$). Since $\mathrm{SibSubTW}(\mathbf{B}, \mathbf{b})$ (defined in eq. (6))returns the maximum subtree weight of sibling blocks in $\mathbf{b}$, and block $\mathbf{b}_1$ and $\mathbf{b}_2$ are on sibling relationship, we can claim $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_1) \geq \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_2)$ and $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_2) \geq \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_1)$. Because block $\mathbf{f}$ is in $\mathbf{B}^{\Delta}$, we have $\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\} \subseteq \mathbf{B}^{\mathsf{max}}$. Thus

$$\begin{aligned}
&\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}_1) + \mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}_2) \\
\leq &\mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_1) - \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_2) + \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_2) - \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_1) \\
\leq &0
\end{aligned}$$

Thus it can not be $\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}_1) > 0$ and $\mathrm{Adv}((\mathbf{B}^{\mathsf{max}}, \mathbf{B}^{\mathsf{min}}, \mathbf{f}), \mathbf{b}_2) > 0$. □

Lemma A.5 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. If flag block $\mathbf{f}_{-1} = \perp$, then $\mathbf{C}_{-1}$ must be a prefix of pivot chain $\mathrm{Pivot}(e.\mathsf{block})$.*

Proof. Let $\mathbf{b} := e.\mathsf{block}$, $\mathbf{B} := \mathbf{b}.\mathsf{past}$. Since $\mathbf{f}_{-1} = \perp$, for any block $\mathbf{b}' \in \mathbf{B}^{\mathsf{max}}_{-1}$, $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}') = \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}_{-1}, \mathbf{b}'\big) - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1}, \mathbf{b}'\big)$. By claim 4.5.1, for any block $\mathbf{b}' \in \mathbf{C}_{-1}$, $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}') > 0$. Since $\mathbf{B}_{-1}$ is the local state of the honest node who generates $\mathbf{b}$, according to claim 4.1, $\mathbf{B}^{\mathsf{min}}_{-1} \subseteq \mathbf{B}_{-1} \subseteq \mathbf{B}^{\mathsf{max}}_{-1}$. So we have

$$\forall \mathbf{b}' \in \mathbf{C}_{-1}, \mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}') \geq \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}_{-1}, \mathbf{b}'\big) - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1}, \mathbf{b}'\big) > 0.$$

According to lemma A.1, block $\mathrm{Tip}(\mathbf{C}_{-1})$ must be in $\mathrm{Pivot}(\mathbf{B}_{-1})$ and thus $\mathbf{C}_{-1}$ is a prefix of pivot chain $\mathrm{Pivot}(\mathbf{B}_{-1})$ when $\mathbf{f}_{-1} = \perp$. Since $\mathrm{Pivot}(\mathbf{B}_{-1}) \circ \mathbf{b} = \mathrm{Chain}(\mathbf{b})$, $\mathbf{C}_{-1}$ is also a prefix of $\mathrm{Chain}(\mathbf{b})$. □

Lemma A.6 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, any graph $\mathbf{B}$ satisfying $\mathbf{B}^{\mathsf{min}} \subseteq \mathbf{B} \subseteq \mathbf{B}^{\mathsf{max}}$ and any block $\mathbf{b}$ which is not genesis block. We have*

$$\mathrm{SubTW}(\mathbf{B}, \mathbf{b}) - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b}) \leq \mathrm{Adv}(\mathcal{S}, \mathbf{b}) + \mathrm{SubTW}\big(\mathbf{B}^{\Delta}, \mathbf{b}.\mathsf{parent}\big).$$

Proof. By the definition of $\mathrm{SubTW}(\cdot)$, for any block $\mathbf{b}$, we have

$$\mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}) - \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}\big) = \mathrm{SubTW}\big(\mathbf{B}^{\Delta}, \mathbf{b}\big).$$

As for the upper bound of $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}) - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}\big)$ for any block $\mathbf{b}$ except the genesis block $\mathbf{g}$. If $\mathrm{Chldn}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}.\mathsf{parent})\backslash\{\mathbf{b}\}$ is empty set, then $\mathrm{Chldn}(\mathbf{B}^{\mathsf{min}}, \mathbf{b}.\mathsf{parent})\backslash\{\mathbf{b}\}$ will also be empty set and $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}) - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}\big) = 0$. Otherwise let $\mathbf{b}'$ be the block with maximum subtree weight in $\mathrm{Chldn}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}'.\mathsf{parent})\backslash\{\mathbf{b}'\}$. Then $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}') = \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}')$. If $\mathbf{b}' \notin \mathrm{Chldn}(\mathbf{B}^{\mathsf{min}}, \mathbf{b}.\mathsf{parent})\backslash\{\mathbf{b}\}$, it must be $\mathbf{b}' \in \mathbf{B}^{\Delta}$. So all the blocks in subtree of $\mathbf{b}'$ not appears in $\mathbf{B}^{\mathsf{min}}$, and thus $\mathrm{SubT}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}'\big) = 0$. If $\mathbf{b}' \in \mathrm{Chldn}(\mathbf{B}^{\mathsf{min}}, \mathbf{b}.\mathsf{parent})\backslash\{\mathbf{b}\}$, then $\mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}'\big) \geq \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}'\big)$. In summary for the case $\mathbf{b}'$ exists, we have

$$\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}') - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}'\big) \leq \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}') - \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}'\big) = \mathrm{SubTW}\big(\mathbf{B}^{\Delta}, \mathbf{b}'\big).$$

Recalling that $\mathbf{b}'.\mathsf{parent} = \mathbf{b}.\mathsf{parent}$, we have the following result for both cases that $\mathbf{b}'$ exists or not.

$$\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}) - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}\big) \leq \mathrm{SubTW}\big(\mathbf{B}^{\Delta}, \mathbf{b}.\mathsf{parent}\big) - \mathrm{SubTW}\big(\mathbf{B}^{\Delta}, \mathbf{b}\big).$$

Notice that $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) \geq \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}\big) - \mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b})$ by definition. We have

$$\begin{aligned}
\mathrm{Adv}(\mathcal{S}, \mathbf{b}) + \mathrm{SubTW}\big(\mathbf{B}^{\Delta}, \mathbf{b}.\mathsf{parent}\big) \geq &\mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}) - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}\big) \\
\geq &\mathrm{SubTW}(\mathbf{B}, \mathbf{b}) - \mathrm{SibSubTW}(\mathbf{B}, \mathbf{b})
\end{aligned}$$

□

Lemma A.7 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. If e is the* hGenRls *event of a flag block (a.k.a. $\mathbf{f}_{-1} = \perp$ and $\mathbf{f} = e.\mathsf{block}$), then we have $\forall \mathbf{b}' \in \mathrm{Chain}(e.\mathsf{block}), \mathrm{Adv}(\mathcal{S}, \mathbf{b}') > \eta_{\mathsf{w}} - s_{\mathsf{h}} - s_{\mathsf{m}}$.*

Proof. We denote block $e.\mathsf{block}$ by $\mathbf{b}$ and denote graph $\mathbf{b}.\mathsf{past}$ by $\mathbf{B}_{-1}$. Since $\mathbf{b}$ is an honest block, $\mathbf{B}_{-1}$ must be the local state of an honest node when adversary state is $\mathcal{S}$, thus $\mathbf{B}_{-1}^{\mathsf{min}} \subseteq \mathbf{B}_{-1} \subseteq \mathbf{B}_{-1}^{\mathsf{max}}$ by claim 4.1. Since $\mathrm{Chain}(\mathbf{b})\backslash\{\mathbf{b}\} = \mathrm{Pivot}(\mathbf{B}_{-1})$, according to the necessary condition of pivot chain, for any block $\mathbf{b}'$ in $\mathrm{Chain}(\mathbf{b})\backslash\{\mathbf{b}\}$, $\mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}') \geq 0$. Since $\mathbf{b}.\mathsf{parent}$ is the last block of the pivot chain $\mathrm{Pivot}(\mathbf{B}_{-1})$, $\mathbf{B}_{-1}$ must have no child block in $\mathbf{B}_{-1}$. Thus $\mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}) = 0$. So we have

$$\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b}), \mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}') \geq 0.$$

Since $\mathbf{f}_{-1} = \perp$ and $\mathbf{B}_{-1}^{\mathsf{max}} \subseteq \mathbf{B}_{-1} \subseteq \mathbf{B}_{-1}^{\mathsf{min}}$, according to lemma A.6, we have

$$\begin{aligned}\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b})\backslash\{\mathbf{g}\}, \quad & \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}') + \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\Delta}, \mathbf{b}'.\mathsf{parent}\big) \\ \geq & \mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}') \\ \geq & 0.\end{aligned}$$

According to our rule in maintaining the flag block, when $\mathbf{f}_{-1} = \perp$ and $\mathbf{f} = e.\mathsf{block}$, $\mathbf{B}_{-1}^{\Delta}$ has no honest block with block weight η_{w} and $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$. Let $\mathbf{T}_{-1} := \mathrm{SubT}\big(\mathbf{B}_{-1}^{\Delta}, \mathrm{Tip}(\mathbf{C}_{-1})\big)$, according to the definition of special status (definition 4.2), we have $\mathrm{TotalW}(\mathbf{T}_{-1} \cap \mathbf{M}_{-1}) < s_{\mathsf{m}}$ and $|\{\mathbf{b} \in \mathbf{B}_{-1}^{\Delta}\backslash\mathbf{M}_{-1} \mid \mathbf{b}.\mathsf{weight} = 1\}| < s_{\mathsf{h}}$. Since there is no honest block with block weight η_{w} in $\mathbf{B}_{-1}^{\Delta}$, $|\{\mathbf{b} \in \mathbf{B}_{-1}^{\Delta}\backslash\mathbf{M}_{-1} \mid \mathbf{b}.\mathsf{weight} = \eta_{\mathsf{w}}\}| = 0$. Thus $\mathrm{TotalW}(\mathbf{T}_{-1}) < s_{\mathsf{m}} + s_{\mathsf{h}}$. Since $\mathbf{f}_{-1} = \perp$, according to lemma A.5, $\mathbf{C}_{-1}$ is a prefix of $\mathrm{Chain}(\mathbf{b})$. For any block $\mathbf{b}'$ in $\mathrm{Chain}(\mathbf{b})\backslash\mathbf{C}_{-1}$, we have $\mathrm{SubT}\big(\mathbf{B}_{-1}^{\Delta}, \mathbf{b}'.\mathsf{parent}\big) \subseteq \mathbf{T}_{-1}$. Thus,

$$\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b})\backslash\mathbf{C}_{-1}, \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\Delta}, \mathbf{b}'.\mathsf{parent}\big) \leq \mathrm{TotalW}(\mathbf{T}) < s_{\mathsf{m}} + s_{\mathsf{h}}.$$

Since genesis block $\mathbf{g}$ must be in $\mathbf{C}_{-1}$, for all the blocks $\mathbf{b}'$ in $\mathrm{Chain}(\mathbf{b})\backslash\mathbf{C}_{-1}$, the previous two inequalities give $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}') > -s_{\mathsf{h}} - s_{\mathsf{m}}$. According to claim 4.5.1, for the blocks $\mathbf{b}'$ in $\mathbf{C}_{-1}$, we have $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}') > 0$. Thus

$$\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b}), \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}') > -s_{\mathsf{h}} - s_{\mathsf{m}}.$$

According to the rule in updating adversary state, hGenRls event does not modify $\mathbf{B}^{\mathsf{min}}$, so $\mathbf{B}_{-1}^{\mathsf{min}} = \mathbf{B}^{\mathsf{min}}$. By claim 4.3, $\mathbf{f} \notin \mathbf{B}^{\mathsf{min}}$. For all the blocks $\mathbf{b}'$ in $\mathrm{Chain}(\mathbf{b})$, $\mathrm{SubT}(\{\mathbf{b}\}, \mathbf{b}') = \{\mathbf{b}\}$. Note that $\mathbf{b}$ and $\mathbf{f}$ refers the same block, and the block weight of flag block is η_{w}. So we have

$$\mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}, \mathbf{b}'\big) = \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}'\big) + \eta_{\mathsf{w}}.$$

For any block $\mathbf{b}' \in \mathrm{Chain}(\mathbf{b})$ and $\mathbf{b}_1' \in \mathrm{Chldn}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}'.\mathsf{parent})\backslash\{\mathbf{b}'\}$, it must be $\mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{b}_1'\big) = \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_1')$ because $\mathbf{B}^{\mathsf{max}}$ is one block $\mathbf{b}$ different from $\mathbf{B}_{-1}^{\mathsf{max}}$ and it is not in subtree of $\mathbf{b}_1'$. Thus

$$\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b}), \mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}') = \mathrm{SibSubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{b}'\big).$$

Summarize all the previous results, we have

$$\begin{aligned}\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{b}), \quad & \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}, \mathbf{b}'\big) - \mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}') \\ = & \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}'\big) - \mathrm{SibSubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{b}'\big) + \eta_{\mathsf{w}} \\ > & \eta_{\mathsf{w}} - s_{\mathsf{m}} - s_{\mathsf{h}}\end{aligned}$$

□

Now we will show that when the adversary state is updated from $\mathcal{S}_{-1}$ to $\mathcal{S}$, one of $\mathbf{C}_{-1}$ and $\mathbf{C}$ must be the prefix of another.

Lemma A.8 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. One of $\mathbf{C}_{-1}$ and $\mathbf{C}$ must be the prefix of another.*

Proof. We prove this property by contradiction. We assume $\mathbf{C}_{-1}$ is not the prefix of $\mathbf{C}$ and $\mathbf{C}$ is not the prefix of $\mathbf{C}_{-1}$. Let $\mathbf{b}_\mathsf{c}$ be the last block of common prefix between $\mathbf{C}$ and $\mathbf{C}_{-1}$. Let $\mathbf{b}_1 := \mathrm{Next}(\mathbf{C}_{-1}, \mathbf{b}_\mathsf{c})$, $\mathbf{b}_2 := \mathrm{Next}(\mathbf{C}, \mathbf{b}_\mathsf{c})$. There should be $\mathbf{b}_1 \neq \mathbf{b}_2$, $\mathbf{b}_1 \neq \perp$ and $\mathbf{b}_2 \neq \perp$. According to claim 4.5.1, $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_1) > 0$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}_2) > 0$. Since $\mathbf{b}_1$ and $\mathbf{b}_2$ have the same parent block, according to lemma 4.4, $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_2) \leq 0$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}_1) \leq 0$. Thus

$$\mathrm{Adv}(\mathcal{S}, \mathbf{b}_1) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_1) < 0 \quad \textit{and} \quad \mathrm{Adv}(\mathcal{S}, \mathbf{b}_2) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_2) > 0.$$

Recalling that $\mathrm{Adv}(\mathcal{S}, \mathbf{b})$ is defined by $\mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}, \mathbf{b}\big) - \mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b})$. We discuss in two cases with $\mathbf{f}_{-1} = \mathbf{f}$ and $\mathbf{f}_{-1} \neq \mathbf{f}$.

Case 1: $\mathbf{f}_{-1} = \mathbf{f}$, claim 4.3 guarantees $\mathbf{f} \notin \mathbf{B}^{\mathsf{min}}$ and $\mathbf{f}_{-1} \notin \mathbf{B}^{\mathsf{min}}_{-1}$ when $\mathbf{f} \neq \perp$. So for any block $\mathbf{b}$

$$\begin{aligned}&\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b})\\ =&\big(\mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}\big) - \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}_{-1}, \mathbf{b}\big)\big) - \big(\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}) - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1}, \mathbf{b}\big)\big).\end{aligned}$$

Case 1.1: e is hGenRls or mRls event.

It will be $\mathbf{B}^{\mathsf{min}}_{-1} = \mathbf{B}^{\mathsf{min}}$ and $\mathbf{B}^{\mathsf{max}}_{-1} \subseteq \mathbf{B}^{\mathsf{max}}$. So for any block $\mathbf{b}$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) \leq 0$. It can not be $\mathrm{Adv}(\mathcal{S}, \mathbf{b}_2) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_2) > 0$.

Case 1.2: e is Arvl event.

It will be $\mathbf{B}^{\mathsf{min}}_{-1} \subseteq \mathbf{B}^{\mathsf{min}}$ and $\mathbf{B}^{\mathsf{max}}_{-1} = \mathbf{B}^{\mathsf{max}}$. So for any block $\mathbf{b}$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) \geq 0$. It can not be $\mathrm{Adv}(\mathcal{S}, \mathbf{b}_1) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_1) < 0$.

Case 1.3: e is a mGen event.

It will be $\mathbf{B}^{\mathsf{min}}_{-1} = \mathbf{B}^{\mathsf{min}}$ and $\mathbf{B}^{\mathsf{max}}_{-1} = \mathbf{B}^{\mathsf{max}}$. So for any block $\mathbf{b}$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) = 0$. It can not be $\mathrm{Adv}(\mathcal{S}, \mathbf{b}_2) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_2) > 0$.

Case 2: $\mathbf{f}_{-1} \neq \mathbf{f}$. There could be three possible sub-cases according to the rule updating $\mathbf{f}$.

Case 2.1: $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} \neq \perp$ and $\mathbf{f} = \perp$.

It will be $\mathbf{B}^{\mathsf{min}}_{-1} \cup \{\mathbf{f}_{-1}\} \subseteq \mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}$ and $\mathbf{B}^{\mathsf{max}}_{-1} \subseteq \mathbf{B}^{\mathsf{max}}$. So for any block $\mathbf{b}$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) \leq 0$. It can not be $\mathrm{Adv}(\mathcal{S}, \mathbf{b}_2) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_2) > 0$.

Case 2.2: $e.\mathsf{type} = \mathsf{Arvl}$, $\mathbf{f}_{-1} = e.\mathsf{block}$ and $\mathbf{f} = \perp$.

$\mathbf{f}_{-1}$ is included into $\mathbf{B}^{\mathsf{min}}$. So $\mathbf{B}^{\mathsf{min}}_{-1} \cup \{\mathbf{f}_{-1}\} = \mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}$. An Arvl event does not update $\mathbf{B}^{\mathsf{max}}$, so $\mathbf{B}^{\mathsf{max}}_{-1} = \mathbf{B}^{\mathsf{max}}$. Thus for any block $\mathbf{b}$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) = 0$. It can not be $\mathrm{Adv}(\mathcal{S}, \mathbf{b}_2) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_2) > 0$.

Case 2.3: $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \perp$ and $\mathbf{f} = e.\mathsf{block}$.

Let $\mathbf{B}_\mathsf{f} = \mathbf{f}.\mathsf{past}$. According to lemma A.7, $\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{f})$, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}') > \eta_\mathsf{w} - s_\mathsf{m} - s_\mathsf{h} \geq s_\mathsf{m} + s_\mathsf{h}$. (Recalling that we require $\eta_\mathsf{w} \geq 2s_\mathsf{m} + 2s_\mathsf{h}$.) According to claim 4.5.3, $\mathrm{Chain}(\mathbf{f})$ is a prefix of $\mathbf{C}$. Since $\mathbf{f} \neq \perp$ and e is the hGenRls event of $\mathbf{f}$, according to lemma A.5, $\mathbf{C}_{-1}$ is a prefix of $\mathrm{Chain}(\mathbf{f})$. Thus $\mathbf{C}_{-1}$ is a prefix of $\mathbf{C}$.

As a summary, the cases except case 2.3 are proved by contradiction. For the case 2.3, we show $\mathbf{C}_{-1}$ is a prefix of $\mathbf{C}$ directly. □

Lemma A.9 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$.*

$\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathrm{Tip}(\mathbf{C})$ *only if one of the following condition holds:*

- $e.\mathsf{type} = \mathsf{Arvl}$ *and* $\mathbf{f}_{-1} = \mathbf{f}$.
- $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \perp$ *and* $\mathbf{f} = e.\mathsf{block}$.

$\mathrm{Tip}(\mathbf{C}) \prec \mathrm{Tip}(\mathbf{C}_{-1})$ *only if one of the following condition holds:*

- $e.\mathsf{type} \in \{\mathsf{hGenRls}, \mathsf{mRls}\}$ *and* $\mathbf{f}_{-1} = \mathbf{f}$.
- $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} \neq \perp$ *and* $\mathbf{f} = \perp$.

Proof. If $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathrm{Tip}(\mathbf{C})$, let $\mathbf{b} := \mathrm{Next}(\mathbf{C}, \mathrm{Tip}(\mathbf{C}_{-1}))$. According to claim 4.5.2, since $\mathbf{b}$ is the child block of $\mathrm{Tip}(\mathbf{C}_{-1})$, $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) \leq s_\mathsf{h} + s_\mathsf{m}$. According to claim 4.5.2, since $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{b}$ and $\mathbf{b} \in \mathbf{C}$, so $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) > s_\mathsf{h} + s_\mathsf{m}$. So $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) > 0$.

In the proof of lemma A.8, there is no block satisfying $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) > 0$ in case 1.1, 1.3, 2.1, 2.2. Thus $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathrm{Tip}(\mathbf{C})$ only if one of the following condition holds.

- $e.\mathsf{type} = \mathsf{Arvl}$ and $\mathbf{f}_{-1} = \mathbf{f}$.
- $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \perp$ and $\mathbf{f} = e.\mathsf{block}$.

If $\mathrm{Tip}(\mathbf{C}) \prec \mathrm{Tip}(\mathbf{C}_{-1})$, let $\mathbf{b} := \mathrm{Next}(\mathbf{C}_{-1}, \mathrm{Tip}(\mathbf{C}))$. Since $\mathbf{b}$ is the child block of $\mathrm{Tip}(\mathbf{C}_{-1})$ and $\mathbf{b} \preceq \mathrm{Tip}(\mathbf{C})$, according to claim 4.5.1, $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) \leq 0$. Since $\mathbf{b} \in \mathbf{C}_{-1}$, according to claim 4.5.2, $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) > 0$. So $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) < 0$.

In the proof of lemma A.8, there is no block satisfying $\mathrm{Adv}(\mathcal{S}, \mathbf{b}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}) < 0$ in case 1.2, 1.3, 2.2. It also shows $\mathbf{C}_{-1}$ is a prefix of $\mathbf{C}$ in case 2.3. Thus $\mathrm{Tip}(\mathbf{C}) \prec \mathrm{Tip}(\mathbf{C}_{-1})$ only if one of the following condition holds.

- $e.\mathsf{type} \in \{\mathsf{hGenRls}, \mathsf{mRls}\}$ and $\mathbf{f}_{-1} = \mathbf{f}$.
- $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} \neq \perp$ and $\mathbf{f} = \perp$.

□

Lemma A.10 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. Let $\mathbf{b}_\mathsf{c} := \mathrm{Tip}(\mathrm{Pivot}(\mathbf{B}_{-1}) \cap \mathbf{C}_{-1})$.*

If $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$ and $\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True}$, then $\mathrm{Adapt}(e.\mathsf{block}) = \mathsf{con}$.

Proof. Let $\mathbf{b} := e.\mathsf{block}$ and $\mathbf{B}_{-1} := \mathbf{b}.\mathsf{past}$. Since $\mathbf{B}_{-1}$ is the local state of the honest node who generates $\mathbf{b}$, according to claim 4.1, $\mathbf{B}_{-1}^{\mathsf{min}} \subseteq \mathbf{B}_{-1} \subseteq \mathbf{B}_{-1}^{\mathsf{max}}$. Since $\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True}$ and $\mathbf{B}_{-1}^{\mathsf{min}} \subseteq \mathbf{B}_{-1}$, it can be verified that

$$\mathrm{Old}(\mathbf{B}_{-1}, \mathbf{b}_\mathsf{c}) = \mathsf{True}.$$

Let $\mathbf{b}_{\mathsf{c}1} = \mathrm{Next}(\mathrm{Pivot}(\mathbf{B}_{-1}), \mathbf{b}_\mathsf{c})$. If $\mathbf{b}_{\mathsf{c}1} = \perp$, $\mathbf{b}_\mathsf{c}$ will be the last block in the pivot chain of $\mathbf{B}_{-1}$. Since we have $\mathrm{Old}(\mathbf{B}_{-1}, \mathbf{b}_\mathsf{c}) = \mathsf{True}$, $\mathrm{Adapt}(\mathbf{b}) = \mathsf{con}$ according to the second rule in definition 3.1.

In the following, we discuss the case $\mathbf{b}_{\mathsf{c}1} \neq \perp$ according to whether $\mathbf{C}_{-1}$ is a prefix of $\mathrm{Pivot}(\mathbf{B}_{-1})$ or not. Since $\mathbf{b}_{\mathsf{c}1} \neq \perp$, it is the child block of $\mathbf{b}_\mathsf{c}$.

Case 1: $\mathbf{C}_{-1}$ is a prefix of (or equals to) $\mathrm{Pivot}(\mathbf{B}_{-1})$.

In this case, $\mathbf{b}_\mathsf{c} = \mathrm{Tip}(\mathbf{C}_{-1})$. Since $\mathbf{b}_{\mathsf{c}1}$ is the child of $\mathbf{b}_\mathsf{c}$, according to claim 4.5.2,

$$\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_{\mathsf{c}1}) \leq s_\mathsf{h} + s_\mathsf{m}.$$

Let $\mathbf{T}_{-1} := \mathrm{SubT}\big(\mathbf{B}_{-1}^{\Delta}, \mathrm{Tip}(\mathbf{C}_{-1})\big)$, according to the definition of special status (definition 4.2), we have $\mathrm{TotalW}(\mathbf{T}_{-1} \cap \mathbf{M}_{-1}) < s_\mathsf{m}$, $|\{\mathbf{b}' \in \mathbf{B}_{-1}^{\Delta} \backslash \mathbf{M}_{-1} \mid \mathbf{b}'.\mathsf{weight} = 1\}| < s_\mathsf{h}$ and $|\{\mathbf{b}' \in \mathbf{B}_{-1}^{\Delta} \backslash \mathbf{M}_{-1} \mid \mathbf{b}'.\mathsf{weight} = \eta_\mathsf{w}\}| \leq 2$. Thus

$$\mathrm{SubT}\big(\mathbf{B}_{-1}^{\Delta}, \mathbf{b}_{\mathsf{c}1}.\mathsf{parent}\big) = \mathrm{TotalW}(\mathbf{T}) < s_\mathsf{m} + s_\mathsf{h} + 2\eta_\mathsf{w}.$$

Recalling that we require $\eta_\mathsf{a} \geq 2s_\mathsf{m} + 2s_\mathsf{h} + 2\eta_\mathsf{w}$. According to lemma A.6,

$$\begin{aligned}
&\mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}1}) - \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}1}) \\
\leq &\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_{\mathsf{c}1}) + \mathrm{SubT}\big(\mathbf{B}_{-1}^{\Delta}, \mathbf{b}_{\mathsf{c}1}.\mathsf{parent}\big) \\
< &2s_\mathsf{m} + 2s_\mathsf{h} + 2\eta_\mathsf{w} \\
\leq &\eta_\mathsf{a}.
\end{aligned}$$

Since we are given $\mathrm{Old}(\mathbf{B}, \mathbf{b}_\mathsf{c}) = \mathsf{True}$, we have $\mathrm{Adapt}(\mathbf{b}) = \mathsf{con}$ according to the first rule in definition 3.1.

Case 2: $\mathbf{C}_{-1}$ is not a prefix of (or equals to) $\mathrm{Pivot}(\mathbf{B}_{-1})$.

In this case, we claim that $\mathbf{b}_\mathsf{c}$ is not the last block in $\mathbf{C}_{-1}$. Let $\mathbf{b}_{\mathsf{c}2} := \mathrm{Next}(\mathbf{C}_{-1}, \mathbf{b}_\mathsf{c})$. According to claim 4.5.1, $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{b}_{\mathsf{c}2}) > 0$. Since $\mathbf{b}_{\mathsf{c}1}$ and $\mathbf{b}_{\mathsf{c}2}$ are in sibling relations and $\mathrm{SibSubTW}(\cdot)$ returns the maximum subtree weight among sibling blocks (defined in eq. (6)), we have $\mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}1}) \leq \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}2})$ and $\mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}2}) \leq \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}1})$. (For the special case $\mathbf{b}_{\mathsf{c}2} \notin \mathbf{B}_{-1}$, these two inequalities also hold.) So we have

$$\begin{aligned}
&\mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}1}) - \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}1}) \\
\leq &\mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}2}) - \mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}_{\mathsf{c}2}) \\
\leq &\mathrm{SibSubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{b}_{\mathsf{c}2}\big) - \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{min}} \cup \{\mathbf{f}_{-1}\}, \mathbf{b}_{\mathsf{c}2}\big) + \eta_\mathsf{w} \\
< &\eta_\mathsf{w} \\
\leq &\eta_\mathsf{a}.
\end{aligned}$$

Since we are given $\mathrm{Old}(\mathbf{B}, \mathbf{b}_\mathsf{c}) = \mathsf{True}$, we have $\mathrm{Adapt}(\mathbf{b}) = \mathsf{con}$ according to the first rule in definition 3.1. □

A.3 Case discussions for potential value (Part 1)

Common settings For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. In this sub-section, we study the upper bound of block potential value difference $P(\mathcal{S}, \mathbf{b}) - P(\mathcal{S}_{-1}, \mathbf{b})$ under the assumption that $P(\mathcal{S}_{-1}, \mathbf{b}) \neq \perp$ and $P(\mathcal{S}, \mathbf{b}) \neq \perp$. All the lemmas in this section assumes $P(\mathcal{S}_{-1}, \mathbf{b}) \neq \perp$ and $P(\mathcal{S}, \mathbf{b}) \neq \perp$. We will also not repeat them in the following lemmas (except lemma A.23) since they are not referred outside this section. The upper bound of

We define symbol $\mathbf{c}_{-1} := \mathrm{Next}(\mathbf{C}_{-1}, \mathbf{b})$ and $\mathbf{c} := \mathrm{Next}(\mathbf{C}, \mathbf{b})$ for given $\mathbf{C}_{-1}$, $\mathbf{C}$ and $\mathbf{b}$ in the context. So $\mathbf{c}$ and $\mathbf{c}_{-1}$ are the child of block $\mathbf{b}$ when they are not $\perp$. According to lemma A.8, one of $\mathbf{C}_{-1}$ and $\mathbf{C}$ must be the prefix of another. So if $\mathbf{c}_{-1} \neq \perp$ and $\mathbf{c} \neq \perp$, there must be $\mathbf{c}_{-1} = \mathbf{c}$. These property are frequently used and we do not explicitly refer them in the following. We define symbol $\mathbf{N} := \{\mathbf{b}' \in \mathbf{B}^{\mathsf{max}} \backslash \mathbf{M} \mid \mathbf{b}'.\mathsf{weight} = 1\}$ for given $\mathbf{B}^{\mathsf{max}}$ and $\mathbf{M}$ in the context and we define $\mathbf{N}_{-1}$ similarly.

Now we start the case discussion for three components $P_{\mathsf{with}}, P_{\mathsf{adv}}, P_{\mathsf{sp}}$.

A.3.1 The first component

Lemma A.11 *If e is* mGen *event, then $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}) \leq e.\mathsf{block}.\mathsf{weight}$.*
If e is mRls *event and $\mathbf{b} \preceq e.\mathsf{block}$, then $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}) \leq -e.\mathsf{block}.\mathsf{weight}$.*
For other cases, $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}) = 0$.

Proof. Let $\mathbf{b}_\mathsf{e} = e.\mathsf{block}$. Recalling that $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) = \mathrm{SubTW}(\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}}, \mathbf{b})$.

If e is mGen event, we have $\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{gen}}_{-1} = \{\mathbf{b}_\mathsf{e}\}$ and $\mathbf{B}^{\mathsf{max}} = \mathbf{B}^{\mathsf{max}}_{-1}$. So $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}) \leq \mathbf{b}_\mathsf{e}.\mathsf{weight}$.

If e is mRls event, we have $\mathbf{B}^{\mathsf{gen}} = \mathbf{B}^{\mathsf{gen}}_{-1}$, $\mathbf{B}^{\mathsf{max}} \backslash \mathbf{B}^{\mathsf{max}}_{-1} = \{\mathbf{b}_\mathsf{e}\}$ and $\mathbf{b}_\mathsf{e} \in \mathbf{B}^{\mathsf{gen}}$. So $\mathbf{B}^{\mathsf{gen}}_{-1} \backslash \mathbf{B}^{\mathsf{max}}_{-1}$ has one more element $\mathbf{b}_\mathsf{e}$ than $\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}}$. When $\mathbf{b}_\mathsf{e} \preceq \mathbf{b}$, $\mathbf{b}_\mathsf{e} \in \mathrm{SubT}\big(\mathbf{B}^{\mathsf{gen}}_{-1} \backslash \mathbf{B}^{\mathsf{max}}_{-1}, \mathbf{b}\big)$ and thus $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}) \leq -\mathbf{b}_\mathsf{e}.\mathsf{weight}$. Otherwise $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}) = 0$.

If e is hGenRls event, we have $\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{gen}}_{-1} = \{\mathbf{b}_\mathsf{e}\}$ and $\mathbf{B}^{\mathsf{max}} \backslash \mathbf{B}^{\mathsf{max}}_{-1} = \{\mathbf{b}_\mathsf{e}\}$. Thus $\mathbf{B}^{\mathsf{gen}}_{-1} \backslash \mathbf{B}^{\mathsf{max}}_{-1} = \mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}}$ and $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}) = 0$.

If e is Arvl event, we have $\mathbf{B}^{\mathsf{gen}} = \mathbf{B}^{\mathsf{gen}}_{-1}$ and $\mathbf{B}^{\mathsf{max}} = \mathbf{B}^{\mathsf{max}}_{-1}$. Thus $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}) = 0$. □

A.3.2 The second component

In discussing the component P_{adv}, for the case $\mathbf{f} = \mathbf{f}_{-1}$, we discuss the upper bound of $P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1})$ and $P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1})$ respectively and combine them later.

Lemma A.12 *If e is* hGenRls *event, $\mathbf{f}_{-1} = \mathbf{f}$, $\mathsf{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$ and $e.\mathsf{block}.\mathsf{weight} = 1$, then it will be*

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \leq -1.$$

Proof. Let $\mathbf{b}_\mathsf{e} = e.\mathsf{block}$. If $\mathbf{c}_{-1} = \perp$, we have $\mathbf{b} = \mathrm{Tip}(\mathbf{C}_{-1})$. According to lemma A.5, $\mathbf{C}_{-1}$ is a prefix of $\mathbf{b}_\mathsf{e}$. Thus $\mathbf{b} = \mathrm{Tip}(\mathbf{C}_{-1} \cap \mathrm{Chain}(\mathbf{b}_\mathsf{e}))$. Since $P(\mathcal{S}_{-1}, \mathbf{b}) \neq \perp$, we have $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_{-1}, \mathbf{b}) = \mathsf{True}$. According to lemma A.10, $\mathrm{Adapt}(\mathbf{b}_\mathsf{e}) = \mathsf{con}$. So $\mathbf{b}_\mathsf{e}.\mathsf{weight} \neq 1$ according to equation 9. This contradicts the assumption $\mathbf{b}_\mathsf{e}.\mathsf{weight} = 1$ in this lemma. Thus, $\mathbf{c}_{-1}$ cannot be $\perp$.

Since $e.\mathsf{type} = \mathsf{hGenRls}$ and $\mathbf{b}_\mathsf{e}.\mathsf{weight} = 1$, $\mathbf{N}$ must has one more element $\mathbf{b}_\mathsf{e}$ compared to $\mathbf{N}_{-1}$. Since $\mathbf{c}_{-1} \in \mathbf{C}_{-1}$ and $\mathbf{C}_{-1}$ is a prefix of $\mathbf{b}_\mathsf{e}$, it will be $\mathbf{c}_{-1} \preceq \mathbf{b}_\mathsf{e}$. Thus $\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}_{-1}\big)$ has one more element $\mathbf{b}_\mathsf{e}$ compared to $\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{c}_{-1}\big)$. Since $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$, according to the definition of special status, $\mathbf{B}^{\Delta}_{-1}$ has at most $s_\mathsf{h} - 1$ honest blocks with block weight 1. Thus $|\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{c}_{-1}\big)| \leq s_\mathsf{h} - 1$. So we have

$$\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}_{-1}\big) \cap \mathbf{N}) = \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{c}_{-1}\big) \cap \mathbf{N}_{-1}) + 1 \leq s_\mathsf{h}.$$

Since $\mathbf{c}_{-1} \preceq \mathbf{b}_\mathsf{e}$ and $\mathbf{B}^{\mathsf{max}}$ has one more element $\mathbf{b}_\mathsf{e}$ than $\mathbf{B}^{\mathsf{max}}_{-1}$, all the sibling blocks of $\mathbf{c}_{-1}$ has the same subtree in $\mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}^{\mathsf{max}}_{-1}$. So we have $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}_\mathsf{e}) = \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1}, \mathbf{b}_\mathsf{e}\big)$. We also have $\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\} = \mathbf{B}^{\mathsf{min}}_{-1} \cup \{\mathbf{f}_{-1}\}$ because $e.\mathsf{type} \neq \mathsf{Arvl}$ and $\mathbf{f} = \mathbf{f}_{-1}$. Thus

$$\mathrm{Adv}(\mathcal{S}, \mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) = 0.$$

Recalling that $P_{\mathsf{adv}}(\mathcal{S},\mathbf{c}) = s_{\mathsf{h}} + s_{\mathsf{m}} - \mathrm{Adv}(\mathcal{S},\mathbf{c}) - \min\{s_{\mathsf{h}}, \mathrm{TotalW}(\mathrm{SubT}(\mathbf{B}^{\Delta},\mathbf{c}) \cap \mathbf{N})\}$. Thus we have

$$\mathrm{Adv}(\mathcal{S},\mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S},\mathbf{c}_{-1}) = -1.$$

□

Lemma A.13 *If e is a* mRls *event, $\mathbf{f} = \mathbf{f}_{-1}$, $\mathbf{c}_{-1} \neq \perp$ and $e.\mathsf{block} \in \mathrm{SubT}(\mathbf{B}^{\mathsf{max}},\mathbf{b}) \backslash \mathrm{SubT}(\mathbf{B}^{\mathsf{max}},\mathbf{c}_{-1})$. We have*

$$P_{\mathsf{adv}}(\mathcal{S},\mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1},\mathbf{c}_{-1}) \leq e.\mathsf{block.weight}.$$

For all the other cases satisfying $\mathbf{f} = \mathbf{f}_{-1}$,

$$P_{\mathsf{adv}}(\mathcal{S},\mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1},\mathbf{c}_{-1}) \leq 0.$$

Proof. Let $\mathbf{b}_{\mathsf{e}} := e.\mathsf{block}$. In this proof, we try to find all the cases with $P_{\mathsf{adv}}(\mathcal{S},\mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1},\mathbf{c}_{-1}) > 0$. When $\mathbf{c}_{-1} = \perp$, both $P_{\mathsf{adv}}(\mathcal{S},\mathbf{c}_{-1})$ and $P_{\mathsf{adv}}(\mathcal{S}_{-1},\mathbf{c}_{-1})$ equal to 0. So we only focus on the case with $\mathbf{c}_{-1} \neq \perp$.

We study the difference between $\mathrm{SubT}(\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}) \cap \mathbf{N}_{-1}$ and $\mathrm{SubT}(\mathbf{B}^{\Delta},\mathbf{c}_{-1}) \cap \mathbf{N}$. Since e does not a hGenRls event, we have $\mathbf{N} = \mathbf{N}_{-1}$. Thus

$$\begin{aligned}
&\min\{\mathrm{TotalW}(\mathrm{SubT}(\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}) \cap \mathbf{N}_{-1}), s_{\mathsf{h}}\} - \min\{\mathrm{TotalW}(\mathrm{SubT}(\mathbf{B}^{\Delta},\mathbf{c}_{-1}) \cap \mathbf{N}), s_{\mathsf{h}}\} \\
\leq &\max\{0, \mathrm{TotalW}(\mathrm{SubT}(\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}) \cap \mathbf{N}) - \mathrm{TotalW}(\mathrm{SubT}(\mathbf{B}^{\Delta},\mathbf{c}_{-1}) \cap \mathbf{N})\} \\
\leq &\mathrm{SubTW}\big((\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{B}^{\Delta}) \cap \mathbf{N}, \mathbf{c}_{-1}\big)
\end{aligned}$$

By the definition of P_{adv}, we have

$$P_{\mathsf{adv}}(\mathcal{S},\mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1},\mathbf{c}_{-1}) \leq \mathrm{Adv}(\mathcal{S}_{-1},\mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S},\mathbf{c}_{-1}) + \mathrm{SubTW}\big((\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{B}^{\Delta}) \cap \mathbf{N}, \mathbf{c}_{-1}\big).$$

Notice that $\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{B}^{\Delta} = (\mathbf{B}^{\mathsf{max}}_{-1} \backslash \mathbf{B}^{\mathsf{min}}_{-1}) \backslash (\mathbf{B}^{\mathsf{max}} \backslash \mathbf{B}^{\mathsf{min}})$. If $\mathbf{B}^{\mathsf{min}} = \mathbf{B}^{\mathsf{min}}_{-1}$, then $\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{B}^{\Delta} \subseteq (\mathbf{B}^{\mathsf{max}}_{-1} \backslash \mathbf{B}^{\mathsf{max}}) = \emptyset$. So $\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{B}^{\Delta} \neq \emptyset$ only if $\mathbf{B}^{\mathsf{min}} \neq \mathbf{B}^{\mathsf{min}}_{-1}$ and thus e must be an ArvI event. We can claim $\mathrm{SubTW}\big((\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{B}^{\Delta}) \cap \mathbf{N}, \mathbf{c}_{-1}\big) > 0$ only if e is an ArvI event and $\mathbf{c}_{-1} \preceq \mathbf{b}_{\mathsf{e}}$. In this case, $\mathbf{B}^{\mathsf{max}}_{-1} = \mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}^{\Delta} \subseteq \mathbf{B}^{\Delta}_{-1}$, so we have

$$\begin{aligned}
&\text{If } \mathrm{SubTW}\big((\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{B}^{\Delta}) \cap \mathbf{H}, \mathbf{c}_{-1}\big) > 0, \\
&\quad \mathrm{Adv}(\mathcal{S}_{-1},\mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S},\mathbf{c}_{-1}) + \mathrm{SubTW}\big((\mathbf{B}^{\Delta}_{-1} \backslash \mathbf{B}^{\Delta}) \cap \mathbf{H}, \mathbf{c}_{-1}\big) \\
\leq &\mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}_{-1}, \mathbf{c}_{-1}\big) - \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{c}_{-1}\big) + \mathrm{SubTW}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{c}_{-1}\big) - \mathrm{SubTW}\big(\mathbf{B}^{\Delta}, \mathbf{c}_{-1}\big) \\
= &\mathrm{SubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1}, \mathbf{c}_{-1}\big) - \mathrm{SubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}_{-1}) \\
= &0
\end{aligned}$$

Thus we have

$$P_{\mathsf{adv}}(\mathcal{S},\mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1},\mathbf{c}_{-1}) \leq \max\{0, \mathrm{Adv}(\mathcal{S}_{-1},\mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S},\mathbf{c}_{-1})\}.$$

Now we only need to study in which cases $\mathrm{Adv}(\mathcal{S}_{-1},\mathbf{c}_{-1}) > \mathrm{Adv}(\mathcal{S},\mathbf{c}_{-1})$. It can only happens when $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}},\mathbf{c}_{-1}) > \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1},\mathbf{c}_{-1}\big)$. Thus it must be

$$e.\mathsf{type} \in \{\mathsf{hGenRls}, \mathsf{mRls}\} \quad \text{and} \quad \mathbf{b}_{\mathsf{e}} \in \mathrm{SubT}(\mathbf{B}^{\mathsf{max}},\mathbf{b}) \backslash \mathrm{SubT}(\mathbf{B}^{\mathsf{max}},\mathbf{c}_{-1}).$$

If $e.\mathsf{type} = \mathsf{mRls}$, since $\mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}^{\mathsf{max}}_{-1}$ differ at one block $\mathbf{b}_{\mathsf{e}}$, $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}},\mathbf{c}_{-1}) - \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1},\mathbf{c}_{-1}\big) \leq \mathbf{b}_{\mathsf{e}}.\mathsf{weight}$ and thus

$$\mathrm{Adv}(\mathcal{S}_{-1},\mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S},\mathbf{c}_{-1}) \leq \mathbf{b}_{\mathsf{e}}.\mathsf{weight}.$$

If $e.\mathsf{type} = \mathsf{hGenRls}$, recalling that block $\mathbf{b}$ is the parent of $\mathbf{c}_{-1}$ and $\mathbf{c}_{-1} \in \mathbf{C}_{-1}$, we have $\mathrm{Tip}(\mathrm{Chain}(\mathbf{b}_{\mathsf{e}}) \cap \mathbf{C}_{-1}) = \mathbf{b}$. Since $P(\mathcal{S}_{-1},\mathbf{b}) \neq \perp$, we have $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_{-1},\mathbf{b}) = \mathsf{True}$. According to lemma A.10, $\mathrm{Adapt}(\mathbf{b}_{\mathsf{e}}) = \mathsf{con}$. So $\mathbf{b}_{\mathsf{e}}.\mathsf{weight}$ equals to 0 or η_{w}. We have three sub-cases as follows.

1. If $\mathbf{b}_{\mathsf{e}}.\mathsf{weight} = 0$, then $\mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}^{\mathsf{max}}_{-1}$ differ at one block with zero block weight. So $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}},\mathbf{c}_{-1}) = \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1},\mathbf{c}_{-1}\big)$.

2. If $\mathbf{b}_\mathsf{e}.\mathsf{weight} = \eta_\mathsf{w}$ and $\mathbf{f} \neq \bot$, since $\mathbf{b}_\mathsf{e}$ is an honest block and $e.\mathsf{type} = \mathsf{hGenRls}$, according to the rule in updating flag block, we have $\mathbf{f} = \bot \neq \mathbf{f}$. This contradicts to our assumption.
3. If $\mathbf{f} = \bot$, since $\mathbf{b}_\mathsf{e}$ is an honest block and $e.\mathsf{type} = \mathsf{hGenRls}$, according to lemma A.5, $\mathbf{C}_{-1}$ should be a prefix of $\mathrm{Chain}(\mathbf{b}_\mathsf{e})$ and thus $\mathbf{c}_{-1} \in \mathrm{Chain}(\mathbf{b}_\mathsf{e})$. This contradicts $\mathbf{b}_\mathsf{e} \notin \mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}_{-1})$.

In all, if $e.\mathsf{type} = \mathsf{hGenRls}$, all the three sub-cases cannot be $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) > \mathrm{Adv}(\mathcal{S}, \mathbf{c}_{-1})$.

As a summary for the whole proof, for the case $\mathbf{f} = \mathbf{f}_{-1}$, $P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) > 0$ only if $e.\mathsf{type} = \mathsf{mRls}$, $\mathbf{c}_{-1} \neq \bot$ and $e.\mathsf{block} \in \mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}) \backslash \mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}_{-1})$. For this case, we have

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \leq e.\mathsf{block.weight}.$$

□

Lemma A.14 *If $\mathbf{c}_{-1} \neq \bot$ and $\mathbf{c} = \bot$, then*

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) \leq -s_\mathsf{m}.$$

For the other cases,

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) \leq 0.$$

Proof. If $\mathbf{c} = \mathbf{c}_{-1}$, then $P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) = 0$ holds trivially.

If $\mathbf{c} \neq \mathbf{c}_{-1}$, then one of $\mathbf{c}$ and $\mathbf{c}_{-1}$ equals to $\bot$. (Recalling that $\mathbf{c} = \mathbf{c}_{-1}$ if $\mathbf{c}_{-1} \neq \bot$ and $\mathbf{c} \neq \bot$.)

Case 1: $\mathbf{c} = \bot$ and $\mathbf{c}_{-1} \neq \bot$.

Since $\mathbf{c} = \bot$, $\mathbf{b}$ should be $\mathrm{Tip}(\mathbf{C})$. Since block $\mathbf{c}_{-1}$ is the child block of $\mathbf{b}$ and $\mathbf{c}_{-1} \preceq \mathrm{Tip}(\mathbf{C}_{-1})$, according to claim 4.5.2, $\mathrm{Adv}(\mathcal{S}, \mathbf{c}_{-1}) \leq 0$. Since $\min\{\mathrm{TotalW}(\mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}_{-1}) \cap \mathbf{N}), s_\mathsf{h}\} \leq s_\mathsf{h}$ and $s_\mathsf{h} \geq 0$, we have

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) = -P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) \leq \mathrm{Adv}(\mathcal{S}, \mathbf{c}_{-1}) + s_\mathsf{h} - s_\mathsf{h} - s_\mathsf{m} \leq -s_\mathsf{m}.$$

Case 2: $\mathbf{c} \neq \bot$ and $\mathbf{c}_{-1} = \bot$.

Since $\mathbf{c}_{-1} = \bot$, $\mathbf{b}$ should be $\mathrm{Tip}(\mathbf{C}_{-1})$. Since block $\mathbf{c}$ is the child block of $\mathbf{b}$, we have $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{c}$, according to claim 4.5.1, $\mathrm{Adv}(\mathcal{S}, \mathbf{c}) > s_\mathsf{h} + s_\mathsf{m}$. Since $\min\{\mathrm{TotalW}(\mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}) \cap \mathbf{N}), s_\mathsf{h}\} \geq 0$, we have

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) = P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) \leq s_\mathsf{h} + s_\mathsf{m} - \mathrm{Adv}(\mathcal{S}, \mathbf{c}) < 0.$$

□

Lemma A.15 *If $e.\mathsf{type} = \mathsf{hGenRls}$, $e.\mathsf{block.weight} = \eta_\mathsf{w}$, $\mathbf{f}_{-1} = \bot$ and $\mathbf{f} = e.\mathsf{block}$, then*

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \leq 2s_\mathsf{h} + 2s_\mathsf{m} - \eta_\mathsf{w}.$$

Proof. Let $\mathbf{b}_\mathsf{e} := e.\mathsf{block}$. If $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \bot$ and $\mathbf{f} = e.\mathsf{block}$, according to lemma A.7, for all the block $\mathbf{b}'$ in $\mathrm{Chain}(\mathbf{b}_\mathsf{e})$, we have $\mathrm{Adv}(\mathcal{S}, \mathbf{b}') > \eta_\mathsf{w} - s_\mathsf{h} - s_\mathsf{m} \geq s_\mathsf{h} + s_\mathsf{m}$. According to claim 4.5.3, $\mathrm{Chain}(\mathbf{b}_\mathsf{e})$ is a prefix of $\mathbf{C}$. Since $\mathbf{b}_\mathsf{e} \notin \mathbf{B}^{\mathsf{min}}$, $\mathbf{b}_\mathsf{e}$ cannot be an old enough block and thus $P(\mathcal{S}, \mathbf{b}_\mathsf{e}) = \bot$. So $\mathbf{b}$ cannot be the last block in $\mathbf{C}$. We claim

$$\mathbf{c} \neq \bot.$$

Case 1: $\mathbf{c}_{-1} \neq \bot$. It will be $\mathbf{c} = \mathbf{c}_{-1}$

Recalling that $\mathbf{c} \in \mathrm{Chain}(\mathbf{b}_\mathsf{e})$, so the subtree weight for the sibling blocks of $\mathbf{c}$ does not change. We have $\mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}) = \mathrm{SibSubTW}\big(\mathbf{B}^{\mathsf{max}}_{-1}, \mathbf{c}\big)$. Since $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \bot$ and $\mathbf{f} = e.\mathsf{block}$, $\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}$ has one more block $\mathbf{b}_\mathsf{e}$ than $\mathbf{B}^{\mathsf{min}}_{-1} \cup \{\mathbf{f}_{-1}\}$. So $\mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}_{-1} \cup \{\mathbf{f}_{-1}\}, \mathbf{c}\big) - \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}, \mathbf{c}\big) = -\mathbf{b}_\mathsf{e}.\mathsf{weight}$. Since $\mathbf{b}_\mathsf{e} = \mathbf{f}$, the block weight of $\mathbf{b}_\mathsf{e}$ must be η_w. Thus

$$\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}) - \mathrm{Adv}(\mathcal{S}, \mathbf{c}_{-1}) = -\eta_\mathsf{w}.$$

Since $\mathbf{b}_\mathsf{e}.\mathsf{weight} = \eta_\mathsf{w}$ and $\mathbf{N}$ only contains honest blocks with block weight 1, we have $\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{c}\big) \cap \mathbf{N}_{-1} = \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{N}$. So in this case,

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \leq \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}) - \mathrm{Adv}(\mathcal{S}, \mathbf{c}) + s_\mathsf{h} = s_\mathsf{h} - \eta_\mathsf{w}.$$

Case 2: $\mathbf{c}_{-1} = \bot$.

Recalling that $\mathbf{c} \in \mathrm{Chain}(\mathbf{b}_\mathsf{e})$ and so $\mathrm{Adv}(\mathcal{S}, \mathbf{c}) > \eta_\mathsf{w} - s_\mathsf{h} - s_\mathsf{m}$ according to lemma A.7. Thus

$$P_\mathsf{adv}(\mathcal{S}, \mathbf{c}) - P_\mathsf{adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \le s_\mathsf{h} + s_\mathsf{m} - \mathrm{Adv}(\mathcal{S}, \mathbf{c}) < 2s_\mathsf{h} + 2s_\mathsf{m} - \eta_\mathsf{w}.$$

□

Lemma A.16 *If* $e.\mathsf{type} = \mathsf{hGenRls}$, $e.\mathsf{block.weight} = \eta_\mathsf{w}$, $\mathbf{f}_{-1} \neq \bot$ *and* $\mathbf{f} = \bot$, *we have*

$$P_\mathsf{adv}(\mathcal{S}, \mathbf{c}) - P_\mathsf{adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \le \eta_\mathsf{w}.$$

Proof. Let $\mathbf{b}_\mathsf{e} := e.\mathsf{block}$ and $\mathbf{B}_{-1} := \mathbf{b}_\mathsf{e}.\mathsf{past}$. Since $\mathbf{b}_\mathsf{e}$ is an honest block, according to claim 4.1, $\mathbf{B}_{-1}^{\mathsf{min}} \subseteq \mathbf{B}_{-1} \subseteq \mathbf{B}_{-1}^{\mathsf{max}}$. Since $e.\mathsf{block.weight} = \eta_\mathsf{w}$, $\mathbf{f}_{-1} \neq \bot$ and $\mathbf{f} = \bot$, according to lemma A.9, it cannot be $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathrm{Tip}(\mathbf{C})$. Thus it cannot be $\mathbf{c}_{-1} = \bot \wedge \mathbf{c} \neq \bot$.

Case 1: $\mathbf{c}_{-1} \neq \bot$ and $\mathbf{c} \neq \bot$. It will be $\mathbf{c} = \mathbf{c}_{-1}$.

Since $\mathbf{b}_\mathsf{e}.\mathsf{weight} = \eta_\mathsf{w}$ and $\mathbf{N}$ only contains honest blocks with block weight 1, we have $\mathrm{SubT}\big(\mathbf{B}_{-1}^{\Delta}, \mathbf{c}\big) \cap \mathbf{N}_{-1} = \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{N}$. So in this case,

$$P_\mathsf{adv}(\mathcal{S}, \mathbf{c}) - P_\mathsf{adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \le \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S}, \mathbf{c}).$$

Since $\mathbf{c} \in \mathbf{C}$, according to claim 4.5.1, $\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{c}), \mathrm{Adv}(\mathcal{S}, \mathbf{c}) > 0$. Since $\mathbf{f} = \bot$, we have $\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\} = \mathbf{B}^{\mathsf{min}}$. Since $e.\mathsf{type} = \mathsf{hGenRls}$, we have $\mathbf{B}^{\mathsf{min}} = \mathbf{B}_{-1}^{\mathsf{min}}$. Recalling that $\mathbf{B}_{-1}^{\mathsf{min}} \subseteq \mathbf{B}_{-1} \subseteq \mathbf{B}_{-1}^{\mathsf{max}}$, we have

$$\begin{aligned}
\forall \mathbf{b}' \in \mathrm{Chain}(\mathbf{c}), \quad & \mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{b}') - \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{b}') \\
& \ge \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}'\big) - \mathrm{SibSubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{b}'\big) \\
& \ge \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}}, \mathbf{b}'\big) - \mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{b}') \\
& > 0
\end{aligned}$$

According to lemma A.1, $\mathbf{c} \in \mathrm{Pivot}(\mathbf{B}_{-1})$. Thus $\mathbf{c} \prec \mathbf{b}_\mathsf{e}$. Since $\mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}_{-1}^{\mathsf{max}}$ only differs at block $\mathbf{b}_\mathsf{e}$, we have $\mathrm{SibSubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{c}\big) = \mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{c})$. Recalling that $\mathbf{B}_{-1}^{\mathsf{min}} = \mathbf{B}^{\mathsf{min}}$ and $\mathbf{f} = \bot$, we have

$$\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S}, \mathbf{c}) = \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{min}} \cup \{\mathbf{f}_{-1}\}, \mathbf{c}\big) - \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}, \mathbf{c}\big) \le \eta_\mathsf{w}.$$

Combined with the first inequality in this proof, we have proved $P_\mathsf{adv}(\mathcal{S}, \mathbf{c}) - P_\mathsf{adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \le \eta_\mathsf{w}$ for this case.

Case 2: $\mathbf{c}_{-1} \neq \bot$ and $\mathbf{c} = \bot$.

We prove this case by showing that $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \le \eta_\mathsf{w}$. If $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) > \eta_\mathsf{w}$, recalling that $\mathbf{B}_{-1}^{\mathsf{min}} \subseteq \mathbf{B}_{-1} \subseteq \mathbf{B}_{-1}^{\mathsf{max}}$, we have

$$\begin{aligned}
& \mathrm{SubTW}(\mathbf{B}_{-1}, \mathbf{c}_{-1}) - \mathrm{SibSubTW}(\mathbf{B}_{-1}, \mathbf{c}_{-1}) \\
\ge & \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{c}_{-1}\big) - \mathrm{SibSubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{c}_{-1}\big) \\
\ge & \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{min}} \cup \{\mathbf{f}\}, \mathbf{c}_{-1}\big) - \eta_\mathsf{w} - \mathrm{SibSubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{c}_{-1}\big) \\
> & 0.
\end{aligned}$$

So $\mathbf{c}_{-1}$ is the child with maximum subtree weight of $\mathbf{b}$. If $\mathbf{b} \in \mathrm{Pivot}(\mathbf{B}_{-1})$, there must be $\mathbf{c}_{-1} \in \mathrm{Pivot}(\mathbf{B}_{-1})$. Thus no matter $\mathbf{b}$ belongs to $\mathrm{Pivot}(\mathbf{B}_{-1})$ or not, the sibling blocks of $\mathbf{c}_{-1}$ are not in $\mathrm{Pivot}(\mathbf{B}_{-1})$ and $\mathrm{Chain}(\mathbf{b}_\mathsf{e})$. So we have $\mathrm{SibSubTW}\big(\mathbf{B}_{-1}^{\mathsf{max}}, \mathbf{c}_{-1}\big) = \mathrm{SibSubTW}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}_{-1})$. ($\mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}_{-1}^{\mathsf{max}}$ only differs at block $\mathbf{b}_\mathsf{e}$.) Recalling that $\mathbf{B}^{\mathsf{min}} = \mathbf{B}_{-1}^{\mathsf{min}}$ and $\mathbf{f} = \bot$, we have

$$\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) - \mathrm{Adv}(\mathcal{S}, \mathbf{c}) = \mathrm{SubTW}\big(\mathbf{B}_{-1}^{\mathsf{min}} \cup \{\mathbf{f}_{-1}\}, \mathbf{c}\big) - \mathrm{SubTW}\big(\mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}, \mathbf{c}\big) \le \eta_\mathsf{w}.$$

Thus $\mathrm{Adv}(\mathcal{S}, \mathbf{c}) \ge \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) - \eta_\mathsf{w} > 0$. Since $P(\mathcal{S}, \mathbf{b}) \neq \bot$, we have $\mathbf{b} \in \mathbf{C}$. Since $\mathbf{c}_{-1} \preceq \mathrm{Tip}(\mathbf{C}_{-1})$, $\mathrm{Adv}(\mathcal{S}, \mathbf{c}_{-1}) > 0$ and $\mathbf{c}_{-1}.\mathsf{parent} = \mathbf{b}$, $\mathbf{c}_{-1}$ should also belong to $\mathbf{C}$ according to our rule in maintaining chain $\mathbf{C}$. This contradicts to $\mathbf{c} = \bot$.

So in this case, there must be $\mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \leq \eta_{\mathsf{w}}$. And thus

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) \leq -s_{\mathsf{h}} - s_{\mathsf{m}} + \mathrm{Adv}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) + s_{\mathsf{h}} \leq \eta_{\mathsf{w}} - s_{\mathsf{m}}.$$

Case 3: $\mathbf{c}_{-1} = \bot$ and $\mathbf{c} = \bot$.
In this case, $P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}) = 0 < \eta_{\mathsf{w}}$. □

Lemma A.17 *If* $e.\mathsf{type} = \mathsf{Arvl}$, $e.\mathsf{block.weight} = \eta_{\mathsf{w}}$, $\mathbf{f}_{-1} = e.\mathsf{block}$ *and* $\mathbf{f} = \bot$, *we have*

$$P(\mathcal{S}, \mathbf{c}) - P(\mathcal{S}_{-1}, \mathbf{c}_{-1}) = 0.$$

Proof. Let $\mathbf{b}_{\mathsf{e}} := e.\mathsf{block}$. According to lemma A.9, we have $\mathbf{C}_{-1} = \mathbf{C}$ if $e.\mathsf{type} = \mathsf{Arvl}$, $\mathbf{f}_{-1} = e.\mathsf{block}$ and $\mathbf{f} = \bot$. So

$$\mathbf{c}_{-1} = \mathbf{c}.$$

Since e is the Arvl event of $\mathbf{f}_{-1}$ and $\mathbf{f} = \bot$, we have $\mathbf{B}^{\mathsf{min}}_{-1} \cup \{\mathbf{f}_{-1}\} = \mathbf{B}^{\mathsf{min}} \cup \{\mathbf{f}\}$. Since $\mathbf{b}_{\mathsf{e}}.\mathsf{weight} = \eta_{\mathsf{w}}$ and $\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{N}$ only contains honest blocks with block weight 1, we have $\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{c}_{-1}\big) \cap \mathbf{N}_{-1} = \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{N}$. So in this case,

$$P(\mathcal{S}, \mathbf{c}) - P(\mathcal{S}_{-1}, \mathbf{c}_{-1}) = 0.$$

□

A.3.3 The third component

Recalling that the third component of potential value is defined as follows. Let $\mathbf{c} = \mathrm{Next}(\mathbf{C}, \mathbf{b})$,

$$P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) := \begin{cases} \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M} \backslash \mathbf{S}) & \mathbf{c} \neq \bot \\ 0 & \mathbf{c} = \bot \end{cases}$$

Here we define another intermediate potential value for the third component. Let $\mathbf{c}_{-1} = \mathrm{Next}(\mathbf{C}_{-1}, \mathbf{b})$,

$$P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) := \begin{cases} \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}_{-1}\big) \cap \mathbf{M} \backslash \mathbf{S}_{-1}) & \mathbf{c}_{-1} \neq \bot \\ \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{b}\big) \cap \mathbf{M} \backslash \mathbf{S}_{-1}) & \mathbf{c}_{-1} = \bot \end{cases}$$

Lemma A.18 *If* $e.\mathsf{type} = \mathsf{mGen}$ *and one of the following properties hold,*

- $\mathbf{c}_{-1} \neq \bot$ *and* $\mathbf{c}_{-1} \preceq e.\mathsf{block}$
- $\mathbf{c}_{-1} = \bot$ *and* $\mathbf{b} \preceq e.\mathsf{block}$

then we have

$$P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathbf{b}) \leq e.\mathsf{block.weight}.$$

For all the other cases,

$$P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathbf{b}) \leq 0.$$

Proof. Let $\mathbf{b}_{\mathsf{e}} = e.\mathsf{block}$. In this proof, we try to figure out the cases with $P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathbf{b}) > 0$.

Case 1: $\mathbf{c}_{-1} = \bot$.

In this case, we have $\mathbf{b} = \mathrm{Tip}(\mathbf{C}_{-1})$. According to claim 4.6, $\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{b}\big) \cap \mathbf{M}_{-1} \subseteq \mathbf{S}_{-1}$. Since $\mathbf{M}$ and $\mathbf{M}_{-1}$ only differs at blocks which have not been generated when the adversary state is $\mathcal{S}_{-1}$, we have $\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{b}\big) \cap \mathbf{M}_{-1} = \mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{b}\big) \cap \mathbf{M}$.

$$\begin{aligned} & P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathbf{b}) \\ =& \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{b}\big) \cap \mathbf{M} \backslash \mathbf{S}_{-1}) \\ \leq& \mathrm{TotalW}\left(\big(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{b}\big) \cap \mathbf{M}\big) \setminus \big(\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathbf{b}\big) \cap \mathbf{M}\big)\right) \\ =& \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta} \backslash \mathbf{B}^{\Delta}_{-1}, \mathbf{b}\big) \cap \mathbf{M}) \end{aligned}$$

$\mathrm{SubT}\big(\mathbf{B}^{\Delta}\backslash\mathbf{B}^{\Delta}_{-1},\mathbf{b}\big)\cap\mathbf{M}\neq\emptyset$ only if $e.\mathsf{type}=\mathsf{mRls}$ and $\mathbf{b}\preceq\mathbf{b}_{\mathsf{e}}$. Since $\mathbf{B}^{\Delta}$ and $\mathbf{B}^{\Delta}_{-1}$ can only differ at block $\mathbf{b}_{\mathsf{e}}$, we have

$$P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})-P_{\mathsf{sp}}(\mathcal{S}_{-1},\mathbf{b})\leq\mathbf{b}_{\mathsf{e}}.\mathsf{weight}.$$

Case 2: $\mathbf{c}_{-1}\neq\perp$.

Since $\mathbf{M}$ and $\mathbf{M}_{-1}$ only differs at blocks which have not been generated when the adversary state is $\mathcal{S}_{-1}$, we have $\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}\big)\cap\mathbf{M}_{-1}=\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}\big)\cap\mathbf{M}$.

$$\begin{aligned}
&P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})-P_{\mathsf{sp}}(\mathcal{S}_{-1},\mathbf{b})\\
=&\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta},\mathbf{c}_{-1}\big)\cap\mathbf{M}\backslash\mathbf{S}_{-1})-\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}\big)\cap\mathbf{M}\backslash\mathbf{S}_{-1})\\
\leq&\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}\backslash\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}\big)\cap\mathbf{M}\backslash\mathbf{S}_{-1})\\
\leq&\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}\backslash\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}\big)\cap\mathbf{M}).
\end{aligned}$$

Similar with case 1, $\mathrm{SubT}\big(\mathbf{B}^{\Delta}\backslash\mathbf{B}^{\Delta}_{-1},\mathbf{c}_{-1}\big)\cap\mathbf{M}\neq\emptyset$ only if $e.\mathsf{type}=\mathsf{mRls}$ and $\mathbf{b}\preceq\mathbf{b}_{\mathsf{e}}$. And we have

$$P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})-P_{\mathsf{sp}}(\mathcal{S}_{-1},\mathbf{b})\leq\mathbf{b}_{\mathsf{e}}.\mathsf{weight}.$$

As a summary, $P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})-P_{\mathsf{sp}}(\mathcal{S}_{-1},\mathbf{b})\leq\mathbf{b}_{\mathsf{e}}.\mathsf{weight}$ always holds. And $P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})-P_{\mathsf{sp}}(\mathcal{S}_{-1},\mathbf{b})>0$ only if $\mathbf{c}_{-1}\neq\perp\wedge\mathbf{c}_{-1}\preceq\mathbf{b}_{\mathsf{e}}$ or $\mathbf{c}_{-1}=\perp\wedge\mathbf{b}\preceq\mathbf{b}_{\mathsf{e}}$. □

Lemma A.19 *If* $\mathbf{c}_{-1}\neq\perp$ *and* $\mathbf{c}=\perp$*, we have*

$$(P_{\mathsf{sp}}(\mathcal{S},\mathbf{b})+v)-(P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})+v_{-1})\leq s_{\mathsf{m}}.$$

For other cases,

$$(P_{\mathsf{sp}}(\mathcal{S},\mathbf{b})+v)-(P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})+v_{-1})\leq 0.$$

Proof. We prove this lemma under four cases partitioned by whether $\mathbf{c}=\perp$ and whether $\mathbf{c}_{-1}=\perp$. According to claim 4.6.1, we have $v-v_{-1}\leq\mathrm{TotalW}(\mathbf{S}\backslash\mathbf{S}_{-1})$.

Case 1: $\mathbf{c}_{-1}=\perp$ and $\mathbf{c}=\perp$.

In this case, according to claim 4.6.4, we have $\mathrm{SubT}\big(\mathbf{B}^{\Delta},\mathrm{Tip}(\mathbf{C})\big)\cap\mathbf{M}\subseteq\mathbf{S}$. Since $\mathbf{c}_{-1}=\perp$, we have $\mathrm{Tip}(\mathbf{C})=\mathbf{b}$. Thus

$$\begin{aligned}
&(P_{\mathsf{sp}}(\mathcal{S},\mathbf{b})+v)-(P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})+v_{-1})\\
=&v-v_{-1}-\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta},\mathbf{b}\big)\cap\mathbf{M}\backslash\mathbf{S}_{-1})\\
\leq&v-v_{-1}-\mathrm{TotalW}(\mathbf{S}\backslash\mathbf{S}_{-1})\\
\leq&0.
\end{aligned}$$

Case 2: $\mathbf{c}_{-1}\neq\perp$ and $\mathbf{c}=\perp$.

According to claim 4.6.2, we have $v-v_{-1}\leq s_{\mathsf{m}}$. Thus

$$\begin{aligned}
&(P_{\mathsf{sp}}(\mathcal{S},\mathbf{b})+v)-(P'_{\mathsf{sp}}(\mathcal{S},\mathcal{S}_{-1},\mathbf{b})+v_{-1})\\
=&v-v_{-1}-\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta},\mathbf{c}_{-1}\big)\cap\mathbf{M}\backslash\mathbf{S})\\
\leq&s_{\mathsf{m}}.
\end{aligned}$$

Case 3: $\mathbf{c}_{-1}=\perp$ and $\mathbf{c}\neq\perp$.

According to claim 4.6.4, we have $\mathbf{S}_{-1} \subseteq \mathbf{S}$ and $\mathbf{S}\backslash\mathbf{S}_{-1} \subseteq \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C})\big) \cap \mathbf{M}$. Since $\mathbf{c} \in \mathbf{C}$, we have $\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C})\big) \subseteq \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big)$. Thus

$$\begin{aligned}
&(P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) + v) - (P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \\
=&v - v_{-1} + \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M}\backslash\mathbf{S}) - \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{b}\big) \cap \mathbf{M}\backslash\mathbf{S}_{-1}) \\
\leq&v - v_{-1} + \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M}\backslash\mathbf{S}) - \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M}\backslash\mathbf{S}_{-1}) \\
=&v - v_{-1} - \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M} \cap (\mathbf{S}\backslash\mathbf{S}_{-1})) \\
=&v - v_{-1} - \mathrm{TotalW}(\mathbf{S}\backslash\mathbf{S}_{-1}) \\
\leq&0.
\end{aligned}$$

Case 4: $\mathbf{c}_{-1} \neq \perp$ and $\mathbf{c} \neq \perp$. It will be $\mathbf{c} = \mathbf{c}_{-1}$. For the same reason as case 3, we have

$$\begin{aligned}
&(P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) + v) - (P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \\
=&v - v_{-1} + \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M}\backslash\mathbf{S}) - \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M}\backslash\mathbf{S}_{-1}) \\
\leq&0.
\end{aligned}$$

□

A.3.4 Collect the case discussions

Now we collect the previous results and gives the upper bound for block potential value $P(\mathcal{S}, \mathbf{b}) - P(\mathcal{S}_{-1}, \mathbf{b})$. Similar with the discussion in the second and the third component, we define an intermediate block potential value as

$$P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) := P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) + P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) + P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}).$$

Lemma A.20 *If* $\mathbf{f} = \mathbf{f}_{-1}$*, we have*

$$P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P(\mathcal{S}_{-1}, \mathbf{b}) \leq \Delta(\mathcal{S}_{-1}, e).$$

Proof. First, we define

$$\begin{aligned}
P^{\Delta}_{\mathsf{with}} &:= P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}), \\
P^{\Delta 1}_{\mathsf{adv}} &:= P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}), \\
P^{\Delta 1}_{\mathsf{sp}} &:= P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathbf{b}).
\end{aligned}$$

Thus we have

$$P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P(\mathcal{S}_{-1}, \mathbf{b}) = P^{\Delta}_{\mathsf{with}} + P^{\Delta 1}_{\mathsf{adv}} + P^{\Delta 1}_{\mathsf{sp}}$$

With the assumption $\mathbf{f}_{-1} = \mathbf{f}$, we category all the possible cases for e as follows.

- **Case 1:** $e.\mathsf{type} = \mathsf{mGen}$.
- **Case 2:** $e.\mathsf{type} = \mathsf{Arvl}$.
- **Case 3.1:** $e.\mathsf{type} = \mathsf{mRls}$, $\mathbf{c}_{-1} \neq \perp$ and $e.\mathsf{block} \in \mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{b})\backslash\mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}_{-1})$.
- **Case 3.2:** $e.\mathsf{type} = \mathsf{mRls}$, $\mathbf{c}_{-1} \neq \perp$ and $e.\mathsf{block} \in \mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{c}_{-1})$.
- **Case 3.3:** $e.\mathsf{type} = \mathsf{mRls}$, $\mathbf{c}_{-1} = \perp$ and $e.\mathsf{block} \in \mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{b})$.
- **Case 3.4:** $e.\mathsf{type} = \mathsf{mRls}$ and $e.\mathsf{block} \notin \mathrm{SubT}(\mathbf{B}^{\mathsf{max}}, \mathbf{b})$.
- **Case 4.1:** $e.\mathsf{type} = \mathsf{hGenRls}$ and $e.\mathsf{block}.\mathsf{weight} = 0$.
- **Case 4.2:** $e.\mathsf{type} = \mathsf{hGenRls}$ and $e.\mathsf{block}.\mathsf{weight} = 1$.
- **Case 4.3:** $e.\mathsf{type} = \mathsf{hGenRls}$, $e.\mathsf{block}.\mathsf{weight} = \eta_{\mathsf{w}}$ and $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{True}$.
- **Case 4.4:** $e.\mathsf{type} = \mathsf{hGenRls}$, $e.\mathsf{block}.\mathsf{weight} = \eta_{\mathsf{w}}$ and $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$. According to our rule in updating flag block, if $\mathbf{f}_{-1} = \mathbf{f}$ in this case, there must be $\mathbf{f}_{-1} = \mathbf{f} = \perp$ and that $\mathbf{B}^{\Delta}_{-1}$ has an honest block with block weight η_{w}.

Let $w := e.\mathsf{block.weight}$.

	$P^{\Delta}_{\mathsf{with}}$	$P^{\Delta 1}_{\mathsf{adv}}$	$P^{\Delta 1}_{\mathsf{sp}}$	$\Delta(\mathcal{S}_{-1}, e)$
Case 1	w	0	0	w
Case 2	0	0	0	0
Case 3.1	$-w$	w	0	0
Case 3.2	$-w$	0	w	0
Case 3.3	$-w$	0	w	0
Case 3.4	0	0	0	0
Case 4.1	0	0	0	0
Case 4.2	0	-1	0	-1
Case 4.3	0	0	0	0
Case 4.4	0	0	0	0

Table 1: The upper bounds for each component under different cases (Lemma A.20)

Table 1 shows the upper bounds for $P^{\Delta}_{\mathsf{with}}$, $P^{\Delta 1}_{\mathsf{adv}}$ and $P^{\Delta 1}_{\mathsf{sp}}$ under difference cases. w denotes $e.\mathsf{block.weight}$. The upper bounds for $P^{\Delta}_{\mathsf{with}}$ follow lemma A.11. The upper bounds for $P^{\Delta 1}_{\mathsf{adv}}$ follow lemma A.13 except case 4.2, which follows lemma A.12. The upper bounds for $P^{\Delta 1}_{\mathsf{sp}}$ follow lemma A.18. The last column shows $\Delta(\mathcal{S}_{-1}, e)$ under different cases. For each case (each row in the table), we can check that

$$P^{\Delta}_{\mathsf{with}} + P^{\Delta 1}_{\mathsf{adv}} + P^{\Delta 1}_{\mathsf{sp}} \leq \Delta(\mathcal{S}_{-1}, e).$$

□

Lemma A.21 *If* $\mathbf{f}_{-1} = \mathbf{f}$*, we have*

$$(P(\mathcal{S}, \mathbf{b}) + v) - (P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq 0.$$

Proof. First we define

$$P^{\Delta 2}_{\mathsf{adv}} := P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}_{-1}),$$
$$P^{\Delta 2}_{\mathsf{sp}} := (P(\mathcal{S}, \mathbf{b}) + v) - (P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) + v_{-1}).$$

If $\mathbf{c}_{-1} \neq \perp$ and $\mathbf{c} = \perp$, we have $P^{\Delta 2}_{\mathsf{adv}} \leq -s_{\mathsf{m}}$ (lemma A.14) and $P^{\Delta 2}_{\mathsf{sp}} \leq s_{\mathsf{m}}$ (lemma A.19).
For the other cases, we have $P^{\Delta 2}_{\mathsf{adv}} \leq 0$ (lemma A.14) and $P^{\Delta 2}_{\mathsf{sp}} \leq 0$ (lemma A.19).
Thus we have

$$P(\mathcal{S}, \mathbf{b}) - P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) = P^{\Delta 2}_{\mathsf{adv}} + P^{\Delta 2}_{\mathsf{sp}} \leq 0.$$

□

Lemma A.22 *If* $\mathbf{f}_{-1} \neq \mathbf{f}$*, we have*

$$(P(\mathcal{S}, \mathbf{b}) + v) - (P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e).$$

Proof. First we define

$$P^{\Delta}_{\mathsf{with}} := P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathbf{b}),$$
$$P^{\Delta}_{\mathsf{adv}} := P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{c}_{-1}),$$
$$P^{\Delta 1}_{\mathsf{sp}} := P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \mathbf{b}),$$
$$P^{\Delta 2}_{\mathsf{sp}} := (P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) + v) - (P'_{\mathsf{sp}}(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) + v_{-1}).$$

According to our rule in updating flag block, if $\mathbf{f}_{-1} \neq \mathbf{f}$, e has three possible cases.

- **Case 1:** $e.\mathsf{type} = \mathsf{hGenRls}$, $e.\mathsf{block.weight} = \eta_{\mathsf{w}}$, $\mathbf{f}_{-1} = \perp$ and $\mathbf{f} = e.\mathsf{block}$. According to lemma A.9, in this case, it cannot be $\mathbf{c}_{-1} \neq \perp \wedge \mathbf{c} = \perp$.
- **Case 2:** $e.\mathsf{type} = \mathsf{hGenRls}$, $e.\mathsf{block.weight} = \eta_{\mathsf{w}}$, $\mathbf{f}_{-1} \neq \perp$ and $\mathbf{f} = \perp$. Notice that $\mathbf{f}_{-1} \in \mathbf{B}^{\Delta}_{-1}$ is an honest block with block weight η_{w}.
- **Case 3:** $e.\mathsf{type} = \mathsf{Arvl}$, $e.\mathsf{block.weight} = \eta_{\mathsf{w}}$, $\mathbf{f}_{-1} = e.\mathsf{block}$ and $\mathbf{f} = \perp$. According to lemma A.9, in this case, we have $\mathbf{C}_{-1} = \mathbf{C}$ and thus $\mathbf{c}_{-1} = \mathbf{c}$.

	$P^{\Delta}_{\mathsf{with}}$	$P^{\Delta}_{\mathsf{adv}}$	$P^{\Delta 1}_{\mathsf{sp}}$	$P^{\Delta 2}_{\mathsf{sp}}$	$\Delta(\mathcal{S}_{-1}, e)$
Case: 1	0	$2s_{\mathsf{h}} + 2s_{\mathsf{m}} - \eta_{\mathsf{w}}$	0	0	$2s_{\mathsf{h}} + 2s_{\mathsf{m}} - \eta_{\mathsf{w}}$
Case: 2	0	η_{w}	0	s_{m}	$\eta_{\mathsf{w}} + s_{\mathsf{m}}$
Case: 3	0	0	0	0	0

Table 2: The upper bounds for each component under different cases (Lemma A.22)

Table 2 shows the upper bounds for $P^{\Delta}_{\mathsf{with}}$, $P^{\Delta}_{\mathsf{adv}}$, $P^{\Delta 1}_{\mathsf{sp}}$ and $P^{\Delta 2}_{\mathsf{sp}}$ under difference cases. The upper bounds for $P^{\Delta}_{\mathsf{with}}$ follow lemma A.11. The upper bounds for $P^{\Delta}_{\mathsf{adv}}$ follow lemma A.15 (case 1), lemma A.16 (case 2) and lemma A.17 (case 3). The upper bounds for $P^{\Delta 1}_{\mathsf{sp}}$ follow lemma A.18. The upper bounds for $P^{\Delta 2}_{\mathsf{sp}}$ follow lemma A.19. The last column shows $\Delta(\mathcal{S}_{-1}, e)$ under different cases. For each case (each row in the table), we can check that

$$P^{\Delta}_{\mathsf{with}} + P^{\Delta}_{\mathsf{adv}} + P^{\Delta 1}_{\mathsf{sp}} + P^{\Delta 2}_{\mathsf{sp}} \leq \Delta(\mathcal{S}_{-1}, e).$$

□

Now we reach the result for the section A.3.

Lemma A.23 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. For any block $\mathbf{b}$, if $P(\mathcal{S}_{-1}, \mathbf{b}) \neq \perp$ and $P(\mathcal{S}, \mathbf{b}) \neq \perp$, we have*

$$(P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e).$$

Proof. If $\mathbf{f}_{-1} = \mathbf{f}$, we have $P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) - P(\mathcal{S}, \mathbf{b}_{-1}) \leq \Delta(\mathcal{S}_{-1}, e)$ (lemma A.20) and $(P(\mathcal{S}, \mathbf{b}) + v) - (P'(\mathcal{S}, \mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq 0$ (lemma A.21). If $\mathbf{f}_{-1} \neq \mathbf{f}$, we have $(P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e)$ (lemma A.22). So we always have

$$(P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e).$$

□

A.4 Case discussions for potential value (Part 2)

Common settings For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. In this sub-section, we study the upper bound of block potential value for the case not covered in section A.3. If $P(\mathcal{S}_{-1}, \mathbf{b}) = \perp$ and $P(\mathcal{S}, \mathbf{b}) \neq \perp$, we can not estimate the upper bound of $P(\mathcal{S}, \mathbf{b})$ by $P(\mathcal{S}, \mathbf{b}) - P(\mathcal{S}_{-1}, \mathbf{b})$. We try to estimate $P(\mathcal{S}, \mathbf{b})$ by $P(\mathcal{S}, \mathbf{b}) - P(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1}))$ instead.

In this sub-section, we assume $P(\mathcal{S}_{-1}, \mathbf{b}) = \perp$, $P(\mathcal{S}, \mathbf{b}) \neq \perp$ and $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_{-1}, \mathbf{b}) = \mathsf{True}$. All the lemmas in section A.4 are discussed under these assumptions. We will not repeat them in each lemma (except lemma A.28).

We define symbol $\mathbf{c} := \mathrm{Next}(\mathbf{C}, \mathbf{b})$ for given $\mathbf{C}$ and $\mathbf{b}$ in the context. According to lemma A.8, one of $\mathbf{C}_{-1}$ and $\mathbf{C}$ must be the prefix of another. Since $P(\mathcal{S}_{-1}, \mathbf{b}) = \perp$ and $Old(\mathbf{B}^{\mathsf{min}}_{-1}, \mathbf{b}) = \mathsf{True}$, we have $\mathbf{b} \notin \mathbf{C}_{-1}$. Since $\mathbf{b} \in \mathbf{C}$, $\mathbf{C}_{-1}$ must be a prefix of $\mathbf{C}$ and it is strictly shorter than $\mathbf{C}$. It must be

$$\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{b} \preceq \mathrm{Tip}(\mathbf{C}).$$

According to lemma A.9, we have the following claims. All the proofs will refer this claim implicitly.

Claim A.24 *Under the common assumption of section A.4, there could be one of following two cases for event e.*

- $e.\mathsf{type} = \mathsf{Arvl}$ *and* $\mathbf{f}_{-1} = \mathbf{f}$
- $e = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \perp$ *and* $\mathbf{f} = e.\mathsf{block}$

Lemma A.25 $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}, \mathrm{Tip}(\mathbf{C}_{-1})) \leq 0$.

Proof. Since $e.\mathsf{type} \notin \{\mathsf{mGen}, \mathsf{mRls}\}$, it can be verified that $\mathbf{B}^{\mathsf{gen}}_{-1} \backslash \mathbf{B}^{\mathsf{max}}_{-1} = \mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}}$. Thus $P_{\mathsf{with}}(\mathcal{S}, \mathrm{Tip}(\mathbf{C}_{-1})) = P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1}))$. Since $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{b}$, $\mathrm{SubT}\big(\mathbf{B}^{\mathsf{gen}}_{-1} \backslash \mathbf{B}^{\mathsf{max}}_{-1}, \mathbf{b}\big) \subseteq \mathrm{SubT}\big(\mathbf{B}^{\mathsf{gen}}_{-1} \backslash \mathbf{B}^{\mathsf{max}}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})\big)$. So we have

$$P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}, \mathrm{Tip}(\mathbf{C}_{-1})) \leq 0.$$

□

Lemma A.26 *If* $e = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \perp$ *and* $\mathbf{f} = e.\mathsf{block}$, *we have*

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \perp) \leq 2s_{\mathsf{h}} + 2s_{\mathsf{m}} - \eta_{\mathsf{w}}.$$

If $e.\mathsf{type} = \mathsf{Arvl}$ *and* $\mathbf{f}_{-1} = \mathbf{f}$, *we have*

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \perp) \leq 0.$$

Proof. We discuss two cases respectively.

Case 1: $e = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \perp$ and $\mathbf{f} = e.\mathsf{block}$. Let $\mathbf{b}_{\mathsf{e}} := e.\mathsf{block}$. If $e.\mathsf{type} = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \perp$ and $\mathbf{f} = e.\mathsf{block}$, according to lemma A.7, for all the block $\mathbf{b}'$ in $\mathrm{Chain}(\mathbf{b}_{\mathsf{e}})$, we have $\mathrm{Adv}(\mathcal{S}, \mathbf{b}') > \eta_{\mathsf{w}} - s_{\mathsf{h}} - s_{\mathsf{m}} \geq s_{\mathsf{h}} + s_{\mathsf{m}}$. According to claim 4.5.3, $\mathrm{Chain}(\mathbf{b}_{\mathsf{e}})$ is a prefix of $\mathbf{C}$. Since $\mathbf{b}_{\mathsf{e}} \notin \mathbf{B}^{\mathsf{min}}$, $\mathbf{b}_{\mathsf{e}}$ cannot be an old enough block and thus $P(\mathcal{S}, \mathbf{b}_{\mathsf{e}}) = \perp$. So $\mathbf{b}$ cannot be the last block in $\mathbf{C}$. We claim

$$\mathbf{c} \neq \perp \text{ and } \mathbf{c} \in \mathbf{C}.$$

Since $\mathbf{c} \in \mathbf{C}$, according to lemma A.7, $\mathrm{Adv}(\mathcal{S}, \mathbf{c}) \geq \eta_{\mathsf{w}} - s_{\mathsf{h}} - s_{\mathsf{m}}$. Thus

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \perp) \leq s_{\mathsf{h}} + s_{\mathsf{m}} - \mathrm{Adv}(\mathcal{S}, \mathbf{c}) \leq 2s_{\mathsf{h}} + 2s_{\mathsf{m}} - \eta_{\mathsf{w}}.$$

Case 2: $e = \mathsf{Arvl}$ and $\mathbf{f}_{-1} = \mathbf{f}$.
If $\mathbf{c} = \perp$, then $P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \perp) = 0$ holds trivially.
If $\mathbf{c} \neq \perp$, since $\mathbf{c} \in \mathbf{C}$ and $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{b} \prec \mathbf{c}$, according to claim 4.5.1, $\mathrm{Adv}(\mathcal{S}, \mathbf{c}) > s_{\mathsf{h}} + s_{\mathsf{m}}$. Thus

$$P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \perp) \leq s_{\mathsf{h}} + s_{\mathsf{m}} - \mathrm{Adv}(\mathcal{S}, \mathbf{c}) < 0.$$

□

Lemma A.27 $(P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) + v) - (P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) + v_{-1}) \leq 0$.

Proof. Since $\mathrm{Next}(\mathbf{C}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) = \perp$, we have $P_{\mathsf{sp}}(\mathcal{S}, \mathrm{Tip}(\mathbf{C}_{-1})) = v_{-1}$. According to claim 4.6,

$$\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})\big) \cap \mathbf{M}_{-1} \subseteq \mathbf{S}_{-1}.$$

If $e.\mathsf{type} = \mathsf{hGenRls}$, since $\mathbf{b}_{\mathsf{e}}$ is not malicious block, $\mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})\big) \cap \mathbf{M}_{-1}$ must equal to $\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C}_{-1})\big) \cap \mathbf{M}$. If $e.\mathsf{type} = \mathsf{Arvl}$, then $\mathbf{B}^{\Delta} \subseteq \mathbf{B}^{\Delta}_{-1}$, we have $\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C}_{-1})\big) \cap \mathbf{M}_{-1} \subseteq \mathrm{SubT}\big(\mathbf{B}^{\Delta}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})\big) \cap \mathbf{M}$. Recalling that $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathrm{Tip}(\mathbf{C})$,

$$\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C})\big) \cap \mathbf{M} \subseteq \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C}_{-1})\big) \cap \mathbf{M} \subseteq \mathbf{S}_{-1}.$$

According to the rule in updating $\mathbf{S}$, we have $\mathbf{S} = \mathbf{S}_{-1} \cup \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C})\big) \cap \mathbf{M} = \mathbf{S}_{-1}$. Thus $v - v_{-1} = 0$ according to claim 4.6.3.

If $\mathbf{c} = \perp$, we have

$$(P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) + v) - (P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) + v_{-1}) = 0.$$

If $\mathbf{c} \neq \bot$, since $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{c}$, we have

$$\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M} \subseteq \mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathrm{Tip}(\mathbf{C}_{-1})\big) \cap \mathbf{M} \subseteq \mathbf{S}_{-1}.$$

Thus

$$(P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) + v) - (P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) + v_{-1}) = \mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}, \mathbf{c}\big) \cap \mathbf{M} \backslash \mathbf{S}_{-1}) = 0.$$

□

Lemma A.28 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. For any block $\mathbf{b}$ with $P(\mathcal{S}_{-1}, \mathbf{b}) = \bot$, $P(\mathcal{S}, \mathbf{b}) \neq \bot$ and $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_{-1}, \mathbf{b}) =$ True, we have*

$$(P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e).$$

Proof. First we define

$$\begin{aligned} P^{\Delta}_{\mathsf{with}} &:= P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) - P_{\mathsf{with}}(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})), \\ P^{\Delta}_{\mathsf{adv}} &:= P_{\mathsf{adv}}(\mathcal{S}, \mathbf{c}) - P_{\mathsf{adv}}(\mathcal{S}_{-1}, \bot), \\ P^{\Delta}_{\mathsf{sp}} &:= (P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) + v) - (P_{\mathsf{sp}}(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) + v_{-1}). \end{aligned}$$

Then we have

$$(P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) + v_{-1}) = P^{\Delta}_{\mathsf{with}} + P^{\Delta}_{\mathsf{adv}} + P^{\Delta}_{\mathsf{sp}}.$$

According two claim A.24, we have two possible cases

Case 1: $e.\mathsf{type} = \mathsf{Arvl}$ and $\mathbf{f}_{-1} = \mathbf{f}$

In this case, we have $\Delta(\mathcal{S}_{-1}, e) = 0$. According to lemma A.25, A.26 and A.27, we have $P^{\Delta}_{\mathsf{with}} \leq 0$, $P^{\Delta}_{\mathsf{adv}} \leq 0$ and $P^{\Delta}_{\mathsf{sp}} \leq 0$. So we claim

$$P^{\Delta}_{\mathsf{with}} + P^{\Delta}_{\mathsf{adv}} + P^{\Delta}_{\mathsf{sp}} \leq \Delta(\mathcal{S}_{-1}, e).$$

Case 2: $e = \mathsf{hGenRls}$, $\mathbf{f}_{-1} = \bot$ and $\mathbf{f} = e.\mathsf{block}$ In this case, we have $\Delta(\mathcal{S}_{-1}, e) = 2s_{\mathsf{h}} + 2s_{\mathsf{m}} - \eta_{\mathsf{w}}$. According to lemma A.25, A.26 and A.27, we have $P^{\Delta}_{\mathsf{with}} \leq 0$, $P^{\Delta}_{\mathsf{adv}} \leq 2s_{\mathsf{h}} + 2s_{\mathsf{m}} - \eta_{\mathsf{w}}$ and $P^{\Delta}_{\mathsf{sp}} \leq 0$. So we claim

$$P^{\Delta}_{\mathsf{with}} + P^{\Delta}_{\mathsf{adv}} + P^{\Delta}_{\mathsf{sp}} \leq \Delta(\mathcal{S}_{-1}, e).$$

□

A.5 Proof of Theorem 4.7

Proof. For any $n \geq N(r_1)$, we have assumed the following two conditions.

- $\tilde{P}(\mathcal{S}_n, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) < -\eta_{\mathsf{w}}$
- For any block $\mathbf{b} \in \mathbf{B}^{\mathsf{gen}}_n$ with $\mathbf{B}^{\mathsf{min}}_{N(r_0)} \nsubseteq \mathbf{b}.\mathsf{past}$, it will be $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_n, \mathbf{b}) = \mathsf{True}$.

Let $\mathbf{b}'_n \in \mathbf{C}_n$ be the last block satisfying $\mathbf{B}^{\mathsf{min}}_{N(r_0)} \nsubseteq \mathbf{b}'_n.\mathsf{past}$ in $\mathbf{C}_n$. According to the second assumption, $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}_n, \mathbf{b}'_n) = \mathsf{True}$. According to the definition of global potential value, the block potential value of all the blocks in $\mathrm{Chain}(\mathbf{b}'_n)$ are taken into considered. Thus $\tilde{P}(\mathcal{S}_n, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) \geq \max_{\mathbf{b} \in \mathrm{Chain}(\mathbf{b}'_n)} P(\mathcal{S}_n, \mathbf{b})$. So we have $P(\mathcal{S}_n, \mathbf{b}'_n) < -\eta_{\mathsf{w}}$. According to the definition of block potential value, $P(\mathcal{S}_n, \mathbf{b}'_n) < 0$ only if $\mathbf{b}'_n \neq \mathrm{Tip}(\mathbf{C}_n)$. So we can let $\mathbf{b}_n := \mathrm{Next}(\mathbf{C}_n, \mathbf{b}'_n)$. Since $\mathbf{b}'_n$ is the last block satisfying $\mathbf{B}^{\mathsf{min}}_{N(r_0)} \nsubseteq \mathbf{b}'_n.\mathsf{past}$, there must be $\mathbf{B}^{\mathsf{min}}_{N(r_0)} \subseteq \mathbf{b}_n.\mathsf{past}$.

Notice that $\max_{\mathbf{b} \in \mathrm{Chain}(\mathbf{b}'_n)} P(\mathcal{S}_n, \mathbf{b}) \leq \tilde{P}(\mathcal{S}_n, \mathbf{B}^{\mathsf{min}}_{N(r_0)}) < -\eta_{\mathsf{w}}$, thus we have

$$\forall \mathbf{b} \in \mathrm{Chain}(\mathbf{b}'_n), P(\mathcal{S}_n, \mathbf{b}) < -\eta_{\mathsf{w}}.$$

Since $P(\mathcal{S}_n, \mathbf{b}) \geq -\mathrm{Adv}(\mathcal{S}_n, \mathrm{Next}(\mathbf{C}_n, \mathbf{b}))$, we have

$$\forall \mathbf{b} \in \mathrm{Chain}(\mathbf{b}_n) \backslash \{\mathbf{g}\},\ \mathrm{Adv}(\mathcal{S}_n, \mathbf{b}) < -\eta_{\mathsf{w}}.$$

Since

$$\begin{aligned}\mathrm{Adv}(\mathcal{S}_n, \mathbf{b}) =& \mathrm{SubTW}\big(\mathbf{B}_n^{\mathsf{min}} \cup \{\mathbf{f}_n\}, \mathbf{b}\big) - \mathrm{SibSubTW}(\mathbf{B}_n^{\mathsf{max}}, \mathbf{b}) \\ \leq& \eta_{\mathsf{w}} + \mathrm{SubTW}\big(\mathbf{B}_n^{\mathsf{min}}, \mathbf{b}\big) - \mathrm{SibSubTW}(\mathbf{B}_n^{\mathsf{max}}, \mathbf{b}),\end{aligned}$$

we have

$$\forall \mathbf{b} \in \mathrm{Chain}(\mathbf{b}_n) \backslash \{\mathbf{g}\}, \mathrm{SubTW}\big(\mathbf{B}_n^{\mathsf{min}}, \mathbf{b}\big) - \mathrm{SibSubTW}(\mathbf{B}_n^{\mathsf{max}}, \mathbf{b}) > 0.$$

According to lemma A.1, $\mathbf{b}_n \in \mathrm{Pivot}(\mathcal{B})$ for all the local state $\mathbf{B}_n^{\mathsf{min}} \subseteq \mathcal{B} \subseteq \mathbf{B}_n^{\mathsf{max}}$. It means that for any honest participant, $\mathbf{b}_n$ is in its pivot chain. Since $\mathbf{B}_{N(r_0)}^{\mathsf{min}} \subseteq \mathbf{b}_n.\mathsf{past}$, for any block $\tilde{\mathbf{b}} \in \mathbf{B}_{N(r_0)}^{\mathsf{min}}$, its history is determined by block $\mathbf{b}_n$. Formally, $\mathrm{Prefix}(\mathcal{C}(\mathcal{B}), \tilde{\mathbf{b}})$ is a prefix of $\mathcal{C}(\mathbf{b}_n.\mathsf{past})$.

Then we will show that for any $n \geq N(r_1)$, $\mathbf{b}_n$ and $\mathbf{b}_{n+1}$ are the same block. Since $\mathbf{b}_n$ and $\mathbf{b}_{n+1}$ is the first block $\mathbf{b}'$ satisfying $\mathbf{B}_{N(r_0)}^{\mathsf{min}} \subseteq \mathbf{b}'.\mathsf{past}$ in $\mathbf{C}_n$ and $\mathbf{C}_{n+1}$. So there can not be $\mathbf{b}_n \prec \mathbf{b}_{n+1}$ or $\mathbf{b}_{n+1} \prec \mathbf{b}_n$. According to lemma A.8, one of $\mathbf{C}_n$ and $\mathbf{C}_{n+1}$ must be the prefix of another, thus there must be $\mathbf{b}_n = \mathbf{b}_{n+1}$.

So for all the $n \geq N(r_1)$, block $\mathbf{b}_n$ refers the same block and it is in the pivot chain of all the honest participants. According to the block ordering algorithm $\mathcal{C}_{\mathsf{GHAST}}$ (recalling that $\mathcal{C}_{\mathsf{GHAST}}$ is defined the same as $\mathcal{C}_{\mathsf{TG}}$ except the block weight), the history of blocks in $\mathbf{b}_n.\mathsf{past}$ must be a prefix of $\mathcal{C}(\mathbf{b}_n.\mathsf{past})$. Notice that $\mathbf{B}_{N(r_0)}^{\mathsf{min}} \subseteq \mathbf{b}_n.\mathsf{past}$, we have

$$\forall \tilde{\mathbf{b}} \in \mathbf{B}_{N(r_0)}^{\mathsf{min}}, \left| \bigcup_{\substack{r \in \{r_1, \cdots, r_{\mathsf{max}}\} \\ \mathcal{B} \in \mathcal{U}_r}} \mathrm{Prefix}(\mathcal{C}_{\mathsf{GHAST}}(\mathcal{B}), \tilde{\mathbf{b}}) \right| = 1.$$

□

A.6 Proof of Theorem 4.8

Proof. Combining the conclusions in lemma A.23 and lemma A.28, we proved this lemma. □

A.7 Proof of Theorem 4.9

Proof. Recalling that $\tilde{P}(\mathcal{S}, \mathbf{B})$ picks the maximum block potential value of blocks in $\mathbf{C}' := \{\mathbf{b}' \in \mathbf{C} | \mathbf{B} \nsubseteq \mathbf{b}'.\mathsf{past}\}$. Since genesis block $\mathbf{g}$ must be in $\mathbf{C}'$ and $\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{g}) = \mathsf{True}$, there must exists block in $\mathbf{C}'$ whose block potential value is not $\perp$. Let $\mathbf{b}$ be the block with maximum block potential value in $\mathbf{C}'$. (A.k.a. $\mathbf{b} := \arg\max_{\mathbf{b}' \in \mathbf{C}'} P(\mathcal{S}, \mathbf{b}')$). Thus

$$\tilde{P}(\mathcal{S}, \mathbf{B}) = P(\mathcal{S}, \mathbf{b}).$$

Since $P(\mathcal{S}, \mathbf{b}) \neq \perp$, we have $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True}$. In the assumptions of this theorem, $\mathrm{Old}(\mathbf{B}^{\mathsf{min}}, \mathbf{b}') = \mathsf{False} \vee \mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}') = \mathsf{True} \vee \tilde{P}(\mathcal{S}, \mathbf{B}) = P(\mathcal{S}, \mathbf{b}')$ holds for all the block $\mathbf{b}' \in \mathbf{C}'$. So we have

$$\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True}.$$

Since $\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True}$, it must be $\mathbf{b} \in \mathbf{B}_{-1}^{\mathsf{min}}$. Thus $\mathbf{b}.\mathsf{past} \subseteq \mathbf{B}_{-1}^{\mathsf{min}} \subseteq \mathbf{B}_{-1}^{\mathsf{max}}$. Since we assume $\mathbf{B}_{-1}^{\mathsf{max}} \cap \mathbf{B}_{-1} = \mathbf{B}_{-1}^{\mathsf{max}} \cap \mathbf{B}$, thus $\mathbf{b}.\mathsf{past} \cap \mathbf{B} = \mathbf{b}.\mathsf{past} \cap \mathbf{B}_{-1}$. If $\mathbf{B}_{-1} \subseteq \mathbf{b}.\mathsf{past}$, then it will be $\mathbf{B} \subseteq \mathbf{b}.\mathsf{past}$ and thus $\mathbf{b} \notin \mathbf{C}'$. This contradicts $\mathbf{b} \in \mathbf{C}'$. Thus we have

$$\mathbf{B}_{-1} \nsubseteq \mathbf{b}.\mathsf{past}.$$

Case 1: $P(\mathcal{S}_{-1}, \mathbf{b}) \neq \perp$.

Since $P(\mathcal{S}_{-1}, \mathbf{b}) \neq \perp$ and $\mathbf{b} \in \mathbf{C}_{-1}$, we have $\mathbf{b} \in \mathbf{C}'_{-1}$. Recalling that $\tilde{P}(\mathcal{S}_{-1}, \mathbf{B}_{-1})$ picks the maximum block potential value of blocks in $\mathbf{C}'_{-1} := \{\mathbf{b}' \in \mathbf{C}_{-1} | \mathbf{b}'.\mathsf{past} \nsubseteq \mathbf{B}_{-1}\}$, there must be $P(\mathcal{S}_{-1}, \mathbf{b}) \leq \tilde{P}(\mathcal{S}_{-1}, \mathbf{B}_{-1})$. According to lemma A.23, $(P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e)$. Thus

$$(\tilde{P}(\mathcal{S}, \mathbf{B}) + v) - (\tilde{P}(\mathcal{S}_{-1}, \mathbf{B}_{-1}) + v_{-1}) \leq (P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathbf{b}) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e).$$

Case 2: $P(\mathcal{S}_{-1}, \mathbf{b}) = \perp$.

Since $P(\mathcal{S}_{-1}, \mathbf{b}) = \perp$ and $\mathrm{Old}(\mathbf{B}_{-1}^{\mathsf{min}}, \mathbf{b}) = \mathsf{True}$, we have $\mathbf{b} \notin \mathbf{C}_{-1}$. According to lemma A.8, $\mathbf{C}$ must be a prefix of $\mathbf{C}_{-1}$. Thus $\mathrm{Tip}(\mathbf{C}_{-1}) \prec \mathbf{b}$. So we have $\mathrm{Tip}(\mathbf{C}_{-1}).\mathsf{past} \subseteq \mathbf{b}.\mathsf{past}$. Thus

$$\mathbf{B}_{-1} \nsubseteq \mathrm{Tip}(\mathbf{C}_{-1}).\mathsf{past}.$$

Recalling that $\tilde{P}(\mathcal{S}_{-1}, \mathbf{B}_{-1})$ picks the maximum block potential value of blocks in $\mathbf{C}'_{-1} := \{\mathbf{b}' \in \mathbf{C}_{-1} | \mathbf{b}'.\mathsf{past} \nsubseteq \mathbf{B}_{-1}\}$, there must be $P(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) \leq \tilde{P}(\mathcal{S}_{-1}, \mathbf{B}_{-1})$. According to lemma A.28, $(P(\mathcal{S}, \mathbf{b})+v)-(P(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1}))+v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e)$. Thus

$$(\tilde{P}(\mathcal{S}, \mathbf{B}) + v) - (\tilde{P}(\mathcal{S}_{-1}, \mathbf{B}_{-1}) + v_{-1}) \leq (P(\mathcal{S}, \mathbf{b}) + v) - (P(\mathcal{S}_{-1}, \mathrm{Tip}(\mathbf{C}_{-1})) + v_{-1}) \leq \Delta(\mathcal{S}_{-1}, e).$$

□

B The Summation of Event Values

B.1 Decompose Event Values

The definition of potential value is not friendly for analysis probability distribution, we elaborate the decomposition of the event value into several components: $\Delta_{\mathsf{H}}(\mathcal{S}_{-1}, e)$, $\Delta_{\mathsf{M}}(\mathcal{S}_{-1}, e)$, $\Delta_{\mathsf{F}}(\mathcal{S}_{-1}, e)$ and $\Delta_{\mathsf{T}}(\mathcal{S}_{-1}, e)$. And shows that the event value is always no more than the sum of components in lemma B.1.

- When e is a mGen event, $\Delta_{\mathsf{M}}(\mathcal{S}_{-1}, e) := e.\mathsf{block.weight}$. For other cases, $\Delta_{\mathsf{M}}(\mathcal{S}_{-1}, e) := 0$.
- When e is a $\mathsf{hGenRls}$ event and $\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$, $\Delta_{\mathsf{H}}(\mathcal{S}_{-1}, e) := -(\eta_{\mathsf{w}} - 2s_{\mathsf{h}} - 2s_{\mathsf{m}})/\eta_{\mathsf{w}} \cdot e.\mathsf{block.weight}$. For other cases. It equals to 0 for other cases.
- $\Delta_{\mathsf{F}}(\mathcal{S}_{-1}, e) := 2\eta_{\mathsf{w}} - 2s_{\mathsf{h}} - s_{\mathsf{m}}$ if e is an $\mathsf{hGenRls}$ event of block with block weight η_{w} and $\mathbf{B}_{-1}^{\Delta}$ has an honest block with block weight η_{w}. It equals to 0 for other cases.
- $\Delta_{\mathsf{T}}(\mathcal{S}_{-1}, e) := -s_{\mathsf{m}}$ if e is an $\mathsf{hGenRls}$ event of block with block weight η_{w} and $\mathbf{B}_{-1}^{\Delta}$ has at least two honest blocks with block weight η_{w}. It equals to 0 for other cases.

Lemma B.1 *For any adversary state $\mathcal{S}$ appearing in the execution of ghast protocol, let $\mathcal{S}_{-1}$ be the last adversary state and e be the event updates $\mathcal{S}_{-1}$ to $\mathcal{S}$. We have*

$$\Delta(\mathcal{S}_{-1}, e) \leq \Delta_{\mathsf{M}}(\mathcal{S}_{-1}, e) + \Delta_{\mathsf{H}}(\mathcal{S}_{-1}, e) + \Delta_{\mathsf{F}}(\mathcal{S}_{-1}, e) + \Delta_{\mathsf{T}}(\mathcal{S}_{-1}, e).$$

Proof. For the case $e.\mathsf{type} \in \{\mathsf{mRls}, \mathsf{Arvl}\}$, the event value and all its components always be 0. So the inequality holds trivially.

If $e.\mathsf{type} = \mathsf{mGen}$, the event value components $\Delta_{\mathsf{H}}, \Delta_{\mathsf{F}}, \Delta_{\mathsf{T}}$ are all 0. So we have $\Delta(\mathcal{S}_{-1}, e) = e.\mathsf{block.weight} = \Delta_{\mathsf{M}}(\mathcal{S}_{-1}, e)$.

If $e.\mathsf{type} = \mathsf{hGenRls}$ and $e.\mathsf{block.weight} = 0$, the event value and all its components must be 0.

If $e.\mathsf{type} = \mathsf{hGenRls}$ and $e.\mathsf{block.weight} = 1$, the event value components $\Delta_{\mathsf{M}}, \Delta_{\mathsf{F}}, \Delta_{\mathsf{T}}$ are all 0. When $\mathrm{Spe}(\mathcal{S}_{-1}, e) = \mathsf{False}$, we have $\Delta(\mathcal{S}_{-1}, e) = -1$ and $\Delta_{\mathsf{H}}(\mathcal{S}_{-1}, e) = -1 + (2s_{\mathsf{h}} + 2s_{\mathsf{m}})/\eta_{\mathsf{w}} \geq -1$. When $\mathrm{Spe}(\mathcal{S}_{-1}, e) = \mathsf{True}$, we have $\Delta(\mathcal{S}_{-1}, e) = 0$ and $\Delta_{\mathsf{H}}(\mathcal{S}_{-1}, e) = 0$. So the inequality holds for this case.

If $e.\mathsf{type} = \mathsf{hGenRls}$ and $e.\mathsf{block.weight} = \eta_{\mathsf{w}}$, there must be $\Delta_{\mathsf{M}}(\mathcal{S}_{-1}, e) = 0$. Table 3 lists the value of $\Delta_{\mathsf{H}}(\mathcal{S}_{-1}, e)$, $\Delta_{\mathsf{F}}(\mathcal{S}_{-1}, e)$, $\Delta_{\mathsf{T}}(\mathcal{S}_{-1}, e)$ and $\Delta(\mathcal{S}_{-1}, e)$ under all the possible cases. We can check that $\Delta_{\mathsf{H}}(\mathcal{S}_{-1}, e) + \Delta_{\mathsf{F}}(\mathcal{S}_{-1}, e) + \Delta_{\mathsf{T}}(\mathcal{S}_{-1}, e) \geq \Delta(\mathcal{S}_{-1}, e)$ holds for all the cases.

□

Let $w_1 := 2s_{\mathsf{m}} + 2s_{\mathsf{h}} - \eta_{\mathsf{w}}$ and $w_2 := 2\eta_{\mathsf{w}} - 2s_{\mathsf{h}} - s_{\mathsf{m}}$.

		Δ_{H}	Δ_{F}	Δ_{T}	Δ
$\lvert\mathbf{H}_{-1}\rvert = 0$ [1]	$\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{False}$	w_1	0	0	w_1
	$\mathrm{Spe}(\mathcal{S}_{-1}) = \mathsf{True}$	0	0	0	0
$\lvert\mathbf{H}_{-1}\rvert = 1$ [1]	$\mathbf{f} = \perp$	$\geq w_1$	w_2	0	0
	$\mathbf{f} \neq \perp$	$\geq w_1$	w_2	0	$\eta_{\mathsf{w}} + s_{\mathsf{m}}$
$\lvert\mathbf{H}_{-1}\rvert \geq 2$ [1]	$\mathbf{f} = \perp$	$\geq w_1$	w_2	$-s_{\mathsf{m}}$	0
	$\mathbf{f} \neq \perp$	Impossible [2]			

1. Let $\mathbf{H}_{-1}$ includes all the honest blocks in $\mathbf{B}^{\Delta}_{-1}$ with block weight η_{w}.
2. If $\mathbf{f} \neq \perp$, there must be $|\mathbf{H}_{-1}| = 1$ according to our rule in updating flag block.

Table 3: Event value and its components under different cases.

B.2 Probability for each Component

Recalling that $\vec{\eta} := (\eta_{\mathsf{d}}, \eta_{\mathsf{w}}, \eta_{\mathsf{a}}, \eta_{\mathsf{t}}, \eta_{\mathsf{b}})$ and $(m, \beta, d, \mathcal{A}, \mathcal{Z})$ admissible w.r.t. $(\Pi^{\vec{\eta}}_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})$ in our analysis. $\mathsf{View}^{(\Pi_{\mathsf{GHAST}}, \mathcal{C}_{\mathsf{GHAST}})}(\mathcal{Z}, \mathcal{A}, \kappa)$ is the random variable denote the joint view of all the participant nodes and the adversary in all rounds. We denote it as View in section B.2. Now we define several random variables determined by View. Let View_r denote the joint view before round r.

Let e_n denote the n^{th} event since the ghast protocol launched and $\mathcal{S}_{n-1}$ denote the adversary state when event e_n happens. Similarly, symbols $\mathbf{B}^{\mathsf{gen}}_n$, $\mathbf{B}^{\mathsf{max}}_n$, $\mathbf{B}^{\mathsf{min}}_n$, $\mathbf{B}^{\Delta}_n$, $\mathbf{M}_n$, $\mathbf{f}_n$, $\mathbf{C}_n$, $\mathbf{S}_n$, v_n denote corresponding components of $\mathcal{S}_n$ in the context for any subscript. For any round r, let $N(r)$ be the index of last event before round r. We define random variables M_r, H_r, F_r as follows:

$$M_r := \sum_{i=N(r)+1}^{N(r+1)} \Delta_{\mathsf{M}}(\mathcal{S}_{i-1}, e_i) \quad H_r := \sum_{i=N(r)+1}^{N(r+1)} \Delta_{\mathsf{H}}(\mathcal{S}_{i-1}, e_i) \quad F_r := \sum_{i=N(r)+1}^{N(r+1)} \Delta_{\mathsf{F}}(\mathcal{S}_{i-1}, e_i) \tag{16}$$

For the case no event happens in round r, it will be $N(r) = N(r+1)$, so all the three random variables equal to 0 in this case.

We use valued random variable S_r to denote if there exists an adversary state $\mathcal{S}$ with $\mathrm{Spe}(\mathcal{S}) = \mathsf{True}$ in phase 3 of round r. Futhermore, we use boolean-valued random variables $S^{\mathsf{M}}_r, S^{\mathsf{H1}}_r, S^{\mathsf{H2}}_r$ to distinguish the reason in triggering special status. S^{M}_r denotes there exists adversary state $\mathcal{S}$ in phase 3 or round r which satisfies the first rule in the definition of special status (definition 4.2). S^{H1}_r and S^{H2}_r correspond to the second rule and the third rule. The random variables equal to 1 for the "True" statement and equal to 0 otherwise.

Claim B.2 *For any given round r, $S_r \leq S^{\mathsf{H1}}_r + S^{\mathsf{H2}}_r + S^{\mathsf{M}}_r$.*

Lemma B.3 *For any given round $r_1 < r_2$, we have*

$$(v_{N(r_2)} - v_{N(r_1)})/s_{\mathsf{m}} \geq \sum_{i=r_1}^{r_2-1} S^{\mathsf{M}}_i/(d+1) - 1.$$

Proof. For any round r with $S^{\mathsf{M}}_r = 1$, we have adversary state $\mathcal{S}_n$ with $N(r) < n \leq N(r+1)$ such that

$$\mathrm{TotalW}(\mathrm{SubT}\big(\mathbf{B}^{\Delta}_n, \mathrm{Tip}(\mathbf{C}_n)\big) \cap \mathbf{M}_n) \geq s_{\mathsf{m}}.$$

Let $\mathbf{T}_n := \mathrm{SubT}\big(\mathbf{B}^{\Delta}_n, \mathrm{Tip}(\mathbf{C}_n)\big)$. According to claim 4.6, we have $\mathbf{T}_n \cap \mathbf{M}_n \subseteq \mathbf{S}_n$. Since $\mathbf{B}^{\Delta}_n$ only contains blocks generated no earlier than round $r-d+1$. Let $n' = N(r-d)$, we have $\mathbf{B}^{\Delta}_n \cap \mathbf{B}^{\mathsf{gen}}_{n'} = \emptyset$. Thus $(\mathbf{T}_n \cap \mathbf{M}_n) \cap \mathbf{S}_{n'} = \emptyset$. So we have

$$\mathrm{TotalW}(\mathbf{S}_n \backslash \mathbf{S}_{n'}) \geq s_{\mathsf{m}}.$$

According to claim 4.6, $\mathrm{TotalW}(\mathbf{S}_{i+1}\backslash\mathbf{S}_i) \le \min\{s_{\mathsf{m}}, v_{i+1} - v_i\}$ and $\mathbf{S}_i \subseteq \mathbf{S}_{i+1}$ and $v_i - v_{i-1} \ge 0$ hold for all the i. So we have

$$\begin{aligned}\mathrm{TotalW}(\mathbf{S}_n\backslash\mathbf{S}_{n'}) \le& \sum_{i=N(r-d)+1}^{N(r+1)} \mathrm{TotalW}(\mathbf{S}_i\backslash\mathbf{S}_{i-1}) \\ \le& \sum_{i=N(r-d)+1}^{N(r+1)} \min\{s_{\mathsf{m}}, v_i - v_{i-1}\} \\ \le& \min\{s_{\mathsf{m}}, v_{N(r+1)} - v_{N(r-d)}\}\end{aligned}$$

Thus we claim $v_{N(r+1)} - v_{N(r-d)} \ge s_{\mathsf{m}}$ holds if $S_r^{\mathsf{M}} = 1$. Recalling that v_i is non-decreasing in terms of i, we have

$$\begin{aligned}\sum_{i=r_1+d}^{r_2-1} S_i^{\mathsf{M}} \le& \left(\sum_{i=r_1+d}^{r_2-1} v_{N(i+1)} - v_{N(i-d)}\right)/s_{\mathsf{m}} \\ \le& (d+1)/s_{\mathsf{m}} \cdot \left(v_{N(r_2)} - v_{N(r_1)}\right)\end{aligned}$$

The rest part $\sum_{i=r_1}^{r_1+d-1} S_i^{\mathsf{M}} \le d$ holds trivially. Thus

$$(v_{N(r_2)} - v_{N(r_1)})/s_{\mathsf{m}} \ge \sum_{i=r_1}^{r_2-1} S_i^{\mathsf{M}}/(d+1) - 1.$$

□

Lemma B.4 *For any given round $r_1 < r_2$, we have*

$$\sum_{i=N(r_1)+1}^{N(r_2)} -\Delta_{\mathsf{T}}(\mathcal{S}_{i-1}, e_i)/s_{\mathsf{m}} \ge \sum_{j=r_1}^{r_2-1} S_j^{\mathsf{H2}}/(d+1) - 1.$$

Proof. For any round r with $S_r^{\mathsf{H2}} = 1$, we have adversary state $\mathcal{S}_n$ with $N(r) < n \le N(r+1)$ such that $\mathbf{B}_n^{\Delta}$ contain at least three honest blocks with block weight η_{w}. Suppose event $e_{n'}$ be the latest $\mathsf{hGenRls}$ event of these three blocks. So $\mathbf{B}_{n'-1}^{\Delta}$ contain at least two honest blocks with block weight η_{w}. Thus $\Delta_{\mathsf{T}}(\mathbf{B}_{n'-1}^{\Delta}, e_{n'}) = -s_{\mathsf{m}}$. Since all the blocks in $\mathbf{B}_n^{\Delta}$ should be generated no earlier than round $r-d+1$, we have $n' > N(r-d)$. Recalling that $\Delta_{\mathsf{T}}(\mathcal{S}_{i-1}, e_i) \le 0$ for all i, we have

$$S_r^{\mathsf{H2}} \le \sum_{i=N(r-d)+1}^{N(r+1)} -\Delta_{\mathsf{T}}(\mathcal{S}_{i-1}, e_i)/s_{\mathsf{m}}.$$

Thus we have

$$\begin{aligned}\sum_{i=r_1}^{r_2-1} S_i^{\mathsf{H2}} =& \sum_{i=r_1}^{r_1+d-1} S_i^{\mathsf{H2}} + \sum_{i=r_1+d}^{r_2-1} S_i^{\mathsf{H2}} \\ \le& d + \sum_{i=r_1+d}^{r_2-1} \sum_{j=N(i-d)+1}^{N(i+1)} -\Delta_{\mathsf{T}}(\mathcal{S}_{i-1}, e_i)/s_{\mathsf{m}} \\ \le& d + (d+1) \cdot \sum_{i=N(r_1)+1}^{N(r_2)} -\Delta_{\mathsf{T}}(\mathcal{S}_{i-1}, e_i)/s_{\mathsf{m}}.\end{aligned}$$

□

Lemma B.5 *For any round $r_1 < r_2$, let $u := (\eta_{\mathsf{w}} - 2s_{\mathsf{m}} - 2s_{\mathsf{h}})/\eta_{\mathsf{w}}$, $p_1(t) = \exp\left(\frac{(e^{t\eta_{\mathsf{w}}}-1)\cdot\beta m}{\eta_{\mathsf{w}}\eta_{\mathsf{d}}}\right)$, $p_2(t) = \exp\left(\frac{(e^{-tu\eta_{\mathsf{w}}}-1)\cdot(1-\beta)m}{\eta_{\mathsf{w}}\eta_{\mathsf{d}}}\right)$, $p_3(t) = \exp(-ts_{\mathsf{m}}/(d+1))$ and $p(t) := p_1(t)\cdot\max\{p_2(t), p_3(t)\}$. For any round r, let $X_r := M_r + H_r - s_{\mathsf{m}}/(d+1)\cdot S_r$. For any $t > 0$, any View_r and any $k \in \mathbb{R}$, we have*

$$\Pr\left[\sum_{i=r_1}^{r_2-1} X_i \geq k \middle| \mathsf{View}_{r_1}\right] \leq p(t)^{r_2-r_1}/e^{tk}.$$

and

$$\Pr\left[\sum_{i=r_1}^{r_2-1} M_i \geq k \middle| \mathsf{View}_{r_1}\right] \leq p_1(t)^{r_2-r_1}/e^{tk}.$$

Proof. Let View'_r denote the joint view in View before the phase 3 of round r. For any $r \geq r_1$ and $t > 0$, we have the following discussion.

In phase 3 of round r, suppose the adversary queries the oracle $\mathsf{H}(\cdot)$ x_1 times with the block $\mathbf{b}_{\mathsf{new}}$ satisfying $\mathrm{Adapt}(\mathbf{b}_{\mathsf{new}}.\mathsf{past}) = \mathsf{False}$ and queries the oracle x_2 times the block $\mathbf{b}_{\mathsf{new}}$ satisfying $\mathrm{Adapt}(\mathbf{b}_{\mathsf{new}}.\mathsf{past}) = \mathsf{True}$. (Note that $\mathbf{b}_{\mathsf{new}}$ is not a valid block. But its past set $\mathbf{b}_{\mathsf{new}}.\mathsf{past}$ is fixed before querying random oracle.) Since we only allow adversary control βm malicious nodes, $x_1 + x_2 \leq \beta m$.

For any given x_1, x_2 and View', let A_i ($i \in [x_1]$) and B_j ($j \in [x_2]$) denote the block weight if the adversary finds a solution to make the block valid and equals to 0 otherwise. So

$$M = \sum_{i\in[x_1]} A_i + \sum_{j\in[x_2]} B_j.$$

B_j corresponds to the queries satisfying $\mathrm{Adapt}(\mathbf{b}_{\mathsf{new}}.\mathsf{past}) = \mathsf{True}$. So $B_j = \eta_{\mathsf{w}}$ with probability $1/(\eta_{\mathsf{d}}\eta_{\mathsf{w}})$ and $B_i = 0$ otherwise. Thus $E\left[e^{tB_j}|\mathsf{View}'_r, x_1, x_2\right] = (\eta_{\mathsf{d}}\eta_{\mathsf{w}} - 1 + e^{-t\eta_{\mathsf{w}}})/(\eta_{\mathsf{d}}\eta_{\mathsf{w}})$.

A_i corresponds to the queries satisfying $\mathrm{Adapt}(\mathbf{b}_{\mathsf{new}}.\mathsf{past}) = \mathsf{False}$. So $A_i = 1$ with probability $1/\eta_{\mathsf{d}}$ and $A_i = 0$ otherwise. Thus $E\left[e^{tA_i}|\mathsf{View}'_r, x_1, x_2\right] = (\eta_{\mathsf{d}} - 1 + e^{-t})/\eta_{\mathsf{d}}$. According to AM-GM inequality, $e^{-t\eta_{\mathsf{w}}}/\eta_{\mathsf{w}} + (\eta_{\mathsf{w}} - 1)/\eta_{\mathsf{w}} \geq e^{-t}$. Thus $(\eta_{\mathsf{d}} - 1 + e^{-t})/\eta_{\mathsf{d}} \leq (\eta_{\mathsf{d}}\eta_{\mathsf{w}} - 1 + e^{-t\eta_{\mathsf{w}}})/(\eta_{\mathsf{d}}\eta_{\mathsf{w}})$.

Since all the A_i and B_j are independent conditioned on View'_r, we have

$$E\left[e^{tM_r}\middle|\mathsf{View}'_r\right] \leq \left((\eta_{\mathsf{d}}\eta_{\mathsf{w}} - 1 + e^{t\eta_{\mathsf{w}}})/(\eta_{\mathsf{d}}\eta_{\mathsf{w}})\right)^{x_1+x_2}.$$

Notice that $(\eta_{\mathsf{d}}\eta_{\mathsf{w}} - 1 + e^{t\eta_{\mathsf{w}}})/(\eta_{\mathsf{d}}\eta_{\mathsf{w}}) \leq \exp((e^{t\eta_{\mathsf{w}}} - 1)/(\eta_{\mathsf{d}}\eta_{\mathsf{w}}))$, recalling $x_1 + x_2 \leq \beta m$, we have

$$E\left[e^{tM_r}\middle|\mathsf{View}'_r\right] \leq \exp((e^{t\eta_{\mathsf{w}}} - 1)\beta m/(\eta_{\mathsf{d}}\eta_{\mathsf{w}})) = p_1(t).$$

M_r is the sum of block weight for malicious blocks generated in round r. If $S_r = 0$, then H_r is $-u$ times the sum of block weight for honest blocks generated in round r. And the sum of block weight is independent with S_r, Similar with the previous claim, for all the View'_r and $t > 0$,

$$E\left[e^{tH_r}\middle|\mathsf{View}'_r, S_r = 0\right] \leq \exp((e^{-tu\eta_{\mathsf{w}}} - 1)(1-\beta)m/(\eta_{\mathsf{d}}\eta_{\mathsf{w}})) = p_2(t).$$

If $S_r = 1$, we have $H_r \leq 0$. Thus

$$E\left[e^{t(H_r - s_{\mathsf{m}}/(d+1)\cdot S_r)}\middle|\mathsf{View}'_r\right] \leq \max\{p_2(t), p_3(t)\}.$$

Since M_r, H_r, S_r are independent under given View'_r, thus for any given $r \geq r_1$, View'_r and $t > 0$

$$E\left[e^{tX_r}\middle|\mathsf{View}'_r\right] \leq p_1(t)\cdot\max\{p_2(t), p_3(t)\} = p(t).$$

When $r \geq r_1$, View_{r_1} and X_i for $i \in [r_1, r)$ are determined only by View'_r. So for any View_{r_1}, $r \geq r_1$, X_i for $i \in [r_1, r)$ and $t > 0$, we have

$$E\left[e^{tX_r}\middle|e^{t\sum_{i=r_1}^{r-1} X_i}, \mathsf{View}_{r_1}\right] \leq p(t).$$

Thus

$$E\left[e^{t\sum_{i=r_1}^{r} X_i}\middle|\mathsf{View}_{r_1}\right] \le p(t)\cdot E\left[e^{t\sum_{i=r_1}^{r-1} X_i}\middle|\mathsf{View}_{r_1}\right].$$

By induction, we have

$$E\left[e^{t\sum_{i=r_1}^{r_2} X_i}\middle|\mathsf{View}_{r_1}\right] \le p(t)^{r_2-r_1}.$$

According to the Markovs inequality, for any $t>0$, any View_r and any $k\in\mathbb{R}$, we have

$$\Pr\left[\sum_{i=r_1}^{r_2-1} X_i \ge k\middle|\mathsf{View}_{r_1}\right] \le p(t)^{r_2-r_1}/e^{tk}.$$

Similarly, since we have $E\left[e^{tM_r}\middle|\mathsf{View}'_r\right]\le p_1(t)$, thus

$$\Pr\left[\sum_{i=r_1}^{r_2-1} M_i \ge k\middle|\mathsf{View}_{r_1}\right] \le p_1(t)^{r_2-r_1}/e^{tk}.$$

□

Lemma B.6 *For any given round $r_1<r_2$, let $q:=(1-\beta)md/(\eta_{\mathsf{w}}\eta_{\mathsf{d}})$, $\mathrm{B}(n,p)$ denote the binomial distribution with experiment times n and success probability p, Y_1 follows the probability distribution $\mathrm{B}((1-\beta)m(r_2-r_1),1/(\eta_{\mathsf{w}}\eta_{\mathsf{d}}))$. For any View_{r_1}, $k_2\in\mathbb{N}$ and $y\in\mathbb{N}$, we have*

$$\Pr\left[\sum_{i=r_1}^{r_2-1} F_i \ge k_2\cdot(2\eta_{\mathsf{w}}-2s_{\mathsf{h}}-s_{\mathsf{m}})\middle|\mathsf{View}_{r_1}\right] \le \Pr[Y_1\ge y+1] + \Pr_{Y_2\sim\mathrm{B}(y,q)}[Y_2\ge k_2].$$

Proof. We refer the blocks satisfy $\mathcal{H}^{\mathsf{weight}}(\mathbf{b}.\mathsf{digest}) \ge 2^{\kappa}/\eta_{\mathsf{w}}$ as *tagged blocks* in this proof. For any given $y\in\mathbb{N}$, let n_j ($j\in[y]$) denote the index of the j^{th} $\mathsf{hGenRls}$ event of honest tagged block $\mathbf{b}$ no earlier than round r_1. Let r'_j denote the round index of event n_j.

Recalling that only a tagged block can have block weight η_{w}. If $r'_{y+1}\ge r_2$, we claim for any $r_1\le i<r_2$, $F_i=2\eta_{\mathsf{w}}-2s_{\mathsf{h}}-s_{\mathsf{m}}$ only if there exists n_j such that $r'_j-r'_{j-1}<d$. (We set $r'_0=r_1$.) So we have $\sum_{i=r_1}^{r_2-1}F_i\ge k_2\cdot(2\eta_{\mathsf{w}}-2s_{\mathsf{h}}-s_{\mathsf{m}})$ only if there are at least k_2 different $j\in[y]$ satisfy $r'_j-r'_{j-1}<d$. For any $j\in[y]$, $r'_j-r'_{j-1}<d$ only if the honest nodes queries oracle $\mathsf{H}(\cdot)$ at least $(1-\beta)\cdot dm$ times between event event e_{n_j} (included) and event $e_{n_{j-1}}$ (excluded). It will happens with probability no more than $1-(1-1/(\eta_{\mathsf{w}}\eta_{\mathsf{d}}))^{(1-\beta)md}\le(1-\beta)md/(\eta_{\mathsf{w}}\eta_{\mathsf{d}})=q$ and independent among different j. So we have

$$\Pr\left[r'_{y+1}\ge r_2\wedge\sum_{i=r_1}^{r_2-1}F_i\ge k_2\cdot(2\eta_{\mathsf{w}}-2s_{\mathsf{h}}-s_{\mathsf{m}})\middle|\mathsf{View}_{r_1}\right] \le \Pr_{Y_2\sim\mathrm{B}(y,q)}[Y_2\ge k_2].$$

Notice that $r'_{y+1}\ge r_2$ represents the honest nodes generate at most y tagged blocks between round r_1 (included) and round r_2-1 (included). The honest blocks query oracle $\mathsf{H}(\cdot)$ $(1-\beta)\cdot m(r_2-r_1)$ times, and find a valid block satisfies this property with probability $1/(\eta_{\mathsf{w}}\eta_{\mathsf{d}})$ in each query. Thus $r'_{y+1}<r_2$ holds with probability

$$\Pr_{Y_1\sim\mathrm{B}((1-\beta)m(r_2-r_1),1/(\eta_{\mathsf{w}}\eta_{\mathsf{d}}))}[Y_1\ge y+1].$$

Applying the union bound, we have

$$\Pr\left[\sum_{i=r_1}^{r_2-1}F_i\ge k_2\cdot(2\eta_{\mathsf{w}}-2s_{\mathsf{h}}-s_{\mathsf{m}})\middle|\mathsf{View}_{r_1}\right] \le \Pr_{Y_1}[Y_1\ge y+1] + \Pr_{Y_2\sim\mathrm{B}(y,q)}[Y_2\ge k_2].$$

□

Lemma B.7 *For any given round $r_1 < r_2$, let $\mathrm{B}(n,p)$ denote the binomial distribution with experiment times n and success probability p. For any $y \in \mathbb{Z}^+$, let $d_y := \lceil d/y \rceil$, $Z_1 \sim \mathrm{B}((1-\beta)m(y+1)d_y, 1/\eta_{\mathsf{d}})$, $q := \Pr[Z_1 \geq s_{\mathsf{h}}]$, $Z_2 \sim \mathrm{B}(\lceil (r_2-r_1)/(d_y(y+1)) \rceil, q)$. For any $k_3 \in \mathbb{N}$ and View_{r_1}, we have*

$$\Pr\left[\sum_{i=r_1}^{r_2-1} S_i^{\mathsf{H1}} \geq (y+1)\cdot(k_3+1)\cdot d_y \middle| \mathsf{View}_{r_1}\right] \leq (y+1)\cdot \Pr[Z_2 \geq k_3].$$

Proof. We divide the rounds in $[r_1, r_2)$ into several intervals with length d_y except the last one. The j^{th} interval is $[r_1 + (j-1)\cdot d_y, r_1 + j\cdot d_y)$. Let $T_j = 1$ denote if there exists a round i with $S_i^{\mathsf{H1}} = 1$ in the j^{th} interval and $T_j = 0$ for other cases. Thus

$$\sum_{i=r_1}^{r_2-1} S_i^{\mathsf{H1}} \leq \sum_{j=1}^{\lceil (r_2-r_1)/d_y \rceil} d_y \cdot T_j.$$

Since $S_i^{\mathsf{H1}} = 1$ only if honest nodes generates at least s_{h} blocks in round $(i-d, i]$. So we claim $T_j = 1$ ($j \geq y+1$) only if honest nodes generates at least s_{h} blocks in round $[r_1 + (j-y-1)\cdot d_y, r_1 + j\cdot d_y)$. Since the honest nodes will query oracle $\mathsf{H}(\cdot)$ in $(1-\beta)m(y+1)\cdot d_y$ times during round $[r_1 + (j-y-1)\cdot d_y, r_1 + j\cdot d_y)$, they will generate at least s_{h} honest blocks with probability

$$q = \Pr_{Z_1 \sim \mathrm{B}((1-\beta)m(y+1)\cdot d_y, 1/\eta_{\mathsf{d}})}[Z_1 \geq s_{\mathsf{h}}].$$

Notice that T_{j_2} and T_{j_1} are independent event if $j_2 - j_1 \geq y+1$. So for any $j' \in \{0, 1, \cdots, y\}$, we construct a list of random variables for all the T_j with $j \in [y+1, r_2 - r_1)$ and j modulo $y+1$ equals to j'. The list contains at most $\lceil (r_2-r_1)/(d_y(y+1)) \rceil$ independent random variables. So the sum of these random variables is lager or equal k with probability $\Pr[Z_2 \geq k_3 + 1]$. Taking the union bound, we have

$$\Pr\left[\sum_{j=y+1}^{\lceil (r_2-r_1)/d_y \rceil} T_j \geq (y+1)k_3 \middle| \mathsf{View}_{r_1}\right] \leq (y+1)\Pr[Z_2 \geq k_3]$$

Since $\sum_{j=1}^{y} T_j < y+1$ holds trivially, we have

$$\Pr\left[\sum_{i=r_1}^{r_2-1} S_i^{\mathsf{H1}} \geq (y+1)\cdot(k_3+1)\cdot d_y \middle| \mathsf{View}_{r_1}\right] \leq (y+1)\cdot \Pr[Z_2 \geq k_3].$$

□

B.3 Summarize all the Components

Now we summarize the previous lemmas. First we define a function $\phi(\lambda, \beta, \vec{\eta}, r_\Delta, \rho, s_{\mathsf{m}}, s_{\mathsf{h}}, \vec{t})$. $\vec{t}$ is a tuple of four parameters in $\mathbb{R}^+$, which are denoted by t_1, t_2, t_3, t_4. λ equals to $m(d+1)/\eta_{\mathsf{d}}$ and $r_\Delta \in \mathbb{R}^+$.

$$\begin{aligned}
f(\mu, n) &:= \frac{e^{n-\mu}}{(n/\mu)^n} \\
x_1 &:= e^{t_1\eta_{\mathsf{w}}} - 1 \\
x_2 &:= e^{t_1(2s_{\mathsf{h}} + 2s_{\mathsf{m}} - \eta_{\mathsf{w}})} - 1 \\
k &:= \rho - (t_3+1)\cdot(2\eta_{\mathsf{w}} - 2s_{\mathsf{h}} - s_{\mathsf{m}}) - 1.02\cdot s_{\mathsf{m}}\cdot(t_4+2) - 2s_{\mathsf{m}} \\
q_1 &:= \exp(r_\Delta \cdot \lambda\left(x_1\beta + x_2(1-\beta)\right)/\eta_{\mathsf{w}} - t_1 k) \\
q_2 &:= \exp(r_\Delta \cdot (\lambda x_1 \beta/\eta_{\mathsf{w}} - t_1 s_m) - t_1 k) \\
q_3 &:= f(r_\Delta \cdot (1-\beta)\lambda/\eta_{\mathsf{w}}, t_2) + f(t_2 \cdot (1-\beta)\lambda/\eta_{\mathsf{w}}, t_3) \\
x_3 &:= f(1.02\cdot\lambda(1-\beta), s_{\mathsf{h}}) \\
q_4 &:= 101 \cdot f((r_\Delta + 1)\cdot x_3, t_4) \\
\phi(\lambda, \beta, \vec{\eta}, r_\Delta, \rho, s_{\mathsf{m}}, s_{\mathsf{h}}, \vec{t}) &:= \max\{q_1, q_2\} + q_3 + q_4
\end{aligned} \tag{17}$$

Lemma B.8 *For any given round $r_1 < r_2$, let $r_\Delta := (r_2 - r_1)/(d+1)$, $\lambda = m(d+1)/\eta_{\mathsf{d}}$. When $d > 10^4$, for arbitrary $s_{\mathsf{m}} \geq 0$, $s_{\mathsf{h}} \geq 0$ and positive parameters in $\vec{t}$, we have*

$$\Pr\left[\sum_{i=N(r_1)+1}^{N(r_2)} \Delta(\mathcal{S}_{i-1}, e_i) - \left(v_{N(r_2)} - v_{N(r_1)}\right) \geq \rho \middle| \mathsf{View}_{r_1}\right] \leq \phi(\lambda, \beta, \vec{\eta}, r_\Delta, \rho, s_{\mathsf{m}}, s_{\mathsf{h}}, \vec{t}).$$

Proof. This proof inherits symbols in equation (17).

Notice that $m/\eta_{\mathsf{d}} \cdot (r_2 - r_1) = \lambda \cdot r_\Delta$, for any round r, let $X_r := M_r + H_r - s_{\mathsf{m}}/(d+1) \cdot S_r$. According to lemma B.5, we have

$$\Pr\left[\sum_{i=r_1}^{r_2-1} X_i \geq k \middle| \mathsf{View}_{r_1}\right] \leq \max\{q_1, q_2\}. \tag{18}$$

If random variable T is in binomial distribution $\mathrm{B}(n,p)$, for any $z > np$, according to chernoff bound, we have

$$\Pr[T \geq z] \leq \frac{e^{z/(np)-1}}{(z/(np))^{z/(np)}} = f(np, z).$$

According to lemma B.6, (let variable k_2 in lemma B.6 equals to $\lceil t_3 \rceil$ and variable y in it equals to $\lfloor t_2 \rfloor$), we have

$$\Pr\left[\sum_{i=r_1}^{r_2-1} F_i \geq (t_3+1) \cdot (2\eta_{\mathsf{w}} - 2s_{\mathsf{h}} - s_{\mathsf{m}}) \middle| \mathsf{View}_{r_1}\right] \leq q_3. \tag{19}$$

Let variable y in lemma B.7 equals to 100 and variable k_3 in it equals to $\lceil t_4 \rceil$. Since $d > 10^4$, we have $\lceil d/100 \rceil \cdot (100+1) \leq 1.02(d+1)$. According to lemma B.7,

$$\Pr\left[\sum_{i=r_1}^{r_2-1} S_i^{\mathsf{H1}} \geq 1.02 \cdot (t_4+2) \cdot (d+1) \middle| \mathsf{View}_{r_1}\right] \leq q_4 \tag{20}$$

Summarize the previous lemmas, we can link the random variables to the summation of event values.

$$\begin{aligned}
&\sum_{i=N(r_1)+1}^{N(r_2)} \Delta(\mathcal{S}_{i-1}, e_i) \\
\leq &\sum_{i=N(r_1)+1}^{N(r_2)} \left(\Delta_{\mathsf{M}}(\mathcal{S}_{i-1}, e_i) + \Delta_{\mathsf{H}}(\mathcal{S}_{i-1}, e_i) + \Delta_{\mathsf{F}}(\mathcal{S}_{i-1}, e_i) + \Delta_{\mathsf{T}}(\mathcal{S}_{i-1}, e_i)\right) && \text{Lemma B.1} \\
\leq &\sum_{i=r_1}^{r_2-1} \left(M_i + H_i + F_i - \frac{s_{\mathsf{m}}}{d+1} \cdot S_i^{\mathsf{H2}}\right) + s_{\mathsf{m}} && \text{Eq. (16) \& Lemma B.4} \\
\leq &\sum_{i=r_1}^{r_2-1} \left(X_i + F_i + \frac{s_{\mathsf{m}}}{d+1} \cdot \left(S_i^{\mathsf{M}} + S_i^{\mathsf{H1}}\right)\right) + s_{\mathsf{m}} && \text{Claim B.2} \\
\leq &\sum_{i=r_1}^{r_2-1} \left(X_i + F_i + \frac{s_{\mathsf{m}}}{d+1} \cdot S_i^{\mathsf{H1}}\right) + 2s_{\mathsf{m}} + \left(v_{N(r_2)} - v_{N(r_1)}\right) && \text{Lemma B.3}
\end{aligned}$$

Since $\rho = k + (t_3+1) \cdot (2\eta_{\mathsf{w}} - 2s_{\mathsf{h}} - s_{\mathsf{m}}) + 1.02 \cdot (t_4+2) + 2s_{\mathsf{m}}$, according to equations 18, 19 and 20, for the parameters $s_{\mathsf{m}} \geq 0$, $s_{\mathsf{h}} \geq 0$ and t_1, t_2, t_3, t_4, we claim

$$\Pr\left[\sum_{i=N(r_1)+1}^{N(r_2)} \Delta(\mathcal{S}_{i-1}, e_i) - \left(v_{N(r_2)} - v_{N(r_1)}\right) \geq \rho \middle| \mathsf{View}_{r_1}\right] \leq \max\{q_1 + q_2\} + q_3 + q_4.$$

□

We define function $\tilde{\phi}(\lambda, \beta, \vec{\eta}, r_\Delta, \rho)$ as the minimum value that $\phi(\lambda, \beta, \vec{\eta}, r_\Delta, \rho, s_\mathsf{m}, s_\mathsf{h}, \vec{t})$ can achieve by picking parameters $s_\mathsf{h}, s_\mathsf{m}$ and $\vec{t}$. Then we have the following lemma

Lemma B.9 *When $\lambda \geq 0.8\log(500/\delta)$ and $\eta_\mathsf{w} = 30\lambda/\delta$, for any $\varepsilon > 0$, if*

$$r_\Delta \geq \max\left\{(3+\rho/\eta_\mathsf{w})\cdot\frac{600}{\delta^2}, \log\left(\frac{4}{\varepsilon}\right)\cdot\frac{3000}{\delta^3}, \log\left(\frac{404}{\varepsilon}\right)\cdot\frac{200}{\delta}\right\},$$

then

$$\tilde{\phi}(\lambda, \beta, \vec{\eta}, r_\Delta, \rho) \leq \varepsilon.$$

Proof. In this proof, we try to find a feasible solution for $\phi(\lambda, \beta, \vec{\eta}, r_\Delta, -\rho, s_\mathsf{m}, s_\mathsf{h}, \vec{t}) < \varepsilon$. This proof inherits symbols in equation (17). Let $s_\mathsf{m} = 1.5\lambda$, $s_\mathsf{h} = 3\lambda$ and $t_1 = \delta^2/(150\lambda)$. Then we have $x_1 = e^{\delta/5} - 1$ and $x_2 = e^{-\delta/5\cdot(1-7\delta/10)} - 1 \leq (e^{-\delta/5} - 1)\cdot(1 - 7\delta/10)$. Thus

$$\begin{aligned}
&\lambda\left(x_1\beta + x_2(1-\beta)\right)/\eta_\mathsf{w} \\
=&\frac{\delta}{30(2-\delta)}\cdot\left((e^{\delta/5}+e^{-\delta/5}-2)\cdot(1-7\delta/10) - 0.7\delta\cdot(e^{\delta/5}-1)\right) \\
\leq&\frac{\delta}{30(2-\delta)}\cdot\left(\frac{\delta^2}{25} - \frac{7\delta^2}{50}\right) \\
\leq&-\frac{\delta^3}{600}
\end{aligned}$$

So

$$q_1 \leq \exp(-r_\Delta\cdot\delta^3/600 - t_1\cdot k_1).$$

Since it can be verified that $\lambda x_1\beta/\eta_\mathsf{w} \leq \frac{\delta^2}{300}$ and $-t_1 * s_\mathsf{m} = -\frac{\delta^2}{100}$, so we have

$$q_2 \leq \exp(-r_\Delta\cdot\delta^2/150 - t_1\cdot k_1).$$

Let $t_2 = \sqrt{2}\cdot r_\Delta\cdot(1-\beta)\lambda/\eta_\mathsf{w}$ and $t_3 = \sqrt{2}\cdot(1-\beta)\lambda/\eta_\mathsf{w}\cdot t_2$, we skim the detailed computation and claim

$$q_3 < 2\exp(-r_\Delta\cdot(1-\beta)^2\delta^2/450) < 2\exp(-r_\Delta\cdot\delta^2/1800).$$

Notice that $s_\mathsf{h} \geq 3\cdot\lambda$. Thus

$$x_3 \leq f(1.02\cdot\lambda\cdot(1-\beta), 3\cdot\lambda) < \exp(-1.25\lambda) = \frac{\delta}{500}.$$

Let $t_4 = 4\cdot\delta/500\cdot(r_\Delta+1)$, notice that $r_\Delta \geq \log(404)\cdot 200 > 1000$, thus $r_\Delta + 1 < 1.001 r_\Delta$. So we have

$$q_4 < 101\cdot\exp(-r_\Delta\cdot\delta/200).$$

Notice that $t_3 = 2(1-\beta)^2\lambda^2/\eta_\mathsf{w}^2\cdot r_\Delta \leq \delta^2/450\cdot r_\Delta$ and $t_4 < \delta/120$. We have

$$\begin{aligned}
k_1 &\geq -\rho - 2\eta_\mathsf{w}\cdot t_3 - 1.02 s_\mathsf{m} t_4 - 3.04 s_\mathsf{m} - 2\eta_\mathsf{w} \\
&\geq -\rho - 2.26\eta_\mathsf{w} - 2\eta_\mathsf{w}\cdot t_3 - 1.02 s_\mathsf{m} t_4 \\
&> -\rho - 2.26\eta_\mathsf{w} - r_\Delta\cdot\lambda(2\delta/15 + \delta/60) \\
&> -\rho - 2.26\eta_\mathsf{w} - r_\Delta\cdot\lambda\cdot 3\delta/20
\end{aligned}$$

Applying the lower bound of k_1 to q_1 and q_2, we can get

$$\begin{aligned}
q_1 &< \exp(-r_\Delta\cdot\delta^3/1500 + t_1(\rho + 3\eta_\mathsf{w})) \\
q_2 &< \exp(-r_\Delta\cdot 17\delta^3/3000 + t_1(\rho + 3\eta_\mathsf{w}))
\end{aligned}$$

When $r_\Delta \geq (3+\rho/\eta_{\mathsf{w}}) \cdot \frac{600}{\delta^2}$, we have $t_1(\rho+3\eta_{\mathsf{w}}) \leq r_\Delta \cdot \delta^3/3000$, thus $q_1 < \exp(-r_\Delta \cdot \delta^3/3000)$ and $q_2 < \exp(-r_\Delta \cdot 16\delta^3/3000)$. Since $r_\Delta \geq \log\left(\frac{4}{\varepsilon}\right) \cdot \frac{3000}{\delta^3}$, we have $q_1 < \varepsilon/4$, $q_2 < \varepsilon/4$ and $q_3 < \varepsilon/2$. Since $r_\Delta \geq \log\left(\frac{404}{\varepsilon}\right) \cdot \frac{200}{\delta}$, $q_4 < \varepsilon/4$. Thus we have

$$\phi(\lambda, \beta, \vec{\eta}, r_\Delta, -\rho, s_{\mathsf{m}}, s_{\mathsf{h}}, \vec{t}) < \varepsilon.$$

□

B.4 Proof of Theorem 4.11

Proof. For any given round $r_1 < r_2$, let $r_\Delta := (r_2 - r_1)/(d+1)$. According to lemma B.8, we have

$$\Pr\left[\sum_{i=N(r_1)+1}^{N(r_2)} \Delta(\mathcal{S}_{i-1}, e_i) - \left(v_{N(r_2)} - v_{N(r_1)}\right) \geq \rho \middle| \mathsf{View}_{r_1}\right] \leq \tilde{\phi}(\lambda, \beta, \vec{\eta}, r_\Delta, \rho).$$

According to lemma B.9, when $\lambda \geq 0.8\log(500/\delta)$, $\eta_{\mathsf{w}} = 30\lambda/\delta$ and

$$r_\Delta \geq \max\left\{(3+\rho/\eta_{\mathsf{w}}) \cdot \frac{600}{\delta^2}, \log\left(\frac{4}{\varepsilon}\right) \cdot \frac{3000}{\delta^3}, \log\left(\frac{404}{\varepsilon}\right) \cdot \frac{200}{\delta}\right\},$$

we have

$$\tilde{\phi}(\lambda, \beta, \vec{\eta}, r_\Delta, \rho) \leq \varepsilon.$$

So this theorem is proved. □

C Timer Chain and Old Enough Blocks

Lemma C.1 *Let* $\mathbf{B}_{n,r} := \left\{\mathbf{b} \in \mathbf{B}_n^{\mathsf{gen}} \middle| \mathbf{B}_{N(r)}^{\mathsf{min}} \nsubseteq \mathbf{b}.\mathsf{past} \vee \mathbf{b} \in \mathbf{B}_{N(r)}^{\mathsf{min}}\right\}$. *For any round* r *and* $\gamma \geq 0$, *if* $\beta \geq 0.1$, $\eta_{\mathsf{t}} \geq 2\lambda/\delta$ *and* r_Δ *satisfying*

$$r_\Delta \geq \frac{\eta_{\mathsf{t}}\eta_{\mathsf{d}}}{m} \cdot \max\left\{\frac{128}{\delta^2} \cdot \log(\frac{8400}{\varepsilon\delta^2}), \frac{8(\gamma+2)}{\delta}\right\}$$

we have

$$\Pr\left[\exists n \geq N(r+r_\Delta), \mathrm{MaxTH}(\mathbf{B}_n^{\mathsf{min}}) - \mathrm{MaxTH}(\mathbf{B}_{n,r}) \leq \gamma\right] \leq \varepsilon.$$

Proof. Let $\delta := 1-\beta/(1-\beta)$, $\tau := \sqrt[4]{(1-\beta)/\beta}$, $z := \lfloor \eta_{\mathsf{t}}\eta_{\mathsf{d}}/(m\beta\tau)\rfloor$, $c := (d+1)\cdot m/(\eta_{\mathsf{t}}\eta_{\mathsf{d}})$ and $f(x) := \frac{e^x}{(1+x)^{1+x}}$. Notice that $\tau \leq 2$ when $\beta \geq 0.1$.

Recalling that $N(r)$ denotes the index of latest event before round r, e_n denotes the n^{th} event and $\mathcal{S}_{n-1}$ denotes the adversary state before event e_n. $\mathbf{B}_n^{\mathsf{gen}}, \mathbf{B}_n^{\mathsf{max}}, \mathbf{B}_n^{\mathsf{min}}$ denote the corresponding component in $\mathcal{S}_n$.

For any given round r and positive integer $k \in \mathbb{Z}^+$ with $k \geq 10$, we define real number r_+ and random variables $X(r,k)$ and $Y(r,k)$ as follows

$$\begin{aligned} r_+ &:= r + z \cdot k \\ X(r,k) &:= \mathrm{MaxTH}(\mathbf{B}_{N(r_+)}^{\mathsf{min}}) - \mathrm{MaxTH}(\mathbf{B}_{N(r)}^{\mathsf{max}}) \\ Y(r,k) &:= \text{the number of malicious timer blocks generated in rounds } [r, r_+). \end{aligned}$$

We try to study probability distribution for $X(r,k)$ and $Y(r,k)$. In an admissible environment, if an honest node generates or receives a block in round r, all the honest nodes will receive such a block before phase 2 of round $r+d$. So if an honest node constructs a timer block $\mathbf{b}_1$ in round r, for any honest timer block $\mathbf{b}_2$ generated no earlier than round $r+d$, it will be $\mathrm{TimerHeight}(\mathbf{b}_2) \geq \mathrm{TimerHeight}(\mathbf{b}_1) + 1$. We construct an event list as the following steps:

1. Find the first $\mathsf{hGenRls}$ event of timer block started with round $r+d$.

2. Skip the subsequent $(1-\beta)md$ queries for oracle $\mathsf{H}(\cdot)$ from honest nodes after this event. (Recall that honest nodes query this oracle $(1-\beta)m$ times in each round.) Then find the next $\mathsf{hGenRls}$ event of timer block.
3. Repeat step 2 until reaching the end of round $r_+ - d - 1$.

We claim the timer height of the first element in this event list will be no less than $\mathrm{MaxTH}(\mathbf{B}^{\mathsf{max}}_{N(r)}) + 1$. And for any two consecutive events, the timer height of the latter one must be strict larger than the timer height of the former one. So $\mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_{N(r_+)}) - \mathrm{MaxTH}(\mathbf{B}^{\mathsf{max}}_{N(r)})$ is no less than the length of this event list.

If $X(r,k) \leq \tau k$, the length of such event list is no larger than τk and at most $(1-\beta)md \cdot \tau k$ queries are skipped when contracting such event list. So it only happens when honest nodes find at most τk timer blocks in $(r_+ - r - (\tau k+2)d+1)(1-\beta)m$ queries. An honest node will find a valid timer block with probability $1/(\eta_{\mathsf{d}}\eta_{\mathsf{t}})$ in each query and the outcomes are independent. Notice that

$$\begin{aligned}
&(r_+ - r - (\tau k+2)d+1)(1-\beta)m \\
\geq &\left(\frac{\eta_{\mathsf{t}}\eta_{\mathsf{d}}}{m\beta\tau} - (\tau k + 0.2k)(d+1)\right)\cdot(1-\beta)m \\
= &(1-\beta)k \cdot \eta_{\mathsf{t}}\eta_{\mathsf{d}} \cdot \left(\frac{1}{\beta\tau} - (\tau+0.2)\cdot c\right)
\end{aligned}$$

It can be verified that $\frac{1/(\beta\tau)-\tau^2/(1-\beta)}{\tau+0.2} < \delta/2$. Since $c > \delta/2$,

$$(1-\beta)k \cdot \eta_{\mathsf{t}}\eta_{\mathsf{d}} \cdot \left(\frac{1}{\beta\tau} - (\tau+0.2)\cdot c\right) > \tau^2 k \eta_{\mathsf{d}}\eta_{\mathsf{t}}.$$

Let W be a random variable with probability distribution $\mathrm{B}(\tau^2 k\eta_{\mathsf{d}}\eta_{\mathsf{t}}, 1/(\eta_{\mathsf{d}}\eta_{\mathsf{t}}))$. According to the chernoff bound (lemma D.1), we have

$$\forall k' < \tau^2 \cdot k, \quad \Pr[X(r,k) \leq k'] \leq \Pr[W \leq \tau \cdot k] \leq f\left(\frac{k'}{\tau^2 \cdot k} - 1\right)^{\tau^2 k}.$$

$Y(r,k) \geq k$ only if the adversary generates at least k timer blocks in round $[r, r_+)$, which implies the adversary finds k timer blocks in $(r_+ - r)\beta m \leq k/\tau$ queries. Similarly, we can get

$$\forall k' > k/\tau, \quad \Pr[Y(r,k) \geq k'] \leq f\left(\frac{\tau \cdot k'}{k} - 1\right)^{k/\tau}.$$

Now we will study the probability that

$$\exists n \geq N(r + r_\Delta), \mathrm{MaxTH}(\mathbf{B}_{n,r}) - \mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_n) \leq \gamma.$$

Let t denote the largest timer height such that no malicious block in $\mathbf{B}^{\mathsf{gen}}_{N(r)}$ has timer height t. So there must be an honest block with timer height t in $\mathbf{B}^{\mathsf{max}}_{N(r)}$. We denote the earliest one by $\mathbf{b}_t$. Notice that all the honest blocks $\mathbf{b}$ generated no earlier than round r satisfy $\mathbf{B}^{\mathsf{min}}_{N(r)} \subseteq \mathbf{b}.\mathsf{past}$ in an admissible environment and thus they can not appear in $\mathbf{B}_{n,r}$ for any $n \geq N(r)$.

Let n_1 be the index of $\mathsf{hGenRls}$ event for $\mathbf{b}_t$ (notice that $n_1 < N(r)$). If there exists $n_2 \geq N(r + r_\Delta)$ such that $\mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_{n_2}) - \mathrm{MaxTH}(\mathbf{B}_{n_2,r}) \leq \gamma$, we find integers k_1 and k_2 which satisfy

$$\begin{aligned}
N(r - z\cdot(k_1-1)) \leq &n_1 < N(r - z\cdot k_1) \\
N(r + z\cdot k_2) \leq &n_2 < N(r + z\cdot(k_2+1))
\end{aligned}$$

Let $k_\Delta = k_1 + k_2$. Since $\tau - 1 > \delta/4$ and $r_\Delta \geq 4\cdot z(\gamma+2)/\delta$, we have $k_2 \geq r_\Delta/z = 4\cdot(\gamma+2)/\delta > (\gamma+2)/(\tau-1)$. Thus

$$\mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_{n_2}) - \mathrm{MaxTH}(\mathbf{B}_{n_2,r}) \leq \gamma < (\tau-1)\cdot k_2 - 2 \leq (\tau-1)\cdot k_\Delta - 2.$$

Then one of the following two inequalities must holds

$$\mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_{n_2}) \le t + \tau \cdot k_\Delta$$
$$\mathrm{MaxTH}(\mathbf{B}_{n_2,r}) \ge t + 2 + k_\Delta$$

For the case $\mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_{n_2}) \le t + \tau \cdot k_\Delta$, let $r_s = r - z \cdot k_1$, $r_e = r + z \cdot k_2$. Then $\mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_{N(r_e)}) \le t + \tau \cdot k_\Delta$ because $N(r_e) < n_2$. Since $\mathbf{b}_t$ is the first timer block with height t and it is generated earlier than round r_s, we have $\mathrm{MaxTH}(\mathbf{B}^{\mathsf{max}}_{N(r_s)}) \ge t$. So it will be

$$X(r - z \cdot k_1, k_\Delta) = \mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_{N(r_e)}) - \mathrm{MaxTH}(\mathbf{B}^{\mathsf{max}}_{N(r_s)}) \le \tau \cdot k_\Delta.$$

For the case $\mathrm{MaxTH}(\mathbf{B}_{n_2}) \ge t + 2 + k_\Delta$, let $r_s = r - z \cdot (k_1 - 1)$, $r_e = r + z \cdot (k_2 + 1)$. Since $\mathbf{b}_t$ is the first timer block with height t and it is generated no earlier than round r_s and $\mathbf{B}_{n_2,r} \subseteq \mathbf{B}_{N(r_e),r}$, the adversary generates at least $2 + k_\Delta$ blocks in rounds $[r_s, r_e)$. It means

$$Y(r + z \cdot (k_1 - 1), k_\Delta + 2) \ge k_\Delta + 2.$$

Notice that k_1, k_2 may be dependent with random variable $X(r,k)$ and $Y(r,k)$. So we take a union bound over all the possible k_1, k_2. Since $r_\Delta > z \cdot \frac{64}{\delta^2} \cdot \log(\frac{8400}{\varepsilon\delta^2})$, we have $k_2 \ge \frac{64}{\delta^2} \cdot \log(\frac{8400}{\varepsilon\delta^2})$. Notice that $1/(1-e^{-x}) < 1.01/x$ for $0 < x < 1/64$ and $\tau \le 2$. It can be verified that $f(\tau - 1) \le \exp(-\delta^2/32)$ and $f(1/\tau - 1) \le \exp(-\delta^2/32)$. Let $y := \frac{64}{\delta^2} \cdot \log(\frac{8400}{\varepsilon\delta^2})$, we have

$$\begin{aligned}
&\Pr\left[\exists n \ge N(r + r_\Delta), \mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_n) - \mathrm{MaxTH}(\mathbf{B}_{n,r}) \le \gamma\right] \\
\le &\sum_{k_1=0}^{\infty} \sum_{k_2=\lceil y \rceil}^{\infty} \left(\Pr[X(r - z \cdot k_1, k_\Delta) \le \tau k_\Delta] + \Pr[Y(r - z \cdot (k_1 - 1), k_\Delta + 2) \ge k_\Delta + 2]\right) \\
= &\sum_{k_1=0}^{\infty} \sum_{k_2=\lceil y \rceil}^{\infty} \left(f(1/\tau - 1)^{\tau^2 k_\Delta} + f(1-\tau)^{(k_\Delta+2)/\tau}\right) \\
\le &\sum_{k_1=0}^{\infty} \sum_{k_2=\lceil y \rceil}^{\infty} \left(\exp\left(-\delta^2/32 \cdot \tau^2 k_\Delta\right) + \exp(-\delta^2/32 \cdot (k_\Delta + 2)/\tau)\right) \\
\le &\sum_{k_1=0}^{\infty} \sum_{k_2=\lceil y \rceil}^{\infty} 2\exp(-\delta^2/64 \cdot (k_1 + k_2)) \\
< &\frac{8400}{\delta^4} \cdot \exp\left(-\delta^2/64 \cdot \frac{64}{\delta^2} \cdot \log\left(\frac{8400}{\varepsilon\delta^2}\right)\right) \\
= &\varepsilon
\end{aligned}$$

Next we will study the probability that

$$\mathrm{MaxTH}(\mathbf{B}_{n,n}) - \mathrm{MaxTH}(\mathbf{B}^{\mathsf{min}}_n) \ge \gamma.$$

Similarly, let t denote the largest timer height such that no malicious block in $\mathbf{B}^{\mathsf{gen}}_n$ has timer height t. So there must be an honest block with timer height t in $\mathbf{B}^{\mathsf{max}}_{N(r)}$. We denote the earliest one by $\mathbf{b}_t$. Notice that all the honest blocks $\mathbf{b}$ generated no earlier than round r satisfy $\mathbf{B}^{\mathsf{min}}_{N(r)} \subseteq \mathbf{b}.\mathsf{past}$ in an admissible environment. So any honest block generated no earlier than round r can not be in $\mathbf{B}_{n,r}$ for any n.

□

C.1 Proof of Theorem 4.10

Proof. Let $\mathbf{B}_{n,r} := \left\{\mathbf{b} \in \mathbf{B}_n^{\mathsf{gen}} \middle| \mathbf{B}_{N(r)}^{\mathsf{min}} \nsubseteq \mathbf{b}.\mathsf{past} \vee \mathbf{b} \in \mathbf{B}_{N(r)}^{\mathsf{min}}\right\}$, $r_\Delta := \frac{\eta_{\mathsf{t}}\eta_{\mathsf{d}}}{m} \cdot \max\left\{\frac{129}{\delta^2} \cdot \log(\frac{9000}{\varepsilon\delta^2}), \frac{8(\eta_{\mathsf{b}}+3)}{\delta}\right\}$ and $r_1 := r_2 - r_\Delta + d + 1$.

All the blocks generated earlier than round $r_1 - d$ belong to $\mathbf{B}_{N(r_1)}^{\mathsf{min}}$. So they also belong to $\mathbf{B}_{N(r_2),r_1}$. Notice that $d + 1 = \lambda\eta_{\mathsf{d}}/m \leq \eta_{\mathsf{t}}\eta_{\mathsf{d}}\delta/(2m)$, thus

$$r_2 - r_1 = r_\Delta - d - 1 \geq \frac{\eta_{\mathsf{t}}\eta_{\mathsf{d}}}{m} \cdot \max\left\{\frac{128}{\delta^2} \cdot \log\left(\frac{9000}{\varepsilon\delta^2}\right), \frac{8(\eta_{\mathsf{b}}+2)}{\delta}\right\}.$$

According to lemma C.1 , we have

$$\Pr\left[\mathrm{MaxTH}(\mathbf{B}_{N(r_2)}^{\mathsf{min}}) - \mathrm{MaxTH}(\mathbf{B}_{N(r_2),r_1}) \leq \eta_{\mathsf{b}}\right] \leq \frac{14}{15} \cdot \varepsilon.$$

Notice that $\mathrm{Old}(\mathbf{B}_{N(r_2)}^{\mathsf{min}}, \mathbf{b}) = \mathsf{False}$ holds only if $\mathrm{MaxTH}(\mathbf{B}_{N(r_2)}^{\mathsf{min}}) - \mathrm{TimerHeight}(\mathbf{b}) < \eta_{\mathsf{b}}$. As long as $\mathrm{MaxTH}(\mathbf{B}_{N(r_2)}^{\mathsf{min}}) - \mathrm{MaxTH}(\mathbf{B}_{N(r_2),r_1}) \leq \eta_{\mathsf{b}}$, all the blocks generated earlier than round $r_1 - d$ will be old enough given $\mathbf{B}_{N(r_2)}^{\mathsf{min}}$. This holds with exception probability $14/15 \cdot \varepsilon$.

By the definition of potential value, for any $\mathcal{S}$ and $\mathbf{b}$, we have $P_{\mathsf{with}}(\mathcal{S}, \mathbf{b}) \leq \mathrm{SubTW}(\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}}, \mathbf{b})$, $P_{\mathsf{adv}}(\mathcal{S}, \mathbf{b}) \leq s_{\mathsf{h}} + s_{\mathsf{m}} \leq \eta_{\mathsf{w}}$ and $P_{\mathsf{sp}}(\mathcal{S}, \mathbf{b}) \leq \mathrm{SubTW}\big(\mathbf{B}^{\Delta} \cap \mathbf{M}, \mathbf{b}\big)$. Notice that $\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}}$ only contains malicious blocks (a.k.a. $\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}} \subseteq \mathbf{M}$) and $\mathbf{B}^{\Delta} = \mathbf{B}^{\mathsf{max}} \backslash \mathbf{B}^{\mathsf{min}}$. We claim $\mathbf{B}^{\mathsf{gen}} \backslash \mathbf{B}^{\mathsf{max}}$ and $\mathbf{B}^{\Delta} \cap \mathbf{M}$ are disjoint sets and there union is subset of or equal to $\mathbf{B}^{\mathsf{gen}} \cap \mathbf{M}$. In all,

$$P(\mathcal{S}, \mathbf{b}) \leq \eta_{\mathsf{w}} + \mathrm{SubTW}(\mathbf{B}^{\mathsf{gen}} \cap \mathbf{M}, \mathbf{b}).$$

It implies that $P(\mathcal{S}_{N(r_2)}, \mathbf{b}) - \eta_{\mathsf{w}}$ is no more than the total weight of malicious blocks after $\mathbf{b}$'s generation. Let M_i denote the total block weight of malicious blocks generated in round i. Then, for any block $\mathbf{b}$ generated no earlier than round $r_1 - d$, for any n with $N(r_2) < n \leq N(r_2 + 1)$, we have

$$P(\mathcal{S}_n, \mathbf{b}) - \eta_{\mathsf{w}} \leq \sum_{i=r_1-d}^{r_2} M_i.$$

Let $p_1(t) := \exp\left(\frac{(e^{t\eta_{\mathsf{w}}}-1)\cdot\beta m}{\eta_{\mathsf{w}}\eta_{\mathsf{d}}}\right)$, for any $t > 0$ and any $k \in \mathbb{R}$, according to lemma B.5, we have

$$\Pr\left[\sum_{i=r_1-d}^{r_2} M_i \geq k\right] \leq p_1(t)^{r_\Delta}/e^{tk}.$$

Let $t := \delta^2/(150\lambda)$, $k := 2\lambda/(d+1) \cdot r_\Delta$, we have

$$p_1(t)^{r_\Delta}/e^{tk} \leq \exp(-\delta^2/150 \cdot r_\Delta/(d+1)).$$

Note that $\eta_{\mathsf{t}}\eta_{\mathsf{d}}/m = 2(d+1)$. Thus $r_\Delta/(d+1) \geq \frac{300}{\delta^2} \cdot \log(\frac{9000}{\varepsilon\delta^2}) > \frac{150}{\delta^2} \cdot \log(\frac{9000}{\varepsilon})$. So we claim

$$\exp(-\delta^2/150 \cdot r_\Delta/(d+1)) \leq \frac{\varepsilon}{9000}.$$

It implies with exception probability $\frac{\varepsilon}{9000}$, for any block $\mathbf{b}$ generated no earlier than round $r_1 - d$, for any n with $N(r_2) < n \leq N(r_2 + 1)$, $P(\mathcal{S}_n, \mathbf{b}) - \eta_{\mathsf{w}} \leq 2\lambda/(d+1) \cdot r_\Delta$ holds for any block $\mathbf{b}$ generated no earlier than round $r_1 - d$ and $N(r_2) < n \leq N(r_2 + 1)$.

Now we have showed that for any block generated earlier than round $r_1 - d$, they all become old enough at the beginning of round r_2 with exception probability $\frac{14\varepsilon}{15}$. For the other blocks, their block potential value will never exceed $2\lambda/(d+1) \cdot r_\Delta + \eta_{\mathsf{w}}$ in round r_2 with exception probability $\frac{\varepsilon}{9000}$. Notice that

$$\begin{aligned}
&2\lambda/(d+1) \cdot r_\Delta + \eta_{\mathsf{w}} \\
=&\frac{30\lambda}{\delta} + 4\lambda \cdot \max\left\{\frac{129}{\delta^2} \cdot \log\left(\frac{9000}{\varepsilon\delta^2}\right), \frac{8(\eta_{\mathsf{b}}+3)}{\delta}\right\} \\
<&4\lambda \cdot \max\left\{\frac{140}{\delta^2} \cdot \log\left(\frac{9000}{\varepsilon\delta^2}\right), \frac{8(\eta_{\mathsf{b}}+4)}{\delta}\right\}
\end{aligned}$$

In all, for any $r_2 \geq 0$ and $\varepsilon > 0$, we have

$$\Pr\left[\exists N(r_2) < n \leq N(r_2+1), \exists \mathbf{b} \in \tilde{\mathbf{B}}_{r_2}, P(\mathcal{S}_n, \mathbf{b}) \geq w(\varepsilon)\right] \leq \frac{14\varepsilon}{15} + \frac{\varepsilon}{9000} < \varepsilon.$$

□

C.2 Proof of Theorem 4.12

Proof. Let $\mathbf{B}_{n,r} := \left\{\mathbf{b} \in \mathbf{B}_n^{\mathsf{gen}} \middle| \mathbf{B}_{N(r)}^{\mathsf{min}} \nsubseteq \mathbf{b}.\mathsf{past} \vee \mathbf{b} \in \mathbf{B}_{N(r)}^{\mathsf{min}}\right\}$. If there exists n and $\mathbf{b}$ satisfying $\mathbf{b} \in \mathbf{B}_n^{\mathsf{gen}}$, $\mathbf{B}_{N(r)}^{\mathsf{min}} \nsubseteq \mathbf{b}.\mathsf{past}$ and $\mathrm{Old}(\mathbf{B}_n^{\mathsf{min}}, \mathbf{b}) = \mathsf{False}$, we claim $\mathbf{b} \in \mathbf{B}_{n,r}$ and $\mathrm{MaxTH}(\mathbf{B}_n^{\mathsf{min}}) - \mathrm{TimerHeight}(\mathbf{b}) \leq \eta_{\mathsf{b}}$. Thus, $\mathrm{MaxTH}(\mathbf{B}_n^{\mathsf{min}}) - \mathrm{TimerHeight}(\mathbf{B}_{n,r}) \leq \eta_{\mathsf{b}}$. According to lemma C.1, it will happen with probability ε. □

D Chernoff bound

Lemma D.1 (Multiplicative Chernoff bound) *Let X be a random variable with binomial distribution $\mathrm{B}(n,p)$, then for any $\delta > 0$, we have*

$$\Pr\left[X \geq (1+\delta)np\right] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{np}.$$

$$\Pr\left[X \leq (1-\delta)np\right] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{np}.$$