Incentive Compatibility of Ethereum's PoS Consensus Protocol

Ulysse Pavloff
Université Paris-Saclay, CEA, LIST, Palaiseau, France

Yackolley Amoussou-Guenou
Université Paris-Panthéon-Assas, CRED, Paris, France

Sara Tucci-Piergiovanni
Université Paris-Saclay, CEA, LIST, Palaiseau, France

Abstract

This paper investigates whether following the fork-choice rule in the Ethereum PoS consensus protocol constitutes a Nash equilibrium – i.e., whether the protocol that maintains the canonical chain in Ethereum is incentive-compatible. Specifically, we explore whether selfish participants may attempt to manipulate the fork-choice rule by forking out previous blocks and capturing the rewards associated with those blocks. Our analysis considers two strategies for participants: the obedient strategy, which adheres to the prescribed protocol, and the cunning strategy, which attempts to manipulate the fork-choice rule to gain more rewards. We evaluate the conditions under which selfish participants might deviate from the obedient strategy. We found that, in a synchronous system, following the prescribed fork-choice rule is incentive-compatible. However, in an eventually synchronous system, the protocol is *eventually incentive-compatible* – that is, only a limited number of proposers will find it profitable to fork the chain during the synchronous period. After this sequence of cunning proposers, subsequent proposers will find it more profitable to follow the protocol.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Computer systems organization → Dependable and fault-tolerant systems and networks

Keywords and phrases Ethereum PoS, Game Theory, Block Reward

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2024.7

1 Introduction

Ethereum, the leading public PoS blockchain, with over a million validator nodes, underwent a transition from PoW to PoS in 2022. The shift aimed to create a protocol resembling Nakamoto consensus, while having a finalization mechanism in parallel that grows an unforkable prefix (i.e. blocks permanently belonging to the chain). The finalization mechanism, also called the Finality Gadget [11], integrates a quorum system seen in Byzantine Fault-Tolerant (BFT) Consensus protocols. The resulting protocol mixing Nakamoto-style consensus to resolve forks and a BFT-like consensus to finalize blocks became quite involved and departed significantly from both Nakamoto-style and classical BFT consensus protocols.

The study of this intricate protocol is an active field of research because the possibility to fork introduces vulnerabilities [14, 22, 24, 25, 26, 29] absent in blockchains maintained by pure Byzantine consensus protocols. Indeed, the main source of vulnerability arises from the fact that, unlike blockchains based on classical BFT consensus protocols (e.g., [7, 8, 10]), the voting procedure to reach a super-majority quorum (a majority of honest votes) for a block is conducted through intermediate voting rounds[1], which are recorded via blocks in the so-called *canonical chain*.

[1] A voting round takes place in a so-called *slot* in the Ethereum protocol.

Specifically, the set of voters is partitioned into distinct, non-overlapping groups, and they vote in turn, with one group voting per round. Voters are called *attesters* in Ethereum and we keep this term in the remainder of the paper. Within each round, a proposer is selected. This proposer is responsible for extending the canonical chain by linking the newly proposed block to the one designated by the fork choice rule. After the new proposer's block, the attesters of the round proceed to vote on the block they deem the head of the canonical chain. In a synchronous system, where votes are always received within the corresponding round, when the last group has voted, a quorum has been reached for the block proposed in the first round. Note that this condition is necessary for finalization[2]. However, so-called balancing attacks [24] have been shown to exploit two consecutive Byzantine proposers to first create a fork, i.e., two conflicting proposals, and then, by manipulating a few votes and introducing network delays (e.g., releasing a few Byzantine votes for one of the proposals to only half of the network), split the attesters into two conflicting canonical chains. The attacker then observes the honest votes and balances Byzantine votes across remaining rounds to keep the two chains in a tie, perpetuating the conditions for the attack and stalling finalization indefinitely.

To mitigate this attack, a mechanism called *proposer boost* has been introduced[3]. The proposer boost adds artificial votes to a proposal that is received early in the round. This boost helps reconcile the network in the event of a balancing attack because a subsequent proposer – assumed to be honest and well-connected – can add additional weight to the canonical chain his block is linked to. If the proposer boost represents a good proportion of the total votes expected in the round, it effectively neutralizes the balancing effect of Byzantine votes on the two competing chains, halting the attack.

In this paper, we investigate whether the introduction of the proposer boost can potentially lead to new attacks involving strategic or selfish participants. Specifically, we focus on selfish participants who may attempt to manipulate the fork choice rule by forking out blocks already part of the canonical chain to capture the rewards associated to those blocks[4]. Notably, rewards come not only from transaction fees but also from the votes included in blocks[5]. In this situation, a proposer might attempt to steal votes from previous rounds to increase rewards. For this to happen, the current proposer would need to propose a conflicting block that ranks higher than the block chosen with the fork choice rule, and the intuition is that the proposer boost could facilitate this situation. However, determining the conditions under which a proposer might find it profitable to propose such a conflicting block is a tricky task. This challenge is closely related to predicting the behavior of strategic attesters and subsequent proposers, who may also fork out blocks proposed in previous rounds, and this depending on the proposer boost proportion.

To study these situations, we used game theory by modeling the protocol as a game that begins after a possible asynchronous period with a block that can be forked out by the proposer of the first round of the game and where rounds are synchronous. Indeed, we identified a condition, called *cunning condition*, where a proposer can propose a conflicting block that ranks higher than one chosen by the fork choice rule only due to a non-zero proposer boost and the voting count. We then defined two strategies: the *obedient* strategy, which involves following the protocol, and the *cunning* strategy, which allows participants

[2] Finalization of a block occurs when two intersecting quorums will vote for it.
[3] See ethereum/consensus-specs/pull/2730.
[4] This type of attack is known as an ex-post reorg attack [24].
[5] Votes sent in one round are collected by the next proposer.

to fork out blocks. A cunning proposer might fork out by extending the canonical chain from an older block rather than the one chosen by the fork choice rule, while a cunning attester might anticipate that a subsequent proposer will extend from an older block and then vote for it, then following the cunning proposer, as if the attester had never seen more recent blocks (a behavior that is undetectable in an eventually synchronous system). More specifically, we investigate whether the way Ethereum PoS Consensus protocol builds the canonical chain through its fork-choice rule, referred to as the *(prescribed) protocol* in the remainder of the paper, is incentive-compatible. Said in other words, if the obedient strategy is a Nash equilibrium.

Our first result shows that, in a synchronous system, no such attack can occur, as the cunning condition never holds. In other words, the protocol is incentive-compatible in this setting (Theorem 5). Our second result demonstrates that, in an eventually synchronous system, such an attack can occur. Specifically, we show that the obedient strategy is not a Nash equilibrium, implying that the protocol is not incentive-compatible in this setting (Proposition 6). Our main result is a positive finding in the eventually synchronous setting: in the sequence of voting rounds, for all Nash equilibria, there exists a round after which no participant will deviate from the protocol. The intuition behind this is that the opportunity to fork out blocks, represented by the *cunning condition*, diminishes over successive rounds. Therefore, from a certain round onward, the protocol becomes incentive-compatible. We thus describe the protocol as *eventually incentive-compatible* (Theorem 9). We also show that if attesters are restricted to be obedient, at least one proposer will deviate from the protocol and fork out a block. However, if attesters can also be cunning, at most one proposer will deviate, and if the proposer boost is 0.5 or higher, no proposer will deviate. This counterintuitively suggests that allowing participants to act strategically can lead to fewer forks than assuming attesters are always obedient. On the practical side, because the proposer boost is today fixed to 0.4, assuming attesters to be obedient or cunning leads to the same outcome, with exactly one proposer deviating from the protocol after an asynchronous period if the cunning condition holds. We believe these findings can help to better understand the role of the proposer boost and its effects to guide the design of future versions of the protocol.

Related Work. Our work contributes to the line of research trying to combine the use of game theory with distributed computing (e.g., [1, 2, 4]), with a specific focus on blockchain systems (e.g., [3, 5, 6, 9, 15, 16, 19, 17, 23, 28, 33]), albeit contrarily to them, we focus on analyzing the Ethereum PoS protocol. The closest works to ours are [13] and [31]. [13, 31] use game theory to study selfish behavior in Bitcoin when the only source of rewards is with transaction fees. They show that the Bitcoin blockchain becomes unstable since block miners fork the Bitcoin blockchain to get the most lucrative transactions. These results do not apply to Ethereum PoS since there is no block reward in Ethereum PoS by design and Ethereum PoS features the presence of attesters. We focus on analyzing the behavior of the block proposers and attesters in Ethereum PoS by using game theory and we show that the protocol is stable, even if due to initial asynchronous setting, one proposer can gain more than following the prescribed strategy. To the best of our knowledge, [30] is one of the few works using game theory on the Ethereum PoS protocol. [30] studies at what time the proposer is better off to propose its block, while we analyze to which block the proposer should add its block. We also consider that proposers always propose their block at the beginning of the slot (or round), hence benefit from the proposer boost. A line of research studies the vulnerabilities and resilience of the Ethereum PoS protocol, but they consider only honest and Byzantine processes (e.g., [21, 24, 25, 26, 29]). Contrarily to them,

we consider selfish players, and our analysis does not uncover any critical vulnerability of the Ethereum PoS protocol, but shows that a selfish participant prefers to deviate from the prescribed protocol to earn more and some blocks can be forked out from time to time. [27, 32] show how Byzantine actors can benefit from the incentive mechanisms in place in the Ethereum PoS blockchain to break the security of the blockchain. We differ from them since [27, 32] do not consider rational selfish players, but only uncover vulnerabilities when Byzantine processes take advantage of the incentive mechanism in place. Contrary to [32] we do not consider bribery or collusion between proposers and attesters in this analysis.

The paper is organized as follows. Section 2 gives an overview of the Ethereum PoS protocol. Section 3 presents the game, Section 4 our contributions, and Section 5 concludes.

2 Ethereum protocol overview

For our analysis, the main elements of the protocol used are the following:

- **Rounds (Slots).** These are the time frames dictated by the protocol for *proposers* and *attesters* to propose and vote for blocks. We use the terms slot and round interchangeably depending on the context.
- **Proposer.** The proposer's role is to propose a block during a specific round (or slot). There is one proposer per round.
- **Attester.** There is a set of attesters per round (or slot). The attester's role is to produce an attestation, which is a vote for a specific block. Attestations determine the weights of the blocks, which are used by the fork choice rule.
- **Fork choice rule.** The fork choice rule is the protocol's rule that determines, in case of a fork, which block is the head of the chain.
- **Canonical chain.** The canonical chain is the chain designated by the fork choice rule.

Fork Choice Rule. Despite their name, blockchains are closer to block trees. Forks can occur, causing the blockchain to have several branches rather than a single chain. The block tree originates at the genesis block, with each block pointing to its predecessor (the unique parent). To solve forks, the protocol defines a function called the fork choice rule, $\mathcal{F}$, which indicates, at each slot s, on which block to build or attest based on the tree of all blocks, $\mathcal{T}_s$, and the set of attestations received so far, $\mathcal{A}_s$. This block is called the head of the canonical chain. To determine the head of the canonical chain, the fork choice rule follows these steps:

1. Go through the set of all attestations $\mathcal{A}_s = \cup_{i=0}^{s} a_i$, where a_i is the set of attestations received during slot i, and k is the total number of slots. Keep only the last attestation from each attester.
2. For each attestation, add a weight[6] to each block attested to, as well as to all of its parents. This process gives an *attestation weight* to each block using the tree of blocks $\mathcal{T}_n$ and the set of attestations $\mathcal{A}_n$ at slot n. The way in which the attestation weight is computed is explained below.
3. Start from the genesis block and continue along the chain by following the block with the highest attestation weight at each fork in the block tree. Return the block that has no children. This block is the head of the canonical chain.

[6] In the protocol, the weight added is proportional to the stake of the corresponding validator. In our model, each validator is assumed to have the same stake in our analysis, as generally observed in practice. Under this hypothesis and without loss of generality, we choose the stake to be one, so that counting the attestation weight is equivalent to counting the number of attestations for this block or its descendants.

During the execution of the protocol, it is prescribed that the block proposer of slot s executes the fork choice rule $\mathcal{F}(\mathcal{T}_{s-1}, \mathcal{A}_{s-1})$ to determine the parent of its block. During the same slot, the attester should use the fork choice rule $\mathcal{F}(\mathcal{T}_s, \mathcal{A}_{s-1} + \rho a)$ to determine which block to vote for. The notation $\mathcal{A}_{s-1} + \rho a$ indicates that an additional attestation weight of ρa is added on the block of the current round. The addition of the attestation with ρa is called the *proposer boost* and is explained below.

Attestation Weight. We now expand on the notion of attestation weight introduced above. The attestation weight of a block is the sum of all attestations sent for this block, as well as all attestations sent for the descendants of this block. This means that an attestation not only supports a single block but also the entire chain of blocks leading to it. We refer to the *total* attestation weight of a branch of blocks as the sum of all attestations for that branch.

Proposer Boost. The proposer boost, denoted as $\rho \in (0, 1)$, temporarily assigns ρa artificial attestations to a *timely* block, where a represents the number of attesters per slot.[7] This mechanism of proposer boost adds additional attestation weight to a block exclusively during the slot in which it is proposed. Specifically, if a block is received early in slot k, ρa artificial attestations are temporarily added to it. In practice, being early means being received in the first 4 seconds of the slot. [30] indicates that "98% of all blocks are observed by our nodes within four seconds of the slot", highlighting the significant importance of proposer boost usage. Currently, the proposer boost is set at 0.4, effectively adding $0.4 \times a$ to the block's current attestation weight. This adjustment influences the attestation weights so that during slot k, the timely block B_k carries additional weight, thereby affecting the fork choice rule.

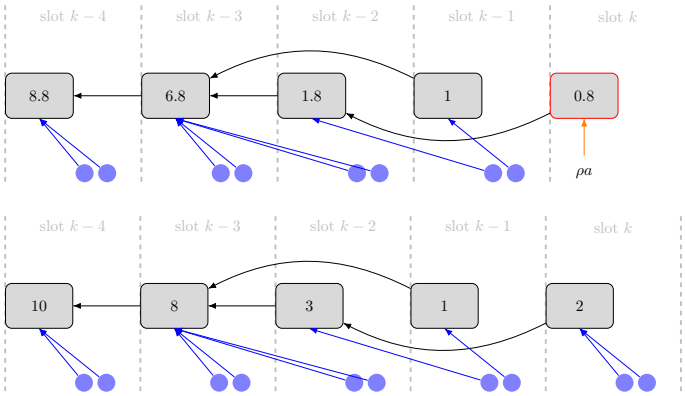

Figure 1 Evaluation of the fork choice rule executed after a timely block proposal in slot k. In this example, there are 2 attesters per slot. Two attestations are sent per slot, represented by a circle pointing to the attested block. Artificial attestations in the current slot k add an attestation weight of ρa for the fork choice rule. At the beginning of slot k, a timely block is proposed, and the proposer boost of $\rho a = 0.8 (= 0.4 \times 2)$ is applied. The fork choice rule selects that block as the head of the chain. To resolve forks, the fork choice rule selects the block with the biggest attestation weight at the fork, the block of slot $k - 2$. When slot k ends, the proposer boost is cleared, and we observe only the two attestations that followed the fork choice rule, attesting the block of slot k.

[7] In the protocol, the number of attesters per slot is fixed and known in advance. The set of attesters per slot is computed such that each validator becomes attester of a slot periodically.

An example of the fork choice rule with the proposer boost in action is illustrated in Figure 1. The figure captures the chain at two different times during slot k: right after the timely block proposal and at the end of the slot. Each time, we show the weight of the blocks as computed by the fork choice rule. In our study, we remain agnostic about the value of ρ to examine its effects across different values. To analyze the impact of the proposer boost on the validators behavior, we assume that proposers propose their blocks at the beginning of their slot such that they get the benefit of the proposer boost.

Rewards. The rewards for proposers and attesters are computed in a verifiable manner. Based on the content of a block in the canonical chain, which includes attestations and transactions, we can determine the rewards for the attesters responsible for these attestations and for the proposer who included them. Proposers also earn rewards from transaction fees. Validators are incentivized to participate in the process of adding new blocks and attesting them through endogenous rewards inscribed in the protocol. A crucial factor in determining rewards is identifying the canonical chain. For instance, a block proposed but not included in the canonical chain will result in zero rewards for the proposer.

- *Attester Rewards.* An attester is rewarded for its attestation based on two factors: the *timeliness* and the *correctness* of its vote. Table 1 indicates the reward for an attester depending on these two factors. An attestation generated at a given slot can be added

Table 1 Attester's rewards based on the inclusion of the attestation in the chain and its vote. The value of x is arbitrary and we maintain the reward ratio of the protocol based on it.

Timeliness	1 slot	≤ 5 slots	≤ 64 slots
Incorrect attestation vote	$20x/27$	$20x/27$	$6x/27$
Correct attestation vote	x	$20x/27$	$6x/27$

  to the chain from the subsequent slot. The bigger the difference between when the attestation is generated and when it is included in the chain, the lower the attestation reward is for the corresponding attester. An attestation is considered correct, if at its emission's slot s, it points to the block of the highest slot $s' \leq s$ that will be part of the canonical chain. Thus, both timeliness and correctness depend on the actions and votes of other validators.

  As presented in Table 1, correctness only impacts the attester's reward if the attestation is included in the following slot. If the attestation is not included in the next slot, a correct or incorrect attestation brings the same rewards. This may incentivize attesters to align their votes with the proposer of the next slot, regardless of whether the proposer's block ultimately becomes part of the canonical chain.
- *Proposer Rewards.* For the proposer, there are two types of rewards: rewards based on attestations and rewards based on transactions. Proposers earn rewards proportional to the attestation's rewards of the attestations included in the blocks they propose. Additionally, the proposer receives a reward for each transaction included in its block in the form of transaction fees. These rewards are formalized in Section 3, where the utility functions are defined.

3 Model & Game

We model the Ethereum PoS consensus protocol as a game where each player (i.e., validators) is either a *proposer* or an *attester*. Players are rational in that they are utility-maximizers. Ideally, in the prescribed behavior, proposers propose blocks, and attesters

broadcast attestations. The game evolves over n sequential *rounds*. There is one proposer and $a \in \mathbb{N}$ attesters per round, hence, we have n proposers and an attesters. The value of n is unknown to the players.

Similar to the approach in [13], we assume the following: (i) The game occurs during a synchronous period, where the network is considered synchronous with no latency. This means that as soon as an information (such as a block, attestation, or transaction) is published (through a broadcast), all players immediately become aware of it; (ii) The synchronous period follows an asynchronous period, during which there may have been delays in information transmission. This second assumption aligns with the hypothesis of network behavior in the Ethereum PoS protocol [12]. Therefore, our game is set in the synchronous period, but the initial state is influenced by events that occurred during the preceding asynchronous period. In this initial asynchronous period, blocks may have an uneven distribution of attestations across rounds, unlike in a permanently synchronous scenario. Each player is agnostic of when the game starts. We also consider a purely synchronous case in which the synchronous period begins with the genesis block (see Theorem 5).

Under this model, the broadcast of the block proposal and attestations in our game are treated as atomic events. Thus, in each round, there are three distinct events:

1. *Block proposal.* The designated proposer for the round[8] proposes a new block, selecting a previously existing block in the observed blockchain as its parent. When a proposer prepares a block, it adds all available transactions and attestations received but not included in parents of the block. Once the transactions and attestations are included, the block is proposed, meaning it is sent to the network.
2. *Generation of transactions.* Transactions are sent by users and observed by proposers and attesters. Each transaction has associated fees that will be available for the **next** proposer to include.
3. *Attestations.* After the block proposal for the round, all attesters of the round choose which previously proposed block to attest and they send their attestations simultaneously. The attestations will be available for the next proposer to include.

The protocol prescribes that proposers (resp. attesters) select as parent (resp. block to attest) the block identified by the fork choice rule, i.e., the head of the canonical chain.

Proposers and attesters are financially motivated to participate in the protocol, and it remains to show whether the incentive to maximize their gains aligns with the protocol.

The game

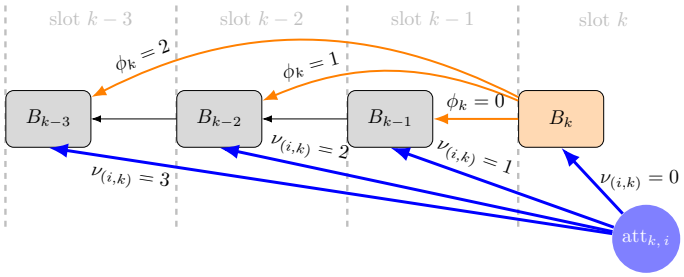

Figure 2 Actions available to an attester (blue) and a proposer (orange). Attester i selects which block to attest with $\nu_{(i,k)}$. The proposer selects the parent of its block with ϕ_k.

[8] In the protocol, the proposer for each slot is deterministically selected with a pseudo random function based on the blockchain data, hence known by all (see [26]).

We denote the set of players (the validators) as $\mathcal{V} = \{P, A\}$, consisting of a set of proposers P and a set of attesters A. Per round, there is one proposer and $a \in \mathbb{N}$ attesters. Hence, the number of proposers is $|P| = n$, and the number of attesters is $|A| = an$, with $a, n \in \mathbb{N}$.

We model the interactions between proposers and attesters during n rounds as a game. In each round, the timeline of events is as follows: (i) a block is proposed at the beginning of the round,[9] (ii) new transactions are proposed, and (iii) all the attesters of the round send their attestations simultaneously. Therefore, the game is dynamic, with each stage corresponding to a round. In each round, the attesters play a simultaneous game following the proposal by the round's proposer. Our interest lies in the actions that the proposers and attesters have, which we now describe.

Actions. When it is their turn,[10] a proposer must choose which block to extend, and an attester must select which block to attest. The action will take the form of a variable that indicates how many rounds prior a proposer attaches its block to, or an attester attests. More formally, the action of the proposer in round k is to assign a value to its variable $\phi_k \in \mathbb{N}$, corresponding to the difference between the current round and the round of the block selected as parent. Similarly, after the block proposal in round k, each attester i of round k must assign a value to its variable $\nu_{i,k} \in \mathbb{N}$ that represents the difference between the current round and the round of the block being attested. We depict a subset of the action space in Figure 2. In more detail:

- At the beginning of round k, a proposer p chooses the parent of its block B_k. We denote this action by $\phi_k \in \mathbb{N}$. $\phi_k = l$ means that B_k's parent is B_{k-1-l}. Thus, if B_k's parent is the block from the previous round $k-1$, then $\phi_k = 0$.
  Blocks contain two types of data: attestations and transactions. There is no limitation on the number of transactions and attestations in a block.[11] A proposer always includes all available transactions and attestations (not included in any predecessors of the block).
- After the block proposal in round k, all attesters of round k simultaneously choose which block to attest. The attestation of attester i in round k points to a specific block determined by $\nu_{(i,k)} \in \mathbb{N}$. $\nu_{(i,k)} = l$ means that the block B_{k-l} is the one attested by attester i in round k. If validator i attests the block in the current round k, then $\nu_{(i,k)} = 0$.

These actions are repeated in each round. Note that not proposing or not attesting to a block is not an available action. Similarly all available attestations/transactions are always included.

The last piece of data needed for our study is to determine whether a block B_k eventually belongs to the canonical chain. This information is represented by $\chi_k \in \{0, 1\}$, where $\chi_k = 1$ if the block from round k eventually belongs to the canonical chain, and $\chi_k = 0$ otherwise. This information becomes known at the end of round n, which marks the end of the game.

Notice that for any round k, always assigning value of 0 to ϕ_k as a proposer (or 0 to $\nu_{(i,k)}$ as an attester) is not necessarily the prescribed action. The prescribed action is to follow the fork choice rule (see Figure 1).

Strategies. A strategy of a player i is a function σ_i, which takes as input the entire tree of blocks in the blockchain, as well as the attestations received so far, and produces as output a number, say $n \in \mathbb{N}$. Since the only information available are the tree of blocks and the attestations, the signature of a player's strategy is $\mathcal{T} \times \mathcal{A} \to \mathbb{N}$, where $\mathcal{T}$ is the set of blocks and $\mathcal{A}$ is the set of available attestations.

---

[9] Every block thus receives the proposer boost in our model.

[10] Recall that each proposer/attester is uniquely assigned to a round, and this information is verifiable, allowing players to take action only in their corresponding round.

[11] This simplification is similar to the one made in [13].

For the proposer of round k, the prescribed strategy is $\sigma_{(0,k)}(\mathcal{T}_{k-1}, \mathcal{A}_{k-1}) = l$, where $\mathcal{F}(\mathcal{T}_{k-1}, \mathcal{A}_{k-1}) = B_{k-1-l}$. For attester i at round k, it is $\sigma_{(i,k)}(\mathcal{T}_k, \mathcal{A}_{k-1}) = l$, where $\mathcal{F}(\mathcal{T}_k, \mathcal{A}_{k-1} + \rho a) = B_{k-l}$. A player *deviates* from the prescribed protocol when their strategy produces a number different from the round of the block resulting from the fork choice rule.

A strategy profile $\sigma = (\sigma_{0,1}, \ldots, \sigma_{a,1}, \sigma_{0,2}, \ldots, \sigma_{a,2}, \ldots, \sigma_{0,n}, \ldots, \sigma_{a,n})$ is a vector where each component is a strategy of the corresponding player. We denote by $\mathcal{S}$ the set of all strategy profiles and by $\mathcal{S}_{(i,k)}$ the set of strategies for the player of component (i,k). In this notation, players with indices (i,k) where $i = 0$ are proposers, while players with indices (i,k) where $1 \leq i \leq a$ are attesters. For clarity, we denote by (σ_{-i}, σ'_i) the strategy profile σ where, instead of playing with strategy σ_i, player i deviates and uses strategy σ'_i instead.

It remains to define the reward of the players. At the end of round n, the payoff of all players is computed and given by the function $u : \mathcal{S} \to \mathbb{R}^{n+an}$ (defined below). The payoff of each player is given by its component in the reward vector, which depends on its type and is determined by a reward function. In the remainder of the paper, for clarity, for any strategy profile σ, we write $u_{i,j}(\sigma)$ instead of $u(\sigma)_{(i,k)}$, where $u_{i,j}(\sigma)$ represents the payoff of player (i,j), and player $(0,j)$ is the proposer of round j.

Payoff. Attesters' rewards vary depending on when their attestations are included in a block and which block they attest to. This can incentivize them to align their attestations with the behavior of future block proposers. Block proposers have a clear incentive to accumulate the maximum transaction fees and attestations to maximize their rewards. One strategy to achieve this is to fork the chain and include in the new block all the attestations and transactions that do not belong to the new chain. However, if the block does not end up in the canonical chain, the block proposer will not receive any rewards. This incentivizes the proposer to consider other behaviors, as we will see.

Given a strategy profile σ, the reward of attester i in round k, player (i,k), depends on a variable $x > 0$ set by the protocol and the round in which the attestation is subsequently included in a block:

$$u_{(i,k)}(\sigma) = \begin{cases} x & \text{if } \sigma_{(i,k)} \text{ sets } \nu_{(i,k)} = \phi_{k+1} \text{ and } \chi_{k+1} = 1, \\ 20x/27 & \text{else if } \exists l \in [\![1,5]\!] : \chi_{k+l} = 1, \\ 6x/27 = 2x/9 & \text{else if } \exists l \in [\![6,64]\!] : \chi_{k+l} = 1, \\ 0 & \text{otherwise (if } \chi_{k+l} = 1, \text{ with } l > 64 \text{ or not included).} \end{cases} \tag{1}$$

Here, $x > 0$ and χ_{n+1} denote the fact that the block of round $n+1$ belongs to the canonical chain. The rewards for the attester can be understood as follows: they are maximized when the attestation votes for the parent of the block in the subsequent round, and this block in the subsequent round ends up in the canonical chain.

The actual rewards for an attester are detailed in Table 1. Note that the reward is influenced by the correctness of the attestation only if it is included in the block of the next round. Additionally, if the block of round $k+1$ does not end up in the canonical chain, the attesters of round k can never receive the maximum reward.

For the proposer of round k, the reward function is given by:

$$u_{(0,k)}(\sigma) = \chi_k \sum_{j=n-\phi_k}^{n} \left(f_{j-1} + \frac{1}{7} \sum_{i=1}^{a} u_{(i,j)}(\sigma) \right). \tag{2}$$

The reward is the sum of transaction fees and attestation rewards over the rounds separating the block from its parent, multiplied by the factor that indicates the block belongs to the canonical chain. Here, $f_j > 0$ represents the transaction fees generated during

round $j-1$. The transaction fees are the incentives that motivate proposers to include transactions in their block. The proposer also receives $1/7$ of what the attesters receive for their attestations being included in a block. The factor χ_n, indicates whether the block ends up in the canonical chain, is applied to the entire reward since; if the block is not included in the canonical chain, it does not yield any rewards.

4 Analysis

In this section, we explore a set of possible strategies for proposers and attesters. Each can either follow the obedient strategy or adopt a cunning strategy. The obedient strategy is the one prescribed by the protocol. In contrast, the cunning strategy may deviate from the protocol by exploiting the proposer boost to remain part of the canonical chain.

4.1 Preliminaries

To ensure clarity and self-containment, we present well-known game-theoretic concepts. They are useful for categorizing strategy equilibria and exploring possible states of the game.

▶ **Definition 1** (Best response). *A strategy σ_i^* is a best response for player i to the strategy profile σ_{-i} of the other players if: $\forall \sigma_i \in \mathcal{S}_i, u_i(\sigma_{-i}, \sigma_i^*) \geq u_i(\sigma_{-i}, \sigma_i)$, where u_i is the payoff function for player i, σ_{-i} is the strategy profile of all other players, and $\mathcal{S}_i$ is the set of all possible strategies for player i.*

▶ **Definition 2** (Nash equilibrium). *A strategy profile $\sigma^* = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_n^*)$ is a Nash equilibrium if each player's strategy σ_i^* is a best response to the strategies σ_{-i}^* of the other players. Formally, $\forall \sigma_i \in \mathcal{S}_i$ and for all players $i, u_i(\sigma_{-i}^*, \sigma_i^*) \geq u_i(\sigma_{-i}^*, \sigma_i)$; where u_i is the payoff function for player i, σ_{-i}^* is the strategy profile of all other players in the equilibrium, and $\mathcal{S}_i$ is the set of all possible strategies for player i.*

In summary, the concept of a best response helps to identify the optimal strategy for a player given the strategies of the other players. Nash equilibrium defines a state where each player's strategy is a best response to the strategies of the other players, ensuring no player can benefit from unilaterally changing their strategy.

4.2 Analyzed strategies

4.2.1 Obedient strategies

As Carlsten et al. [13], we first describe the strategy of proposers and attesters that act as prescribed by the protocol, we refer to them as obedient. However in the case of Ethereum, the action prescribed by the fork choice rule is a bit more complex.

Obedient Proposer ($\sigma^O_{(0,k)}$):
Action: $\phi_k = l$, where $\mathcal{F}(\mathcal{T}_{k-1}, \mathcal{A}_{k-1}) \to B_{k-1-l}$.

The strategy of an obedient proposer at round k is to propose a block B_k linked to the block designated by the fork choice rule $\mathcal{F}(\mathcal{T}_{k-1}, \mathcal{A}_{k-1}) \to B_{k-1-l}$.

Obedient Attester ($\sigma^O_{(i,k)}$):
Action: $\nu_{(i,k)} = l$ (Block attested is B_{k-l}.)

The obedient attester strategy of attester i is to attest to the block designated by the fork choice rule $\mathcal{F}(\mathcal{T}_k, \mathcal{A}_{k-1} + \rho a) \to B_{k-l}$.

We denote by $\sigma^O_{(i,j)}$ the obedient strategy of player (i,j) and by σ^O the strategy profile where all players follow the obedient strategy.

When proposers and attesters follow the obedient strategy, we can evaluate the rewards each of them will receive. Since they will all follow the fork choice rule and there are no delays, no forks will occur, and attesters will attest to the block of their round. Moreover, each attestation will be included in the following round and will be correct. For proposers and attesters following the actions prescribed by the protocol, the rewards are as follows: (a) For each attester i following the obedient strategy, the reward is: $u_{(i,k)}(\sigma^O) = x$, where σ^O is the strategy profile in which all proposers and attesters are obedient. (b) For the proposer of round k, the reward is: $u_{(0,k)}(\sigma^O) = (ax)/7 + f_{k-1}$.

With this strategy profile, attesters obtain the maximum reward attainable (Equation 1). However, there is no maximum reward for a block proposer, as their rewards increase the older their block's parent is.

4.2.2 Cunning strategies

We now examine a strategy that could yield more rewards for validators than simply following the protocol. In some situations, deviating from the protocol can allow validators to accumulate more rewards without incurring penalties. We refer to this as the *cunning* behavior. For a proposer, the strategy involves choosing a block parent for its proposal that maximizes its rewards.

As the block parent's round is further away from the new block, the proposer can include more transactions and attestations to increase its rewards. The ideal block parent, in theory, would be the genesis block. However, for the block to actually yield rewards, it must become part of the canonical chain. The cunning proposer will always propose a block that is considered the head of the canonical chain during its round. For instance, a cunning proposer will not strictly follow the fork choice rule to determine its block's parent. Instead, it will subtly test whether it can choose an older block as the parent while still having its block become the head of the canonical chain. The block that maximizes rewards – typically the oldest possible block – will be selected as the parent by the cunning proposer.

Cunning Proposer $(\sigma^C_{(0,k)})$:
Action: $\phi_k = \max\{x \in \mathbb{N} : \mathcal{F}(\mathcal{T}_x, \mathcal{A}_{x-1} + \rho a) = B_k\}$

The cunning proposer's block extends the block that leaves the most available transactions and attestations while still being selected as the head of the canonical chain by the fork choice rule (for attesters in the same round) due to the proposer boost ρa.

Cunning Attester $(\sigma^C_{(i,k)})$:
Action: $\nu_{(i,k)} = \sigma^C_{0,k+1}(\mathcal{T}_k, \mathcal{A}_{k-1} \cup A^O_k)$.

The cunning attester (i,k) attests to the parent of the block in round $k+1$, as if all other attesters in round k act obediently (A^O_k) and that the proposer of round $k+1$ acts cunningly, irrespective of their effective action which at the time of computation is unknown.

We denote by $\sigma^C_{(0,k)}$ the cunning strategy of the proposer of round k. For an attester acting cunningly corresponds to deviating from the protocol to act in accordance with the following proposer. The assumption made by the cunning attester is that the other attesters of its round follow the prescribed protocol while the next proposer follows the cunning strategy.

▶ Remark 3. The obedient and the cunning strategy can result in the same outcome.

It is important to note that while the cunning and obedient strategies are distinct, the actions resulting from them can sometimes be identical. Indeed, the action taken by a cunning proposer is to propose a block with the oldest possible parent while still ensuring the block is designated by the fork choice rule for the attesters of the round. However, if the oldest possible parent is the same block initially designated by the fork choice rule, the action will align with the protocol, just as it would under the obedient strategy. Therefore, we say that a cunning player follows the protocol if the action is the one prescribed by the protocol. Conversely, we say they act cunningly if the action taken differs from what is prescribed by the protocol, making the cunning strategy truly distinct from the obedient strategy.

▶ **Condition 4** (Cunning condition). *The divergence between cunning and obedient proposer behavior occurs when the branch containing the block designated by the fork choice rule, with a total attestation weight w_f, has a concurrent branch with a total attestation weight w_g with:*

$$w_f - w_g \leq \rho a.$$

We call this inequality the ***cunning condition****.*

First, it is clear that w_f is always greater than w_g, as the block designated by the fork choice rule is on the branch with a total attestation weight of w_f.[12] To understand the cunning condition, we consider two illustrations:

1. The first, and less intuitive, case is presented in Figure 3. This showcases the scenario where $w_g = 0$. A branch can consist of many blocks, a single block, or, no block at all. The newly proposed block can become the head of the canonical chain by attaching itself to the first block with more than ρa attestation weight. If the block designated by the fork choice rule is on a branch with a total attestation weight less than ρa, the cunning behavior differs from the obedient behavior.
2. Another representation of the cunning condition is shown in Figure 4. Here, there are two distinct branches, each with one block. The branch of the block designated by the fork choice rule has an attestation weight of $w_f = 3$, while the concurrent branch has $w_g = 2$. In this case, where $\rho a = 1.2$, the condition is met, allowing the proposer to act cunningly.

Notice that without the proposer boost, the cunning proposer strategy would never differ from the obedient proposer strategy. This strategy relies on the advantage provided by the proposer boost to ensure that its block becomes the head according to the fork choice rule.

Conversely, as the proposer boost increases, the opportunity for the cunning block proposer to act cunningly arises more frequently when all attesters are obedient.

4.3 Strategic analysis

First, we show our results in a synchronous network, where the game starts at the genesis block. In this setting, the prescribed protocol is *incentive-compatible*, in the sense that the strategy profile where everyone follows the obedient strategy is a Nash equilibrium ([18, 20]).

▶ **Theorem 5.** *In a synchronous setting, all validators being obedient is a Nash equilibrium.*

The intuition is that if the game starts after a synchronous period, no partition or message delay would permit the cunning condition to hold. No participant has, therefore, an incentive to fork out a block since such a tentative cannot be perceived as the canonical chain, hence the others will not accept it (as they follow the prescribed protocol). Therefore, it is better

[12] Note, therefore, that in case of no proposer boost, i.e., $\rho = 0$, the cunning condition cannot hold.

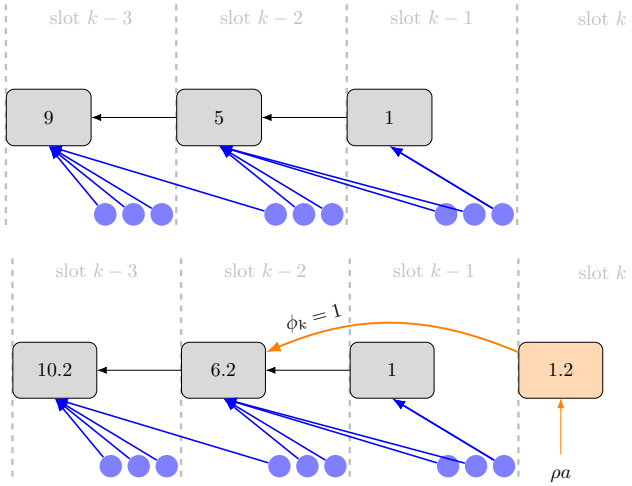

Figure 3 Cunning proposer $(0, k)$ deviating from the prescribed protocol with $\rho a = 1.2 (= 0.4 \times 3)$ when the block designated by the fork choice rule has an attestation weight less than ρa.

for all validators to use the prescribed protocol to add blocks (and/or attestations) on top of the canonical chain, to guarantee to be followed by the others. The prescribed protocol is then incentive-compatible in this setting.

We continue our analysis with an **eventually synchronous network where the cunning condition holds for the first proposer of the game**.

Best response of single proposer among $n-1$ obedient proposers and an obedient attesters. We first study the behavior of one proposer when all others are obedient with respect to their designated round. In the case in which the proposer is associated to the first round of the game, for this proposer the cunning strategy is the best response and the proposer deviates from the protocol as the cunning condition holds (Proposition 6).

▶ **Proposition 6.** *The best-response for the proposer if all the other validators are obedient is the cunning strategy. The proposer deviates from the prescribed protocol iff it is the first proposer of the game.*

Let us note that when everyone else is obedient, the cunning strategy can only differ from the obedient strategy for the first proposer. All subsequent proposers will follow the protocol regardless of whether they follow the cunning or obedient strategy because the cunning condition is never satisfied.

A direct corollary of Proposition 6 is that the obedient strategy is not a Nash equilibrium.

Best response of single proposer among $n-1$ cunning proposers and an obedient attesters. We study the behavior of one proposer when all others proposers are cunning and attesters are obedient. We have two cases: (a) $\rho < 1/2$. The proposer deviates from the prescribed protocol iff it is the first proposer of the game. (Proposition 11 in the Appendix). (b) $\rho \geq 1/2$. In this case, the cunning condition might apply to multiple consecutive proposers. We have two sub-cases. (i) If the cunning condition holds only for the first proposer, then the proposer deviates from the prescribed protocol (Lemma 12 in Appendix), trivially, all the subsequent proposers will follow the protocol because the cunning condition does not hold for them; (ii) Otherwise, the cunning strategy is the best response under the condition presented in Proposition 7. In this scenario, each proposer deviates from the prescribed protocol.

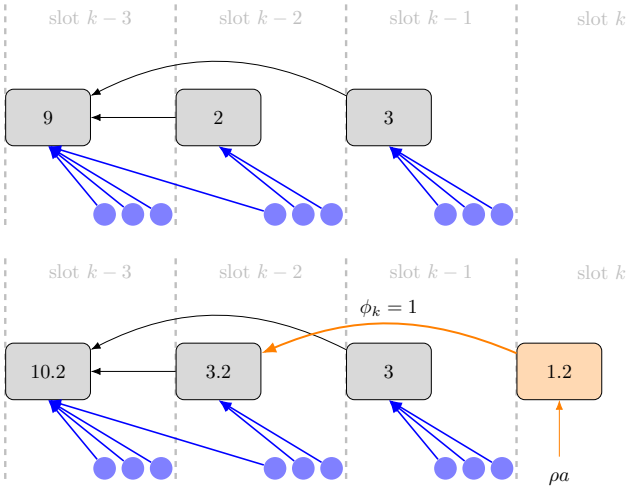

Figure 4 Cunning proposer $(0, k)$ deviating from the prescribed protocol with $\rho a = 1.2 (= 0.4 \times 3)$.

▶ **Proposition 7.** *When all proposers are cunning, attesters are obedient, and $\rho \geq 1/2$, the cunning strategy is a best response if the cunning condition holds for the second proposer and:*

$$(f_{k-2} - f_{k-1})/2 \geq ax/27,$$

where f_k denotes the transaction fees emitted at round k.

We illustrate in Figure 5 the "bouncing" that unfolds due to cunning proposers, giving an intuition for Proposition 7.

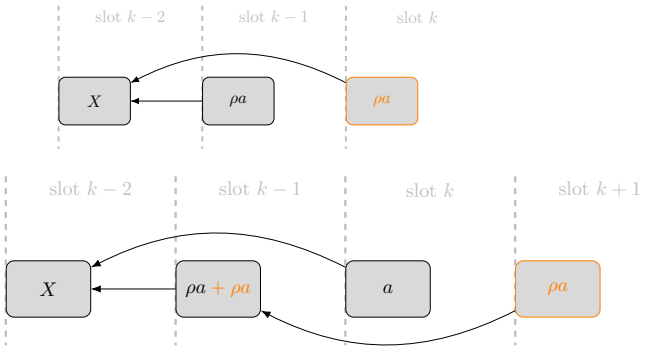

Figure 5 X indicates that the value of the block is irrelevant. In this scenario, the proposer of slot k is cunning and all the attesters are obedient. The proposer of slot k takes advantage of the proposer boost to become the head of the canonical chain. The proposer of slot $k+1$ can cunningly become the head of the canonical chain by attaching its block to slot $k-1$ and can become the head of the chain if $2\rho a \geq a$, i.e., $\rho \geq 1/2$. In conclusion, a proposer boost greater than $1/2$ can create a situation in which multiple forks occur in the presence of cunning proposers and obedient attesters.

Case of attesters. We also analyze the case of attesters, when all proposers follow a cunning strategy. Due to space limitations, results are formally presented in the Appendix (Proposition 14, Proposition 15 and Proposition 17).

If an attester anticipates that all other attesters are compliant, following the protocol is the best response when $\rho < 1/2$. However, if $\rho \geq 1/2$, the best response is to adopt the cunning strategy, leading the attester to deviate from the protocol only if the second proposer satisfies the cunning condition.

If an attester expects all other attesters to be cunning, their best response is also to follow the cunning strategy. This strategy leads first-round attesters to deviate only if the cunning condition holds for the second proposer. However, this deviation prevents the cunning condition from holding for the third proposer, prompting validators in subsequent rounds to follow the protocol.

Best response of single proposer among $n-1$ cunning proposers and an cunning attesters. Now that the attesters can also act cunningly, let us evaluate the best response of proposers. We have two cases: (a) $\rho < 1/2$. The proposer deviates from the prescribed protocol iff it is the first proposer of the game. (Proposition 17 in the Appendix). (b) If $\rho \geq 1/2$, the cunning condition might apply to multiple consecutive proposers. In this case, if the cunning condition holds for the second proposer, the cunning attesters are a threat for the proposers from being cunning, therefore, to guarantee rewards, the obedient strategy is the best response for the proposers, see Proposition 8. In this scenario, each proposer follows the prescribed protocol.

▶ **Proposition 8.** *If $\rho \geq 1/2$, all validators are cunning, and the cunning condition holds for the second proposer, then the obedient strategy is the best-response for the proposers. Otherwise, if the cunning condition does not hold for the second proposer, the cunning strategy is the best-response only for the first proposer.*

A counterintuitive finding is then that with a higher proposer boost ($\rho > 1/2$), the obedient strategy becomes more favorable. This is because the potential rewards from being cunning may not outweigh the risk of being excluded from the canonical chain, especially if cunning attesters refuse to attest the block. Thus, the certainty of inclusion and available rewards can make obedience the preferred strategy.

Eventual Incentive compatibility. We have shown that all validators being obedient is not a Nash equilibrium (Proposition 6). Nonetheless, the outcomes of the strategies (for proposers and attesters) presented so far imply that in all equilibria, there exists a round after which all validators follow the prescribed protocol, as stated by the following theorem.

▶ **Theorem 9.** *In all Nash equilibria, there is a round after which all validators follow the protocol.*

The intuition of Theorem 9 is that the possibility to satisfy the cunning condition diminishes over successive rounds. Therefore, from a certain round onward, the cunning condition does not hold anymore. From that point on, all validators follow the protocol. Hence, the protocol becomes incentive-compatible.

We conclude that under perfect network conditions, regardless of the proposer boost, the obedient strategy eventually prevails, which is in line with Theorem 5.

5 Conclusion & Future Work

In this work, we have analyzed the Ethereum PoS protocol through a game-theoretic lens, focusing on the incentive mechanisms that drive the behavior of proposers and attesters. Our findings reveal that the current design leads strategic validators to eventually adhere to the protocol. Specifically, the protocol prevents prolonged forks during periods of good network conditions. Surprisingly, a high proposer boost (greater than 1/2) can even discourage cunning behavior. This initial analysis concentrated on two strategies: cunning and obedient.

Future research will expand to include a broader range of strategies. Additionally, exploring different assumptions about network conditions, such as more realistic communication delays and crashes, is expected to have a significant impact on the results.

References

1 Ittai Abraham, Lorenzo Alvisi, and Joseph Y. Halpern. Distributed computing meets game theory: combining insights from two fields. *SIGACT News*, 42(2):69–76, 2011. doi:10.1145/1998037.1998055.

2 Ittai Abraham, Danny Dolev, Rica Gonen, and Joseph Y. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In Eric Ruppert and Dahlia Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 53–62. ACM, 2006. doi:10.1145/1146381.1146393.

3 Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. *CoRR, abs/1612.02916*, 2016. arXiv:1612.02916.

4 Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Michael Dahlin, Jean-Philippe Martin, and Carl Porth. BAR fault tolerance for cooperative services. In Andrew Herbert and Kenneth P. Birman, editors, *Proceedings of the 20th ACM Symposium on Operating Systems Principles 2005, SOSP 2005, Brighton, UK, October 23-26, 2005*, pages 45–58. ACM, 2005. doi:10.1145/1095810.1095816.

5 Yackolley Amoussou-Guenou, Bruno Biais, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. Rational behaviors in committee-based blockchains. In *24th International Conference on Principles of Distributed Systems, OPODIS 2020, December 14-16, 2020, Strasbourg, France (Virtual Conference)*, volume 184 of *LIPIcs*, pages 12:1–12:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.OPODIS.2020.12.

6 Yackolley Amoussou-Guenou, Bruno Biais, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. Rational vs byzantine players in consensus-based blockchains. In Amal El Fallah Seghrouchni, Gita Sukthankar, Bo An, and Neil Yorke-Smith, editors, *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*, pages 43–51. International Foundation for Autonomous Agents and Multiagent Systems, 2020. doi:10.5555/3398761.3398772.

7 Yackolley Amoussou-Guenou, Antonella Del Pozzo, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. Dissecting tendermint. In *Networked Systems – 7th International Conference, NETYS 2019, Marrakech, Morocco, June 19-21, 2019, Revised Selected Papers*, volume 11704 of *Lecture Notes in Computer Science*, pages 166–182. Springer, 2019. doi:10.1007/978-3-030-31277-0_11.

8 Lacramioara Astefanoaei, Pierre Chambart, Antonella Del Pozzo, Thibault Rieutord, Sara Tucci Piergiovanni, and Eugen Zalinescu. Tenderbake – A solution to dynamic repeated consensus for blockchains. In Vincent Gramoli and Mohammad Sadoghi, editors, *4th International Symposium on Foundations and Applications of Blockchain 2021, FAB 2021, May 7, 2021, University of California, Davis, California, USA (Virtual Conference)*, volume 92 of *OASIcs*, pages 1:1–1:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/OASICS.FAB.2021.1.

9 Bruno Biais, Christophe Bisiere, Matthieu Bouvard, and Catherine Casamatta. The blockchain folk theorem. *The Review of Financial Studies*, 32(5):1662–1715, 2019.

10 Ethan Buchman, Jae Kwon, and Zarko Milosevic. The latest gossip on BFT consensus. *CoRR*, abs/1807.04938, 2018. arXiv:1807.04938.

11 Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *CoRR*, abs/1710.09437, 2017. arXiv:1710.09437.

12 Vitalik Buterin, Diego Hernandez, Thor Kamphefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, and Yan X. Zhang. Combining GHOST and casper. *CoRR*, abs/2003.03052, 2020. `arXiv:2003.03052`.

13 Miles Carlsten, Harry A. Kalodner, S. Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 154–167. ACM, 2016. `doi:10.1145/2976749.2978408`.

14 Francesco D'Amato, Joachim Neu, Ertem Nusret Tas, and David Tse. Goldfish: No more attacks on ethereum?! In *Financial Cryptography and Data Security – 28th International Conference, FC 2024, Willemstad, Curaçao*, Lecture Notes in Computer Science. Springer, 2024.

15 Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security – 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, volume 8437 of *Lecture Notes in Computer Science*, pages 436–454. Springer, 2014. `doi:10.1007/978-3-662-45472-5_28`.

16 Mehdi Fooladgar, Mohammad Hossein Manshaei, Murtuza Jadliwala, and Mohammad Ashiqur Rahman. On incentive compatible role-based reward distribution in algorand. In *50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2020, Valencia, Spain, June 29 – July 2, 2020*, pages 452–463. IEEE, 2020. `doi:10.1109/DSN48063.2020.00059`.

17 Cyril Grunspan and Ricardo Pérez-Marco. Selfish mining in ethereum. In *2nd International Conference on Mathematical Research for Blockchain Economy, MARBLE 2020, online, August 24, 2020*, Springer Proceedings in Business and Economics, pages 65–90. Springer, 2020. `doi:10.1007/978-3-030-53356-4_5`.

18 Joseph Y. Halpern and Xavier Vilaça. Rational consensus: Extended abstract. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 137–146. ACM, 2016. `doi:10.1145/2933057.2933088`.

19 Dimitris Karakostas, Aggelos Kiayias, and Thomas Zacharias. Blockchain nash dynamics and the pursuit of compliance. In Maurice Herlihy and Neha Narula, editors, *Proceedings of the 4th ACM Conference on Advances in Financial Technologies, AFT 2022, Cambridge, MA, USA, September 19-21, 2022*, pages 281–293. ACM, 2022. `doi:10.1145/3558535.3559781`.

20 Abhiram Kothapalli, Andrew Miller, and Nikita Borisov. Smartcast: An incentive compatible consensus protocol using smart contracts. In *Financial Cryptography and Data Security – FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers*, volume 10323 of *Lecture Notes in Computer Science*, pages 536–552. Springer, 2017. `doi:10.1007/978-3-319-70278-0_34`.

21 Cosimo Laneve, Sergio Solmonte, and Adele Veschetti. A stochastic analysis of the gasper protocol. In *IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom 2024 – Workshops, Biarritz, France, March 11-15, 2024*, pages 518–523. IEEE, 2024. `doi:10.1109/PERCOMWORKSHOPS59983.2024.10502866`.

22 Ryuya Nakamura. Prevention of bouncing attack on ffg, 2019. URL: `https://ethresear.ch/t/prevention-of-bouncing-attack-on-ffg/6114`.

23 Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 305–320. IEEE, 2016. `doi:10.1109/EUROSP.2016.32`.

24 Joachim Neu, Ertem Nusret Tas, and David Tse. Ebb-and-flow protocols: A resolution of the availability-finality dilemma. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 446–465. IEEE, 2021. `doi:10.1109/SP40001.2021.00045`.

25 Joachim Neu, Ertem Nusret Tas, and David Tse. Two more attacks on proof-of-stake ghost/ethereum. In Jorge M. Soares, Dawn Song, and Marko Vukolic, editors, *Proceedings of the 2022 ACM Workshop on Developments in Consensus, ConsensusDay 2022, Los Angeles, CA, USA, 7 November 2022*, pages 43–52. ACM, 2022. doi:10.1145/3560829.3563560.

26 Ulysse Pavloff, Yackolley Amoussou-Guenou, and Sara Tucci Piergiovanni. Ethereum proof-of-stake under scrutiny. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC 2023, Tallinn, Estonia, March 27-31, 2023*, pages 212–221. ACM, 2023. doi:10.1145/3555776.3577655.

27 Ulysse Pavloff, Yackolley Amoussou-Guenou, and Sara Tucci Piergiovanni. Byzantine attacks exploiting penalties in ethereum pos. In *54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2024, Brisbane, Australia, June 24-27, 2024*, pages 53–65. IEEE, 2024. doi:10.1109/DSN58291.2024.00020.

28 Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *Financial Cryptography and Data Security – 20th International Conference, FC 2016, Christ Church, Barbados, February 22-26, 2016, Revised Selected Papers*, volume 9603 of *Lecture Notes in Computer Science*, pages 515–532. Springer, 2016. doi:10.1007/978-3-662-54970-4_30.

29 Caspar Schwarz-Schilling, Joachim Neu, Barnabé Monnot, Aditya Asgaonkar, Ertem Nusret Tas, and David Tse. Three attacks on proof-of-stake ethereum. In Ittay Eyal and Juan A. Garay, editors, *Financial Cryptography and Data Security – 26th International Conference, FC 2022, Grenada, May 2-6, 2022, Revised Selected Papers*, volume 13411 of *Lecture Notes in Computer Science*, pages 560–576. Springer, 2022. doi:10.1007/978-3-031-18283-9_28.

30 Caspar Schwarz-Schilling, Fahad Saleh, Thomas Thiery, Jennifer Pan, Nihar Shah, and Barnabé Monnot. Time is money: Strategic timing games in proof-of-stake protocols. In *5th Conference on Advances in Financial Technologies, AFT 2023, October 23-25, 2023, Princeton, NJ, USA*, volume 282 of *LIPIcs*, pages 30:1–30:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.AFT.2023.30.

31 Itay Tsabary and Ittay Eyal. The gap game. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 713–728. ACM, 2018. doi:10.1145/3243734.3243737.

32 Mingfei Zhang, Rujia Li, and Sisi Duan. Max attestation matters: Making honest parties lose their incentives in ethereum pos. *IACR Cryptol. ePrint Arch.*, page 1622, 2023. URL: https://eprint.iacr.org/2023/1622.

33 Roi Bar Zur, Ittay Eyal, and Aviv Tamar. Efficient mdp analysis for selfish-mining in blockchains. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, AFT '20, pages 113–131, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3419614.3423264.

A Appendix

To prove Theorem 5, we start with the following lemma.

▶ **Lemma 10.** *Once a proposer follows the prescribed protocol, all subsequent validators do so.*

Proof. A proposer $(0, j)$ following the protocol implies that if all attesters of its round also follow the protocol, the next proposer cannot deviate. This is because if all attesters of round j attest the block newly proposed by $(0, j)$, the block will have an attestation weight of a. Since it extends the branch designated by the fork choice rule with an attestation weight $w_f \geq w_g$, where w_g is the attestation weight of any concurrent branch, adding a to w_f ensures that no proposer deviate from the protocol (cf. Condition 4).

Knowing that proposer of round $j+1$ will follow the protocol, the attesters (i, j) of round j follow the protocol as well. No validators can deviate from the protocol after a block is attached to the head of the canonical chain. ◀

Theorem 5. In a synchronous setting, all validators follow the prescribed protocol.

Proof. In a synchronous setting, once the first proposer attach itself to the genesis block, following the protocol, all other validators will follow the protocol as well, as proved by Lemma 10. The first proposer prefer doing so since attaching its block to the genesis is the best-response (and only response). ◀

Proposition 6. The best-response for the proposer if all the other validators are obedient is the cunning strategy. The proposer deviates from the prescribed protocol iff it is the first proposer of the game.

Proof. Let us denote by ϕ_k^C and ϕ_k^O the actions taken by proposer $(0, 0)$ under the cunning strategy and the obedient strategy, respectively. The cunning proposer strategy differs from the obedient strategy when $\phi_k^C > \phi_k^O$. Considering that the rest of the validators follow the obedient strategy, a proposed block that becomes the head of the chain at round k will end up in the canonical chain ($\chi_k = 1$). By construction, $\phi_k^C \geq \phi_k^O$, and in both cases, the proposed block will be the head of the canonical chain and thus belong to the canonical chain ($\chi_k = 1$). Based on the definition of $u_{(0,k)}$ (cf. Equation 2), the reward increases as the sum increases. This implies that $u_{(0,k)}(\sigma^O_{-(0,k)}, \sigma^C_{(0,k)}) \geq u_{(0,k)}(\sigma^O_{-(0,k)}, \sigma^O_{(0,k)})$.

As explained in Lemma 10, if the first proposer follows the protocol all validator will. Thus, only the first proposer can act cunningly since the network's state before the first round is not synchronous, leaving any arrangement of blocks and attestation weights possible. ◀

▶ **Proposition 11.** *If $\rho < 1/2$, the best-response for the proposer is the cunning strategy, if all the other proposers are cunning and attesters are obedient. The proposer deviates from the prescribed protocol iff it is the first proposer of the game.*

Proof. With $\rho < 1/2$, only the first proposer can act cunningly. Let's assume the first proposer acts cunningly, meaning that the cunning condition is satisfied. The maximum gap between w_f and w_g for the first proposer to act cunningly is ρa, such that $w_f = w_g + \rho a$. After the attestations sent by the obedient attesters in the first round, the attestation weight of the branch designated by the fork choice rule becomes $w_g + a$. For the second proposer to act cunningly, it must hold that $w_g + a - w_f \leq \rho a$ (cunning condition for the second proposer). Substituting w_f with the maximum possible gap from w_g, this condition implies that the second proposer can act cunningly if and only if:

$$\begin{aligned} w_g + a - (w_g + \rho a) &\leq \rho a \\ \frac{1}{2} &\leq \rho. \end{aligned} \tag{3}$$

Thus, with cunning proposers and obedient attesters, only the first proposer can act cunningly, deviating from the obedient action. For the first proposer, acting cunningly will yield the maximum rewards. ◀

▶ **Lemma 12.** *The cunning proposer strategy is a best response for a proposer when all other proposers are cunning and attesters are obedient, provided $\rho \geq 1/2$ and the cunning condition does not hold for the second proposer. This causes the first proposer to deviate from the protocol.*

Proof. If the attestation weight of the branch f designated by the fork choice rule, w_f, and the attestation weight w_g of a concurrent branch g are such that $w_g + a - w_f > \rho a$, the second proposer cannot act cunningly with obedient attesters (cunning condition false for the second proposer). We previously showed that if the first proposer is cunning and attaches its block to the concurrent chain g, the obedient attesters will follow, increasing the weight of g to $w_g + a$. By ensuring that $w_g + a - w_f > \rho a$, we prevent the second proposer from changing the canonical chain with the proposer boost. Thus, this condition ensures that the first proposer can be the only one to act cunningly, and in this case, the best response is the cunning strategy. ◀

Proposition 7. When all proposers are cunning, attesters are obedient, and $\rho \geq 1/2$, the cunning strategy is a best response if the cunning condition holds for the second proposer and:

$$\frac{f_{k-2} - f_{k-1}}{2} \geq \frac{ax}{27}, \tag{4}$$

where f_k denotes the transaction fees emitted at round k.

Proof. If the cunning condition is true for the second proposer ($w_g + a - w_f \leq \rho a$) it can attach its block to the branch with attestation weight w_f since the gap with the concurrent chain of weight $w_g + a$ is less than the proposer boost ρa. The obedient attesters of the second round will add an attestation weight of a to w_f. Following this, the gap between the attestation weights of the two concurrent branches will always remain less than ρa, leading all cunning proposers to attach their blocks two rounds prior.

We illustrate in Figure 5 the "bouncing" that will unfold due to cunning proposers. Their resulting reward will be affected, as the repeated bouncing of the canonical chain between the two branches will cause the blocks from each chain to become canonical with a probability of 1/2. No attesters will receive the maximum reward since they would never attest in accordance with the following proposer. The reward of the proposer $(0, k)$ following the cunning strategy will thus be:

$$\begin{aligned} u_{(0,k)}(\sigma_{-(0,k)}, \sigma^C_{(0,k)}) &= \frac{1}{2}\left(\frac{a}{7} \cdot \frac{20x}{27} + f_{k-2} + \frac{a}{7} \cdot \frac{20x}{27} + f_{k-1}\right) \\ &= \frac{a}{7} \cdot \frac{20x}{27} + \frac{f_{k-2} + f_{k-1}}{2}, \end{aligned} \tag{5}$$

with $\sigma_{-(0,k)}$ being the strategy profile in which every proposer is cunning and every attester is obedient.

Being cunning is a best response when $w_f + \rho a \geq a$, if and only if:

$$\begin{aligned} & u_{(0,k)}(\sigma_{-(0,k)}, \sigma^C_{(0,k)}) \geq u_{(0,k)}(\sigma_{-(0,k)}, \sigma^O_{(0,k)}) \\ \Leftrightarrow \quad & \frac{a}{7} \cdot \frac{20x}{27} + \frac{f_{k-2} + f_{k-1}}{2} \geq \frac{ax}{7} + f_{k-1} \\ \Leftrightarrow \quad & \frac{f_{k-2} - f_{k-1}}{2} \geq \frac{ax}{27}, \end{aligned} \tag{6}$$

where $\sigma_{-(0,k)}$ is the strategy profile in which every proposer is cunning and every attester is obedient, and $\sigma^O_{(0,k)}$ is the obedient strategy. Therefore, if f_{k-2} is not sufficiently greater than f_{k-1}, the best response is the obedient strategy; otherwise, the best response is the cunning strategy. ◀

▶ **Observation 13.** *Since transaction fees are positive, they cannot continue to decrease indefinitely with each round. This implies that when the cunning condition holds for the second proposer, eventually one proposer will follow the obedient strategy, thereby stopping the fork.*

Best response of single attester among n cunning proposers and $an - 1$ obedient attesters. Until now, we have described a scenario where all proposers follow the cunning strategy, and attesters follow the obedient strategy, leading to a potentially long fork in which attesters pay the price for the cunning behavior of proposers and do not receive the maximum reward. Let us now analyse the cunning attester strategy, which takes advantage of knowing when the cunning proposer strategy is the best response to act accordingly and secure a higher reward.

Our result yield that when $\rho < 1/2$, the obedient attester and the cunning attester strategy are equivalent, Proposition 14. This implies that attesters will follow the protocol when $\rho < 1/2$. Moreover, if $\rho \geq 1/2$, when all proposers are cunning and attesters are obedient, the best response is the cunning attester strategy. When the cunning condition holds for the second proposer the cunning attester strategy will deviate from the protocol. Otherwise, all attesters will follow the protocol (Proposition 15).

▶ **Proposition 14.** *When $\rho < 1/2$, the obedient attester strategy and the cunning attester strategy are equivalent.*

Proof. This result stems from the fact that when $\rho < 1/2$, only the first proposer can act conspicuously cunningly, meaning they deviate from the protocol (cf. proof of Proposition 11). Therefore, if all subsequent proposers act similarly to obedient proposers and follow the protocol, attesters will never have the opportunity to act cunningly and will follow the protocol as well. ◀

▶ **Proposition 15.** *When $\rho \geq 1/2$ and all proposers are cunning while all other attesters are obedient, the cunning attester strategy is a best response. If the cunning condition holds for the second proposer, the cunning attester strategy will lead the attester to deviate from the protocol. Otherwise, all attesters will follow the protocol.*

Proof. For attesters to exhibit cunning behavior, more than just the first proposer must act cunningly. This occurs if and only if the cunning condition hold for the second attester.

The reward for attester (i, k) following the cunning attester strategy $\sigma^C_{(i,k)}$, while all other attesters are obedient and proposers are cunning $\sigma_{-(i,k)}$, is:

$$u_{(i,k)}(\sigma_{-(i,k)}, \sigma^C_{(i,k)}) = \frac{47x}{54}. \tag{7}$$

Since the attester's reward depends on when their attestation is included in a block, it also depends on whether the blocks belong to the canonical chain. Each attestation is included in the next two blocks, which are on different chains, each having a $1/2$ probability of being in the canonical chain. This gives the cunning attester a reward of $\frac{1}{2}(x + \frac{20x}{27})$. Following the obedient attester strategy leads to a reward of $\frac{20x}{27}$, as in both blocks, the attestation will either attest to the wrong block or be included too late. Thus, $u_{(i,k)}(\sigma_{-(i,k)}, \sigma^C_{(i,k)}) \geq u_{(i,k)}(\sigma_{-(i,k)}, \sigma^O_{(i,k)})$. ◀

Best response of single attester among n cunning proposers and $an-1$ cunning attesters. We study the behavior of an attester when all proposers are cunning and other attesters cunning. In this case the best response is the cunning attester strategy. This strategy only deviates from the protocol for the attesters of the first round if the cunning condition holds for the second proposer. However this deviation will prevent the cunning condition to hold for the third proposer, effectively making all subsequent validators to follow the protocol.

▶ **Proposition 16.** *The cunning attester strategy is a best response for an attester when all validators are cunning. If the cunning condition holds for the second proposer, the cunning attester strategy will lead the attesters of the first round to deviate from the protocol. Otherwise, all attesters will follow the protocol.*

Proof. For attesters to exhibit cunning behavior, more than just the first proposer $(0,0)$ must act cunningly. This occurs if the cunning conditions holds for the second proposer $(0,1)$.

In this case, all attesters of the first round will expect proposer $(0,1)$ to act cunningly and attach itself to the block designated by the fork choice rule at the beginning of the game $(\mathcal{F}(\mathcal{T}_0, \mathcal{A}_{-1}))$, leading them to attest to the head of the branch with total attestation weight w_f. This scenario is represented in Figure 6. As a result, the block proposed by the first proposer $(0,0)$ will not be attested by the attesters. The block proposed by $(0,1)$ will belong to the canonical chain since no other fork is possible for the subsequent proposers. The gap between $w_f + a$ and w_g is too large for the proposer boost to enable further cunning actions, the cunning condition cannot hold anymore. The attesters $(i,0)$ of the first round, who act in accordance with proposer $(0,1)$, receive the maximum reward. Proposers $(0,2)$ and beyond will not have the opportunity to act cunningly, nor will the remaining attesters, resulting in all attesters receiving the maximum reward. ◀

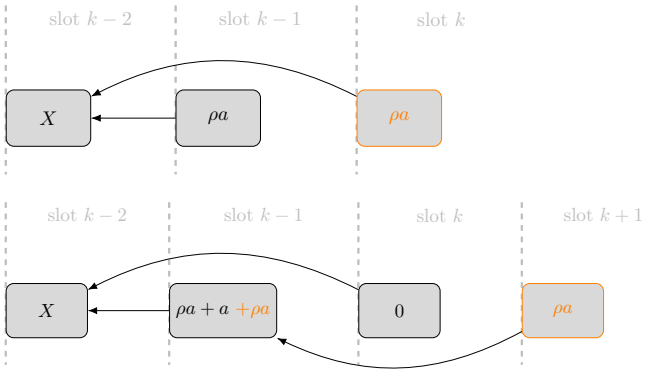

Figure 6 X indicates that the value of the block is irrelevant. In this scenario, all validators act cunningly. The proposer of round k attaches its block to the block from two rounds prior. The cunning attesters in round k attest to the block from round $k-1$ to align with the following proposer's strategy. The proposer of round $k+1$ is then compelled to attach its block to the block from round $k-1$. As a reminder, the proposer boost is equivalent to an attestation weight of ρa for a new block (in orange). This results in the proposer of round k forking alone and receiving no rewards.

▶ **Proposition 17.** *If $\rho < 1/2$ and all validators are cunning, the best response for the first proposer is the cunning strategy. If the cunning condition holds, it will only do so for the first proposer, causing this proposer to deviate from the protocol.*

Proof. This follows directly from Proposition 14 and Proposition 11. ◀

Proposition 8. If $\rho \geq 1/2$, all validators are cunning, and the cunning condition holds for the second proposer, then the obedient strategy is the best-response for the proposers. Otherwise, if the cunning condition does not hold for the second proposer, the cunning strategy is the best-response only for the first proposer.

Proof. If the cunning condition holds for the second proposer, Proposition 16 describes how the scenario would unfold. A possible outcome is represented in Figure 6. The result is that, for the first proposer, the best response is the cunning strategy only if the second proposer cannot act cunningly.

Otherwise, as described in the proof of Proposition 16, if the first proposer remains cunning, they will receive zero reward. All other proposers follow the protocol regardless of the strategy of the first proposer. ◀

Theorem 9. In all Nash equilibria, there is a round after which all validators follow the protocol.

Proof. We know that once a proposer acts obediently, all subsequent validators do so (Lemma 10).

If there is an equilibria in which one proposer follows the obedient strategy and extend the head of the canonical chain, the theorem is valid. We now look at proposers all following the cunning strategy. When all proposers follow the cunning strategy, to have more than the first to effectively act cunningly we need to have $w_g + a - w_f \leq \rho a$ (cunning condition for the second proposer) otherwise the second proposer will extend the head of the canonical chain, validating the theorem. Then in the case of the fork continuing with each proposer thus attaching their block two rounds prior, this makes each of their block have an expectation of $1/2$ to belong to the canonical chain ($\chi = 1/2$). As computed in Lemma 12, their reward will thus be in the form of: $f_{k-2} - f_{k-1} \geq \alpha$, where α is a positive number that depends on the attestation included and their reward associated. No matter the value of α, even taking $\alpha = 0$, a proposer will have as best response to be cunning only if the transaction fees gained with a probability $1/2$ by being cunning are at least superior to the transaction fees obtain with certainty otherwise.

This condition cannot be true for all proposer as the transaction fees are positive and discrete. Eventually, a proposer $(0, k)$ will see previous transactions fees where $f_{k-2} < f_{k-1}$. The best response of proposer $(0, k)$ is to act obediently. ◀