Undetectable Selfish Mining

MARYAM BAHRANI, a16z crypto, USA
S. MATTHEW WEINBERG, Princeton University, USA

Seminal work of Eyal and Sirer [2014] establishes that a strategic Bitcoin miner may strictly profit by deviating from the intended Bitcoin protocol, using a strategy now termed *selfish mining*. More specifically, any miner with $> 1/3$ of the total hashrate can earn bitcoin at a faster rate by selfish mining than by following the intended protocol (depending on network conditions, a lower fraction of hashrate may also suffice).

One convincing critique of selfish mining in practice is that the presence of a selfish miner is *statistically detectable*: the pattern of orphaned blocks created by the presence of a selfish miner cannot be explained by natural network delays. Therefore, if an attacker chooses to selfish mine, users can detect this, and this may (significantly) negatively impact the value of BTC. So while the attacker may get slightly more bitcoin by selfish mining, these bitcoin may be worth significantly less USD.

We develop a selfish mining variant that is provably *statistically undetectable*: the pattern of orphaned blocks is statistically identical to a world with only honest miners but higher network delay. Specifically, we consider a stylized model where honest miners with network delay produce orphaned blocks at each height independently with probability β'. We propose a selfish mining strategy that instead produces orphaned blocks at each height independently with probability $\beta > \beta'$. We further show that our strategy is strictly profitable for attackers with $38.2\% \ll 50\%$ of the total hashrate (and this holds for all natural orphan rates β').

CCS Concepts: • **Theory of computation → Algorithmic game theory**; **Algorithmic game theory and mechanism design**; • **Information systems → Digital cash**; • **Security and privacy → Pseudonymity, anonymity and untraceability**;

Additional Key Words and Phrases: blockchain, cryptocurrency, selfish mining, proof-of-work

ACM Reference Format:
Maryam Bahrani and S. Matthew Weinberg. 2024. Undetectable Selfish Mining. 1, 1 (June 2024), 28 pages. https://doi.org/10.1145/3670865.3673485

1 INTRODUCTION

At their core, cryptocurrencies and other blockchain protocols aim to provide a totally-ordered *ledger* of transactions that can serve as a record of payments. Key to their value proposition is that these ledgers are maintained in a *decentralized manner* — anyone can join their network and start authorizing transactions (in Bitcoin, this process is referred to as *mining* and the participants as *miners*). However, miners must be incentivized in order to authorize transactions, and in particular to follow the intended protocol.

The whitepaper introducing Bitcoin [Nakamoto, 2008] applies a clever technique termed *proof-of-work* — miners are selected to authorize the next block of transactions proportionally to their computational power (often referred to as their *hashrate*, because they use this computational power to repeatedly compute the hash function SHA-256). That same whitepaper already observes

S. Matthew Weinberg is supported by a Ripple UBRI Grant and NSF CAREER CCF-1942497.
Authors' addresses: Maryam Bahrani, maryam.bahrani.14@gmail.com, a16z crypto, New York, NY, USA; S. Matthew Weinberg, smweinberg@princeton.edu, Department of Computer Science, Princeton University, Princeton, NJ, USA.

EC '24, July 8–11, 2024, New Haven, CT, USA

ACM ISBN 979-8-4007-0704-9/24/07.
https://doi.org/10.1145/3670865.3673485

that incentivizing miners to follow the intended protocol can be tricky, however. For example, Nakamoto observes that a miner with $> 1/2$ of the total hashrate in the network can strictly profit by deviating from the Bitcoin protocol. However, for several years it was largely assumed that 50% was the cutoff: as long no miner controlled more than half the total hashrate in the network, everyone should follow the intended protocol.

Eyal and Sirer [2014] prove this assumption false with their seminal paper introducing the Selfish Mining strategy (we will overview their strategy in detail in Section 2). Their strategy is strictly profitable compared to honestly following the Bitcoin protocol even with just $> 1/3$ of the total hashrate. Followup work of Sapirshtein et al. [2016] later provides an improved strategy that is strictly profitable with just $\approx 32.9\%$ of the total hashrate, which is tight (for a strategic miner with poor network connectivity). With better network connectivity, Selfish Mining can be profitable with arbitrarily small fraction of hashrate — see discussion in Section 2.

From a mechanism design perspective, these facts initially appear somewhat alarming: Depending on the hashrate of the largest miner, and also on network conditions, following the protocol is not a Nash equilibrium. Moreover, the Selfish Mining attack is in a sense untraceable: because Bitcoin is pseudonymous, there is no way to distinguish among identities of block creators to determine which blocks were created by a dishonest miner. That is, depending on the hashrate of the largest miner and network conditions, there is an untraceable and strictly profitable deviation from the Bitcoin protocol.

However, one key critique of Selfish Mining (and other strategic deviations) in practice is its *statistical detectability*. That is, the pattern of blocks created in a world with a Selfish Miner does not have any "innocent explanation" due to network latency, and it will be clear that *someone* is Selfish Mining.[1] Should Selfish Mining be detected, it is possible that the value of the underlying cryptocurrency will suffer significantly. Therefore, while Selfish Mining may be profitable when denominated in the underlying cryptocurrency (e.g. BTC), it could be wildly unprofitable when denominated in an objective unit of value (e.g. USD).[2]

The **main result of our paper introduces a *statistically undetectable Selfish Mining* strategy** in Theorems 4 and 5. That is, the pattern of block creation induced by our strategy is statistically identical to that of a world with only honest miners but increased latency. The mathematics behind our results apply to the canonical setting studied in Eyal and Sirer's seminal work: proof-of-work longest-chain protocols with block rewards.[3] Still, we hope the reader takes away the following conceptual point much more broadly: attackers concerned about the impact of their attack on a cryptocurrency's value may seek an undetectable variant. Therefore, designers must consider this possibility when claiming incentive compatibility of protocols (and especially if those claims reference a strategic deviation's impact on the cryptocurrency's value).

We discuss the implications of our results in more detail in Section 6. After overviewing related work below in Section 1.1, we proceed with a detailed description of Selfish Mining, along with a description of our model in Section 2. We follow this with formal statements of our main results in Section 3. Section 4 proves a warmup result to demonstrate some key ideas, and Section 5 proves

[1] To clarify: it may not be clear who is Selfish Mining, nor how to punish them, but it will be clear that *someone* is Selfish Mining. Note also that collecting the relevant data needed to run the statistical test is non-trivial.

[2] Note that even if a deviator were to hedge against this possibility by shortselling the underlying cryptocurrency, there is still significant uncertainty in how the market might react (which may be sufficient to dissuade a risk-averse player, even if hedges exist to make deviating profitable in expectation).

[3] Our techniques may certainly be relevant more broadly to longest-chain protocols (e.g. with transaction fees instead of block rewards, or proof-of-stake instead of proof-of-work), simply because these settings have significant technical overlap. Still, we do not claim that any of our formal results carry to another setting, and applying our techniques in this setting may still require novel insight.

our first main result. Appendix A contains a summary of discussions regarding Selfish Mining in practice. Extensions of our main results and omitted proofs are contained in later appendices.

1.1 Related Work

1.1.1 Strategic Manipulations of Consensus Protocols. Following Eyal and Sirer [2014], there have been numerous papers on Selfish Mining. Most relevant to our work are those that understand the limits of its profitability, such as [Kiayias et al., 2016, Sapirshtein et al., 2016]. In analyzing our most general result, we use tools initially developed in [Sapirshtein et al., 2016].

Strategic manipulation of consensus protocols via some form of "block withholding" is studied in a wide array of related domains. Those most similar to our work consider deviations in proof-of-work longest-chain protocols with transaction fees [Carlsten et al., 2016], proof-of-stake longest-chain protocols [Brown-Cohen et al., 2019, Ferreira and Weinberg, 2021, Neuder et al., 2019, 2020], and leader-selection protocols [Ferreira et al., 2022]. Our work bears some similarity to these in that we also study strategic manipulation of consensus protocols via block withholding, but the models are distinct.

Other works examine different styles of profitable deviations from Proof-of-Work blockchain protocols. For example, [Fiat et al., 2019, Goren and Spiegelman, 2019, Yaish et al., 2022a,b] consider manipulating difficulty adjustments to earn extra rewards. [Yaish et al., 2022a,b] do so in particular by manipulating the timestamp of created blocks.

1.1.2 Strategic Manipulation in Practice. Among the broad line of works referenced above, [Yaish et al., 2022a] is especially impactful, as they also provide clear evidence of strategic manipulation in practice (which is the first such evidence, to the best of those authors' and these authors' knowledge). It is important for future work to understand the impact that their discovery had on the deviator (F2Pool) and/or the underlying currency (ETH) – this would quantitatively impact any risk assessment by miners considering strategic manipulation. In relation to our work, this would impact the importance of statistical undetectability to strategic miners.

We've informally used the term "strategic manipulation" to describe deviations from an intended consensus protocol that net the deviator additional rewards *issued directly by the consensus protocol* in the form of block rewards. There is also a significant history of block producers acting selfishly to profit from economic activity on top of consensus protocols (e.g. double-spend attacks.)

1.1.3 Incentives beyond the Consensus Layer. Beyond incentive compatibility concerns arising from block rewards, there is also a growing literature considering incentive design in other aspects of blockchain protocols. One example is recent work on transaction fee mechanism design (auction design for the inclusion of transactions within a block in the presence of strategic block producers) [Bahrani et al., 2023, 2024, Chung and Shi, 2023, Huberman et al., 2021, Lavi et al., 2019, Roughgarden, 2021, Shi et al., 2023, Yao, 2018]. These works also fit into a growing literature on mechanism design within consensus protocols, but otherwise have no technical overlap with ours.

1.1.4 Detection of Selfish Mining in practice. We are aware of two lines of work related to statistical detection of Selfish Mining. One proposes statistical tests (such as the length of orphaned chains – longer orphan chains are claimed to be indicative of Selfish Mining), and perhaps evaluates them in a simulation environment [Chicarino et al., 2020, Saad et al., 2019, Wang et al., 2021]. To the best of our knowledge, most of these tests have not been deployed in practice, and therefore do not offer an opinion on whether or not Selfish Mining is occurring in practice.

A smaller line of work aims to detect Selfish Mining empirically. [Li et al., 2022] claims to be the first paper to detect selfish mining in practice, building upon previous work of [Li et al., 2020a,b]. Their statistical test looks exclusively at the main chain, and focuses on the distribution

of consecutive blocks mined by the same miner. Their null hypothesis is that the miner of each block on the main chain should be independent of prior blocks, whereas a Selfish Miner would disproportionately mine their blocks consecutively. They claim that observed mining patterns fail their null hypothesis to varying degrees across five blockchains: Bitcoin, Litecoin, Ethereum PoW, Monacoin, and Bitcoin Cash. But, the authors acknowledge that their null hypothesis could fail for other reasons. For example, note that their null hypothesis would also fail when orphans naturally occur and miners tiebreak in favor of their own blocks (this is because a block competing in a natural fork is more likely to wind up in the longest chain when its creator finds the next block).

2 BACKGROUND

2.1 Nakamoto Consensus and the Longest Chain Protocol

Bitcoin and many other altcoins use a *longest-chain protocol* with *proof-of-work*, which is often referred to as *Nakamoto Consensus*. We overview relevant details of the protocol below, and refer the reader to resources such as [Narayanan et al., 2016] for an explanation of why these details capture the Bitcoin protocol. Note that the features we highlight below are exactly the same features from [Eyal and Sirer, 2014] — the stylized model we pose below is not new, and is extensively studied following [Eyal and Sirer, 2014].

Nakamoto Consensus Game (NCG). There is a set N of n miners, and miner i has hashrate $\alpha_i > 0$. Time proceeds in discrete steps $t = 1, 2, \ldots$. At the start of each timestep t, there is a set of blocks B_t that have been previously broadcast, and a set of blocks B_t^i that have been previously created by miner i (perhaps broadcast, perhaps not). Initially, B_1^i is empty for all i, and B_1 contains a single "genesis block." At every time step t:

- A miner m_t is selected so that $m_t := i$ with probability $\alpha_i/(\sum_{j=1}^n \alpha_j)$, independently of all previous rounds. m_t creates a *block* b_t, and m_t chooses the block to which b_t *points*: m_t may choose any block in $B_t \cup B_t^{m_t}$ (that is, the newly created block must point to exactly one block, and that block can be any block that was broadcast before time t, or created by m_t before time t, or both).
- Every miner i can choose to broadcast any blocks in $B_{t+1}^i \setminus B_t$ (that is, every miner i can broadcast any block that they created in a timestep $\leq t$ and have not already broadcast).

We refer to the *height* of a block $h(b_t)$ as the number of blocks in the path leaving b_t (i.e. by following pointers). A *Longest Chain* at time t is any block in B_t of greatest height. If the longest chain at time t is unique, we denote by R_t^i to be the fraction of blocks in the longest chain that were created by miner i. If the longest chain at time t is not unique, let $t' < t$ denote the most recent timestep where the longest chain at t' is unique, and define $R_t^i := R_{t'}^i$. Observe that the longest chain at time 1 is unique, so this is always well-defined. Player i's reward is $\liminf_{t\to\infty} R_t^i$.[4]

In particular, observe in this game that every miner has two decisions every round: (a) if they create a block, what does it point to? and (b) what previously-created blocks do they broadcast? Observe also that miners are paid proportionally to the steady-state fraction of blocks they produce in the longest-chain — this captures the fact that miners are paid primarily according to a block reward that comes with each block, and that Nakamoto Consensus has a difficulty adjustment that causes the length of the longest chain to grow proportionally to time (we again refer the reader to [Eyal and Sirer, 2014, Narayanan et al., 2016] for further details on connecting this game to Bitcoin and related cryptocurrencies).

[4]Our work and all prior works only consider strategy profiles where the limit exists, so the distinction between lim and lim inf is just a formality.

Longest Chain Protocol. The longest-chain protocol asks each miner to use a specific strategy in this game: Whenever you create a block, point to a longest chain, and immediately broadcast your block. We call a miner *honest* if they follow the longest-chain protocol, and *strategic* otherwise. The key question is whether it is a Nash equilibrium for every miner to follow the longest-chain protocol.

It is folklore knowledge from Nakamoto's whitepaper [Nakamoto, 2008] that the longest-chain protocol is not a Nash equilibrium if $\alpha_i > \sum_j \alpha_j/2$ for any i. In that case, Player i can simply ignore all blocks created by other miners and point only to their own most recent block (broadcasting all blocks upon creation). Then Player i will have reward 1, and all other players have reward 0.[5] However, until Eyal and Sirer's seminal work, it was assumed that the longest-chain protocol is indeed a Nash equilibrium when $\alpha_i < \sum_j \alpha_j/2$ for all i.

2.2 Selfish Mining

We now overview the Selfish Mining strategy, and review its performance when all other miners are using the longest-chain protocol. Let's first consider the case where all other miners break ties in favor of the strategic miner i (that is, if the longest chain is not unique, they will point their block towards the one that is created by miner i). Consider the following strategy in NCG:

Strong Selfish Mining:

- When selected to mine a block, point to a longest-chain, breaking ties in favor of a block created by yourself.
- *Do not immediately broadcast this block.* Instead, during any round t where you are not selected to mine, and the longest chain before round t has height h, broadcast a block of height $h + 1$ (if you have one).

If a strategic miner i uses Strong Selfish Mining and all other miners follow the longest chain protocol, then the following occurs. First, *every* block created by miner i is broadcast during the same round as a block created by another miner of the same height. In other words, every height either contains a single block created by a miner $\neq i$, or two conflicting blocks broadcast during the same round, with one created by miner i. Second, because every miner tiebreaks in favor of miner i, all of i's blocks will enter the longest chain (and conflicting blocks will not). Therefore:

Theorem 1 ([Eyal and Sirer, 2014]). *If all miners $\neq i$ use a longest-chain protocol that tiebreaks in for Miner i, and Miner i uses Strong Selfish Mining, then Miner i gets reward $\frac{\alpha_i/\sum_j \alpha_j}{1-\alpha_i/\sum_j \alpha_j} > \alpha_i/\sum_j \alpha_j$.*

Strong Selfish Mining is strictly profitable for any $\alpha_i > 0$, but relies on the fact that all other miners tiebreak in its favor. Eyal and Sirer connect tiebreaking in the above-described Nakamoto Consensus Game to network connectivity in practice. Really, a "timestep" in the Nakamoto consensus game corresponds to "some miner has just succeeded in inverting SHA-256 and created a new block." An extremely well-connected miner can (a) listen for this block to be broadcast, and then (b) broadcast their own previously-withheld block, while still (c) ensuring that their own block arrives at other miners first. Mapping this onto the stylized Nakamoto consensus game yields Strong Selfish Mining. For the rest of this paper, we'll replace "during any round t where you are not selected to mine, and the longest chain before round t has height h" with "during any round t where another miner broadcasts a block of height $h + 1$" to emphasize this connection (and so that our language sounds less tailored to this specific stylization).

[5]Note that, beyond the fact that Player i is accumulating all of the mining rewards, they control the entire content of the ledger, and can launch significantly more malicious attacks.

Of course, it is not realistic to assume that any strategic miner has such strong network connectivity. Eyal and Sirer further pose the (canonical) Selfish Mining strategy that is profitable even when the strategic miner *loses all tiebreaks* — this corresponds to a reality where the poorly-connected miner can still execute (a) and (b), but not (c). Instead, their canonical strategy (paraphrased below) relies only on the fact that honest miners will still always select a strictly longer chain over a strictly shorter one.

Selfish Mining:

- When selected to mine a block, point to a longest-chain, breaking ties in favor of a block created by yourself.
- *Do not (necessarily) immediately broadcast this block.* Instead, for each block b of height h that you create:
  - At the moment where another miner broadcasts a block of height h, broadcast b.
  - Or, during the moment/round where b becomes *pivotal*, broadcast b. A block of height h is pivotal if (a) a block of height $h-1$ has been broadcast by another miner, (b) you created a block of height $h-1$, and (c) you have not yet created a block of height $h+1$.

If a strategic miner i uses Selfish Mining, and all other miners follow the longest-chain protocol, then the following occurs. At all points in time, there may be a unique longest chain with height h. If so, and a miner $\neq i$ creates the next block, it will be broadcast and the height increases to $h+1$. If instead i creates the next block, i will withhold the block and eventually create a conflict at height $h+1$. If there are multiple longest chains with height h, then miner i may have created blocks that haven't been broadcast. If miner i has only one such block, then it is pivotal, and miner i will immediately broadcast it (to ensure that it wins the conflict, because it can only win with a strictly longer chain). If miner i has multiple such blocks, it can safely wait until broadcasting, as long as they broadcast their last hidden block the moment it becomes pivotal. Intuitively, the Selfish Miner faces a risk compared to the longest-chain protocol: when withholding their first block, there is a chance that they do not find either of the next two blocks. In this case, their block is orphaned.[6] If they find exactly one of the next two blocks, then this block becomes pivotal and broadcast — this allows the Strategic Miner to orphan another miner's block (thus increasing its own ratio). If instead they find both of the next two blocks, then they now have a lead of three withheld blocks, and will be able to orphan even more blocks of other miners (thus further increasing its own ratio). Because there is a risk, the strategy is only profitable for sufficiently large α_i.

Theorem 2 ([Eyal and Sirer, 2014]). *If all miners $\neq i$ use a longest-chain protocol that tiebreaks against Miner i, and Miner i uses Selfish Mining, then Miner i gets reward $> \alpha_i$ if and only if $\alpha_i > 1/3$.*

2.3 Detecting Selfish Mining

Because Bitcoin is pseudonymous, an attacker can create a new public key for every block they create, so there is no record of the same miner creating multiple blocks. With no identity attached to any particular block, there is little to distinguish the Selfish Miner's blocks from others.[7]

[6] A block is "orphaned" if it is not in the eventual longest chain.

[7] But they are not *fully* indistinguishable. The Selfish Miner's blocks will always be created before the competing blocks, and this can manifest in a few ways (including timestamps and transaction content). One might therefore propose to mitigate the profitability of Selfish Mining by asking honest miners to tiebreak in favor of blocks that were created most recently. Unfortunately, this creates significant incentive issues (that designers are well aware of): now an attacker need not build upon the current longest-chain because they can just replace it instead. In general, targeted punishments against blocks that could have been created by Selfish Mining may be worthwhile to explore, but this is very far from current norms (and may be impossible to do without creating new incentive issues).

While a Selfish Miner may not be identifiable, their presence is statistically detectable. To help make this point more rigorous, consider the following generalized Nakamoto consensus game, parameterized by a latency ℓ.[8]

Nakamoto Consensus Game, parameterized by ℓ (ℓ-NCG). The Nakamoto Consensus Game, parameterized by $\ell > 0$ is identical to the Nakamoto Consensus Game *except* in who is selected to mine during a step t. Instead of picking a miner proportional to their hashrate, a coin is flipped independently for each miner i that is heads with probability $\ell \cdot \alpha_i$ (we will only ever consider games where $\ell \leq \max_j\{1/\alpha_j\}$). The set of coins is repeatedly flipped until at least one lands heads. Then, all miners whose coins are heads produce a block this round.

As $\ell > 0$ approaches 0, the distribution of miners selected to create blocks in each round approaches that of the original Nakamoto Consensus Game, so we formally define 0-NCG as the original NCG (where exactly one miner is selected each round).

ℓ-NCG is a natural extension of the original stylized model to incorporate a stylized model of latency. Even if every miner is honestly following the longest-chain protocol, there will inevitably be conflicts and orphaned blocks (for example, during any round in which there are multiple miners). We now build up language to formalize what we mean by statistical undetectability.

Definition 1 (View of a Nakamoto Consensus Game). *As a Nakamoto Consensus Game is played, we refer to the* view *as the collection of all blocks that are eventually broadcast (treated as nodes in a directed graph — the only information in the view is the pointer). We also refer to the* view up to height h *as the collection of all blocks of height at most h that are eventually broadcast. Observe that for any ℓ-NCG and any strategy profile, the view is drawn according to some distribution.*[9]

Definition 2 (Statistically Undetectable Deviant Strategy). *We say that a strategy s for miner i in ℓ-NCG is ℓ'-*statistically undetectable *with respect to $\vec{s}_{-i}$ if the view of the game when Miner i uses s and and all other miners use $\vec{s}_{-i}$ is identically distributed to the view of the ℓ'-NCG when Miner i uses some longest-chain strategy and all other miners use $\vec{s}_{-i}$.*

We say that an ensemble of strategies $\{s^\varepsilon\}_{\varepsilon>0}$ is statistically undetectable *with respect to $\vec{s}_{-i}$ if for all $\varepsilon > 0$ there exists an $\ell' \in (\ell, \ell + \varepsilon)$ such that s^ε is ℓ'-statistically undetectable with respect to $\vec{s}_{-i}$.*

In all applications of these definitions, each s_j will be a longest-chain strategy (but may tie-break differently for different applications).

Note that in order to profit, Attacker must create orphans even if the base game is NCG. Therefore, in order for the view to look consistent with fully honest participants, Attacker will need to target an $\ell' > 0$ (and hence, even if one primarily wishes to study the original NCG as the "real world", one must study $\ell' > 0$ for the world created by Attacker).

Let us now quickly understand why both Selfish Mining Strategies are statistically detectable. Both claims will use the following notation (which our later proofs will also use) – the purpose of this is to ease notational burden and support a reduction from analyzing the full n-player game to a 2-player game. To ease notation here, and in the rest of the paper, we will w.l.o.g. always consider Player 1 to be the strategic player.

[8] Appendix B contains a brief discussion relating this stylized latency model to richer latency models considered in Distributed Systems.

[9] Note that while view does contain orphaned blocks, nodes do not propagate already-orphaned blocks for them to enter the view – nodes only propagate blocks that currently appear to be in the longest chain but might later be orphaned (nodes who are interested to detect selfish mining then store them in their own view, but do not propagate them once orphaned). Indeed, every block ever broadcast (and added to our view) is broadcast only during a period when it is believed to be in the longest chain.

Definition 3 (Single/Pair). *We say that height h in a view of a Nakamoto Consensus Game has state Pair if there is a block of height h created by Player 1, and also a block of height h created by a Player > 1 (possibly multiple players). Otherwise, h has state Single.*

Definition 4 (SP-Simple). *Consider any execution of a Nakamoto Consensus Game where all other players use the same deterministic longest-chain strategy. Consider also modifying the execution so that in every round, if at least one Miner > 1 is selected to create a block, instead only Miner 2 creates a block. A strategy for Miner 1 is* SP-Simple *if it takes identical actions in both games.*

That is, when playing against a profile of identical deterministic longest-chain strategies, an SP-Simple strategy is agnostic to how many blocks are created by other miners during a particular round, or which miner created them, given that some other miner created a block during this round.

Proposition 1. *Let s be any SP-Simple strategy for Miner 1 and $\vec{s}_{-1}$ be a profile where all other players use the same longest-chain strategy s', and s' tiebreaks in either lexicographical or reverse lexicographical order.*[10] *Then for all ℓ, ℓ', s is ℓ'-statistically undetectable for ℓ-NCG with hashrates $\vec{\alpha}$ with respect to $\vec{s}_{-i}$ if and only if s is ℓ'-statistically undetectable for ℓ-NCG with hashrates $\langle \alpha_1, \frac{1-\prod_{i=2}^{n}(1-\alpha_i \cdot \ell))}{\ell} \rangle$ with respect to s'.*

Moreover, Miner 1's reward using s in ℓ-NCG with hashrates $\vec{\alpha}$ against $\vec{s}_{-i}$ is equal to that when using s in ℓ-NCG with hashrates $\langle \alpha_1, \frac{1-\prod_{i=2}^{n}(1-\alpha_i \cdot \ell))}{\ell} \rangle$ against s'.

A proof of Proposition 1 appears in Appendix D. Intuitively, Proposition 1 follows by observing that a single Player 2 produces a block in the proposed two-player game with the same probability that any player > 1 produces a block in the original game. Proposition 1 allows us to focus just on the two-player game, which has significantly simpler notation.

Proposition 2. *For any $\vec{\alpha}$, and any player i, Strong Selfish Mining is* not *ℓ'-statistically undetectable for any ℓ' in the Nakamoto Consensus Game with respect to the strategy profile where other players use longest-chain and tiebreak in favor of i.*

Proof. We use Proposition 1 and prove the claim for every two-player game. Indeed, observe that in any two-player game parameterized by ℓ, if we let $\beta' = \frac{\alpha_1 \cdot \ell \cdot \alpha_2 \cdot \ell}{1-\alpha_1 \cdot \ell \cdot \alpha_2 \cdot \ell}$ denote the probability that both players produce a block in the same round, then height h is Pair with probability β' independently for all h.

If instead we consider a Nakamoto Consensus Game where Miner 1 uses Strong Selfish Mining, then observe that height h is Single if and only if Miner 2 produces the first block at height h, and is Pair if and only if Miner 1 produces the first block at height h.

So consider any height h such that h is Single. This means that Miner 2 produced the first block at height h and immediately broadcast it. Therefore, Miner 1 produces the first block at height $h+1$ with probability $\alpha_1/(\alpha_1+\alpha_2)$, and height $h+1$ has state Pair with exactly this probability. If instead h is Pair, then Miner 1 has (at least) one withheld block, and Miner 1 produces the first block of height $h+1$ with probability $> \alpha_1/(\alpha_1+\alpha_2)$ (because Miner 2 would need to produce at least the next two blocks in order to produce the first block of height $h+1$). Therefore, the probability of seeing Single vs. Pair at height $h+1$ depends on whether height h is Single vs. Pair, and this cannot be identically distributed to honest parties in a Nakamoto consensus game with any latency. □

Intuitively, Proposition 2 observes that Strong Selfish Mining is more likely to produce a Pair when it has just produced a Pair (because it must already have hidden blocks at the moment a height is determined to be Pair).

[10] That is, s' either always tiebreaks in favor of 1, or never in favor of 1.

Proposition 3. *For any $\vec{\alpha}$, and any player i, Selfish Mining is* not *ℓ'-statistically undetectable for any ℓ' in the Nakamoto Consensus Game with respect to the strategy profile where other players use longest-chain and tiebreak against i.*

Proof. We use Proposition 1 and prove the claim for every two-player game. Indeed, observe that in any two-player game parameterized by ℓ, if we let $\beta' = \frac{\alpha_1 \cdot \ell \cdot \alpha_2 \cdot \ell}{1-\alpha_1 \cdot \ell \cdot \alpha_2 \cdot \ell}$ denote the probability that both players produce a block in the same round, then height h is Pair with probability β' independently for all h.

If instead we consider a Nakamoto Consensus Game where Miner 1 uses Selfish Mining, then observe that height h is Single if and only if Miner 1 has no hidden blocks when the block of height h is broadcast. In particular, if two consecutive states $(h, h+1)$ are (Single, Pair), it must be that Miner 1 has no hidden blocks at height h, and $h+1$ is determined to be state Pair as soon as Miner 1 finds the next block. From here, state $h+2$ is Pair if and only if Miner 1 finds the next block, which occurs with probability $\alpha_1/(\alpha_1+\alpha_2)$.

If instead two consecutive states $(h, h+1)$ are (Pair, Pair), it must be that Miner 1 found a block at height $h+2$ before Miner 2 found a block at height h, state $h+1$ is determined to be Pair the moment this happens. From here, state $h+2$ is Pair with probability $> \alpha_1/(\alpha_1+\alpha_2)$ (because Miner 2 would need to produce at least the next two blocks in order for Miner 1's block at height $h+2$ to become pivotal). Therefore, the probability of seeing Single vs. Pair depends on whether the previous two heights are (Single, Pair) vs. (Pair, Pair), and this cannot be identically distributed to honest parties in a Nakamoto Consensus Game with any latency. □

Importantly, the above propositions note that Selfish Mining is statistically detectable *only by looking at the view of the game.* In particular, it does not require timestamping information (which can easily be manipulated [Yaish et al., 2022a,b] and intentionally has minimal role in the consensus protocol), or real-time monitoring of the network (the entire purpose of a consensus protocol is to cope with the fact that messages arrive at unpredictable times due to latency). Still, we briefly discuss in Section 6 alternative detection methods based on these.

3 MAIN RESULTS AND TECHNICAL OUTLINE

We now state our main results, which provide a statistically undetectable and profitable Selfish Mining strategy. All of our strategies have the following format: they begin with a strictly profitable selfish mining strategy (for our warmup, Strong Selfish Mining. For our first main result, Selfish Mining. For our second main result, a natural extension of Selfish Mining when there are sometimes natural forks), and sometimes "wastes" a hidden lead in order to preserve undetectability. That is, when a strategic miner has (say) five hidden blocks, they can reap the greatest profit by causing a fork with at least the first four. But, this creates a lot of sequential orphans. Our strategies will sometimes broadcast some of these blocks without creating an orphan. This sacrifices profit, but enables the strategy to appear indistinguishable from latency. The challenge is finding an appropriate broadcast pattern to be undetectable, and also analyzing when such patterns are still strictly profitable. We first begin with a warmup theorem that demonstrates some key ideas.

Theorem 3. *Let $\vec{s}_{-1}$ be the strategy profile where every miner uses longest-chain and tiebreaks for Miner* 1*. Then for any $\vec{\alpha}$, there is a statistically undetectable ensemble of strictly profitable strategies for Miner* 1 *in the Nakamoto Consensus Game with respect to $\vec{s}_{-1}$.*

We provide a complete proof of Theorem 3 in Section 4. We give quantitative bounds on the profitability of the strategy in Lemma 3. We then prove our first main result:

Theorem 4. *Let $\vec{s}_{-1}$ be the strategy profile where every miner uses longest-chain and tiebreaks against Miner* 1*. Then for any $\vec{\alpha}$ with $\alpha_1 \geq \frac{3-\sqrt{5}}{2} \cdot \sum_j \alpha_j$,[11] there is a statistically undetectable ensemble of strictly profitable strategies for Miner* 1 *in the Nakamoto Consensus Game with respect to $\vec{s}_{-1}$.*

Our bound of $\frac{3-\sqrt{5}}{2}$ is not tight for our strategy, although we provide a complete description of an infinite Markov chain that can be simulated in order to nail the tight bound to higher precision.[12] Interestingly, even our loose bound is not too far from the 1/3-fraction of hashrate that is required for Selfish Mining to be profitable without concern for undetectability. This establishes that adding undetectability to a profitable strategy (if possible) need not significantly detriment its performance.

We include a complete proof of Theorem 4 in Section 5. Quantitative bounds on the profitability of this strategy are given in Claim 3 and Lemma 4. Finally, we extend Theorem 4 to begin from any parameterized Nakamoto Consensus Game.

Theorem 5. *Let $\vec{s}_{-1}$ be the strategy profile where every miner uses longest-chain and tiebreaks against Miner* 1*. Then for any $\vec{\alpha}$ with $\alpha_1 \geq 0.382 \cdot \sum_j \alpha_j$, there is a statistically undetectable ensemble of strictly profitable strategies for Miner* 1 *in the Nakamoto Consensus Game parameterized by ℓ with respect to $\vec{s}_{-1}$.*[13]

At the end of Section 5, we overview ways in which the proof of Theorem 5 requires additional complexity, and defer a complete proof to the full version of the paper.[14] In summary, Theorem 5 establishes that a strategic miner can strictly profit against honest miners who tiebreak against them in a way so that the view appears as though all miners are honest but with an arbitrarily small increase to latency.

4 WARMUP: TIEBREAKING IN FAVOR OF STRATEGIC MINER

In this section, we prove Theorem 3, which introduces some of the key ideas of our analysis.

For simplicity of notation in this section and in all remaining technical sections, we let $n = 2$ and $\alpha := \alpha_1/(\alpha_1 + \alpha_2)$. This is w.l.o.g. due to Proposition 1. We will also refer to Miner 1 as Attacker and Miner 2 as Honest. We first remind the reader of the concept of a state, specialized to $n = 2$.

Definition 5 (State). *When $n = 2$, observe that the* state *of a height h, $S(h)$, is Pair if and only if both players create a block of height h, and Single if only one player creates a block of height h. We let $S(0) =$ Single, namely there is a unique genesis block.*

We will describe our strategy first as labeling all heights as Single or Pair, and then describe the actual broadcasting strategy to match it. The outline of this section is as follows: Definition 6 describes the labeling strategy. Lemma 1 specifies how the labeling strategy can be implemented via a broadcasting schedule. Lemma 2 argues that this strategy achieves undetectability. Finally, Lemma 3 argues that the strategy is strictly profitable.

[11] Note that $\frac{3-\sqrt{5}}{2} \lessapprox 38.2\%$.

[12] We suspect our bound is not far from tight, however.

[13] We derive a (significantly) more precise sufficient condition on α, β' in order for a strictly profitable and undetectable strategy to exist (see Proposition 5 of the full paper). Theorem 5 follows by verifying that this condition holds for all $\alpha \geq 0.382$ and all $\beta' \in [0, 1]$. It could be that our sufficient condition also holds for all $\alpha \geq \frac{3-\sqrt{5}}{2}$ and all $\beta \in [0, 1]$, but our computational software is not precise enough to verify this. Recall that $\frac{3-\sqrt{5}}{2} \lessapprox 0.382$, so the gap in analysis for $\beta' > 0$ and $\beta' = 0$ is fairly small.

[14] The full version can be found at https://arxiv.org/abs/2309.06847. Appendix E of that version proves Theorem 5, and Appendix D gives a simple analysis of when selfish mining is strictly more profitable than honest mining.

Recall that our goal is to produce a view so that each height is Pair with probability exactly β, for some $\beta := \frac{\alpha \cdot \ell' \cdot (1-\alpha) \cdot \ell'}{1-(1-\alpha) \cdot \ell' \cdot \alpha \cdot \ell'}$,[15] and some ℓ' arbitrarily close to 0 (which corresponds to β arbitrarily close to 0 as well).

Definition 6 (Labeling Strategy). *Define P_h to be the probability that Attacker creates the first block of height h, conditioned on all information available as of the moment that the first block of height $h-1$ is created. Then:*

- *If Honest creates the first block at height h, label h as Single.*
- *If Attacker creates the first block at height h, label h as Pair with probability β/P_h, and Single otherwise.*

Observe that Attacker can easily compute P_h as a function of the number blocks it is hiding: If Attacker has i hidden blocks at the moment the first block of height $h-1$ is created, Attacker will mine the first block of height h unless Honest mines $i+1$ blocks in a row. That is, $P_h = 1-(1-\alpha)^{i+1}$. Observe also that this expression is at least α for all $i \geq 0$, so the labelling strategy is valid (that is, β/P_h is a valid probability) as long as $\beta \leq \alpha$.

Next, we need to define how to implement the proposed labeling strategy with an actual broadcasting strategy in the game. In particular, we need to specify a broadcasting schedule that realizes the labels output by the labeling strategy.

Lemma 1 (Implementability). *The following* broadcasting strategy *realizes the labels output by the labeling strategy in Definition 6:*

- *If $S(h)$ is labeled as Single (and Attacker produced a block of height h): broadcast the block of height h the moment a block of height $h-1$ is broadcast (which might mean broadcasting the moment that Attacker produces a block of height h).*
- *If $S(h)$ is labeled as Pair: broadcast the block of height h the moment an Honest block of height h is published.*

Proof. Clearly, if this can be implemented, it will cause the state of every block to match its label (because there will clearly not be a chance for Honest to conflict with the Single broadcasts, and Honest clearly will conflict with the Pairs).

Moreover, this strategy can be implemented: the state of h is determined the moment a block of height h is created. If you create a block of height h, you learn its state immediately, and can either broadcast it immediately (if labeled Single), or broadcast once contested (if labeled Pair). □

Lemma 2 (Undetectability). *The broadcasting strategy of Lemma 1 applied to the labeling strategy of Definition 6, for any $\beta \leq \alpha$, played against a longest-chain strategy that breaks ties in favor of Attacker, produces a view where each height has state Pair independently with probability β.*

Proof. Observe that at the moment that a block of height $h-1$ is first created, $S(h)$ is equal to Pair with probability exactly β. This is true not only conditioned on $S(1), \ldots, S(h-1)$, but also conditioned on any additional information known at the instant $S(h-1)$ is fixed. Therefore, for all $S(1), \ldots, S(h-1)$, the probability that $S(h)$ is Pair conditioned on $S(1), \ldots, S(h-1)$ is exactly β. Therefore, the distribution of states is independent each round, and equal to Pair w.p. β. □

Lemma 2 concludes undetectability, which holds for β arbitrarily close to 0 (and therefore ℓ' arbitrarily close to 0 as well).

[15] This computation of β follows by observing that in a two-player ℓ'-NCG, both players produce a block during the same timestep with probability $\frac{\alpha \cdot \ell' \cdot (1-\alpha) \cdot \ell'}{1-(1-\alpha) \cdot \ell' \cdot \alpha \cdot \ell'}$, independently.

Lemma 3 (Profitability). *The broadcasting strategy of Lemma 1 applied to the labeling strategy of Definition 6, for any $\beta \leq \alpha$, when played against a longest-chain strategy that breaks ties in favor of Attacker, achieves reward $\alpha + \alpha\beta > \alpha$.*

Proof. The proof is a direct application of Lemma 4 (stated and proved immediately below in Section 4.1), observing that the strategy wins all Pairs. □

Proof of Theorem 3. The proof follows from Proposition 1, Lemma 2, and Lemma 3. □

4.1 Technical Lemmas

Below are two technical lemmas that we use in this section and will reuse in later sections as well.

Lemma 4. *Let $n = 2$, and consider any strategy that eventually broadcasts all blocks when playing against a longest-chain. Then, if this strategy wins a δ fraction of Pairs, and Pairs occur in a β fraction of rounds, it achieves expected reward:*

$$\alpha - (1 - \alpha - \delta) \cdot \beta.$$

Proof. Say that the strategy results in a β fraction of rounds having Pairs. Observe that an α fraction of all blocks are created by Attacker, and a $(1-\alpha)$ fraction of all blocks are Honest. Observe further that every Single round has exactly one block, and every Pair round has one block from each creator. Therefore, we see that $\alpha \cdot (1 + \beta)$ blocks must be produced by Attacker per-round. Moreover, β blocks per round of Attacker are in Pair rounds, leaving $\alpha \cdot (1 + \beta) - \beta$ blocks per round that are Attacker in Single rounds. This means that Attacker wins $\alpha \cdot (1 + \beta) - \beta$ fraction of rounds because they are Attacker Single, and an additional $\delta \cdot \beta$ fraction of rounds because they win δ fraction of Pairs. □

Corollary 1. *An SP-Simple strategy that eventually broadcasts all blocks and wins a δ fraction of Pairs achieves reward strictly better than α if and only if $\delta > 1 - \alpha$.*

Proof. Observe that $\alpha - (1 - \alpha - \delta) \cdot \beta > \alpha \Leftrightarrow -(1 - \alpha - \delta) > 0 \Leftrightarrow \delta > 1 - \alpha$. □

5 MAIN RESULT I: TIEBREAKING AGAINST STRATEGIC MINER

Here, we prove our first main result: a profitable and statistically undetectable Selfish Mining strategy when all miners tiebreak against the strategic miner. In the same canonical model where Selfish Mining was first introduced, we show that a modified strategy is strictly profitable and also statistically undetectable. We begin with some concepts, and will also reuse concepts/definitions from our warmup.

5.1 Concepts

Definition 7 (Pivotal Block). *A block of height h created by Attacker is* pivotal *if height $S(h-1)$ is Pair, and when the honest block of height $h-1$ is broadcast, Attacker does not have a block of height $h+1$. That is, Attacker's block of height h is pivotal if there is a conflict at height $h-1$, and Attacker needs to use this block in order to win it.*

Definition 8 (Safe Block). *A block created by Attacker is* safe *if it is not pivotal. Observe that a block of height h can be safe if either $S(h-1)$ is single, or if Attacker finds a block of height $h+1$ before Honest finds a block of height $h-1$.*

Intuitively, just like Selfish Mining, we really want to broadcast all pivotal blocks immediately, so that we don't risk losing a conflict that we can win right now. If a block is safe, we don't need to broadcast it immediately, but our strategy may choose to do so anyway to maintain undetectability.

5.2 The Strategy

We again first describe the strategy as labeling all heights as Single or Pair, and then describe the actual broadcasting strategy to match it.

Definition 9 (Labeling Strategy). *Define P_h to be the probability that Attacker creates a block of height h and that block is safe, conditioned on all information available* as of the first moment we know $S(h')$ *for all $h' < h$.*[16] *The labeling is strategy is then as follows:*

- *If Honest first creates a block at height h, then h is labeled Single.*
- *If Attacker first creates a block at height h, and it is pivotal, then h is labeled Single.*
- *If Attacker first creates a block at height h, and it is safe, then h is labeled Pair with probability β/P_h and Single otherwise.*

In particular, the instant that we have the necessary information to label a height, we label it (and the instant we have all the necessary information to know whether to flip a coin, we flip it).

Lemma 5 will ensure that the coin flip probabilities in the third bullet are valid. Before that, we will first show the following useful claim about the moment when there is enough information to decide the label of a given height. Importantly, we *don't* necessarily know $S(h)$ the moment its created (unlike in the warmup, where this information was sufficient), so we have to carefully track when it is first determined, and the information available at that time.

Claim 1 (When labels are determined). *For all h, there is sufficient information to determine $S(h)$ by the moment that a block of height $h+1$ is created (possibly sooner).*

Proof. We'll proceed by induction on h. The base case holds vacuously when $h = 0$, since the genesis block is Single by definition.

Let's first consider the moment that the block of height h is created. If the first block of height h is created by Honest, then h is certainly Single (and we learn this immediately when the first block of height h is created, which is before the first block of height $h+1$ is created).

If instead, the first block of height h is created by Attacker, then there are a few cases to consider. Recall that by the inductive hypothesis, we certainly know the labels of all blocks of height $h' < h$. This in particular implies that we have all the necessary information to compute P_h, and can already flip any coins that might be needed to label h. The only remaining uncertainty is around whether h is safe or pivotal. Zooming in on $S(h-1)$:

(a) If $S(h-1)$ is Single, then h is safe (by definition), and we learn this the moment that the first block at height h is created.

(b) If $S(h-1)$ is Pair, and Honest has broadcast a block of height $h-1$, then h is pivotal, and we learn this the moment h is created, so we can label $S(h)$ as Single.

(c) If $S(h-1)$ is Pair, and Honest has *not* yet broadcast a block of height $h-1$, then we do not yet know whether h is safe or pivotal. We learn this the moment either of the following events happens (whichever comes first):
   - Attacker finds another block (which will be at height $h+1$); then h is safe.
   - Honest finds a block at height $h-1$ (which may or may not require Honest to find multiple blocks); then h is pivotal.

   In particular, by the time there is a block of height $h+1$, we certainly know which case occurred first, and thus know whether h is safe or pivotal. From here, we can flip the necessary coins to label $S(h)$. □

Corollary 2. *For all h, P_h can be evaluated by the time the first block of height h is created.*

[16] The value of P_h is easily computable; see Table 2 for explicit expressions.

Lemma 5 (Validity). *The probabilities in Definition 9 are valid for any $\beta \leq \alpha^2$.*

Proof. We will show that for all h, we have $P_{h+1} \geq \alpha^2$, which completes the proof.

Let us again focus on the moment a block of height h is created for the first time. Consider the bullets in the proof of Claim 1. In bullets (a) and (b), we learn $S(h)$ the moment that the block at height h is created. Then if Attacker creates the next two blocks, they will have created a safe block at height $h+1$. Therefore, $P_{h+1} \geq \alpha^2$ in both of these cases.

If we learn $S(h)$ through bullet (c), then:

- If it is because Attacker found a block at height $h+1$, then if they find the next block, this block of height $h+1$ certainly safe (because Attacker found a block of height $h+1$ before Honest found a block of height $h-1$). So $P_{h+1} \geq \alpha \geq \alpha^2$.
- If it is because Honest creates a block at height $h-1$, then Attacker broadcasts their pivotal block of height h and this acts like a new genesis block. From here, if Attacker creates the next two blocks, they will create a safe block at height $h+1$. So $P_{h+1} \geq \alpha^2$. □

Observe some subtlety in the proof of Lemma 5, and in particular that the choice of moments to compute P_h is significant. For example, if we were to compute the probability that Attacker creates a safe block at height h conditioned on $S(h-2)$ being Single, and $S(h-1)$ being Pair, *and Honest has broadcast a block of height $h-1$*, then the Attacker cannot possibly find a safe block of height h (because their block of height h, if created, is immediately pivotal). So, it is important that we compute P_h at the moment that $S(h-1)$ is determined (which is immediately when Attacker finds its first block at height $h-1$, and it is unknown whether Honest will find the next block or not). This highlights some of the subtlety needed to keep the analysis clean.

Next, we need to define how to implement the proposed labeling with a broadcasting strategy.

Lemma 6 (Implementability). *The following* broadcasting strategy *realizes the labels output by the labeling strategy in Definition 9:*

- *If $S(h)$ is labeled as Single (and you created a block of height h): broadcast the block of height h the moment a block of height $h-1$ is broadcast (and you know that $S(h)$ is labeled Single). Note that this may imply broadcasting a block of height h the moment its created.*
- *If $S(h)$ is labeled as Pair: broadcast the block of height h the moment another block of height h is broadcast (and you know that $S(h)$ is labeled Pair).*[17]

Proof. We first need to confirm that for all h that are eventually labeled as Single, we learn that $S(h)$ is Single early enough to broadcast our block of height h according to the rule above *before Honest finds a block of height h*. If we can do this, then h will indeed be Single.

To see this, observe that if a block of height h is pivotal, we learn this immediately when there are two blocks of height $h-1$, and Attacker creates the first block of height h. This is before Honest has a block of height h, and therefore if a block is pivotal, we label it as Single in time.

Similarly, if a block of height h is safe, we learn this either the moment we learn that $S(h-1)$ is Single (which we certainly know by the moment the block of height h is created, perhaps earlier, by Claim 1), or the moment that we create a block of height $h+1$ before Honest has a block of height $h-1$. In both cases, Honest does not have a block of height h. This confirms that if we decide $S(h)$ is Single, we know this before Honest finds a block of height h, and therefore our broadcasting strategy broadcasts our block of height h in time for h to be Single.

For broadcasting Pairs, we just need to confirm that the timing constraints do not conflict with those demanded for Single (that is, we need to confirm that the strategy does not accidentally claim

[17] In particular, if the public longest chain has height $h-2$, and Attacker has two hidden blocks of heights $h-1$ labeled Pair and h labeled Single, when Honest broadcasts a block of height $h-1$, both bullets trigger simultaneously, and the attacker will broadcast both its hidden blocks at once.

to broadcast a block of height $h+1$ before a block of height h that it points to). Assuming this is the case, then because we wait until Honest broadcasts their block of height h before broadcasting ours, there will certainly be a Pair at height h. To see that there is no conflict, observe that the broadcasting strategy for h labeled Pair only asks to wait to broadcast a block of height h until Honest finds a block of height h, whereas for $h+1$ labeled Single it only asks to broadcast a block of height $h+1$ as soon as a block of height h is broadcast. Therefore, there is no conflict where the broadcasting strategy asks to broadcast a block of height $h+1$ before a block of height h. □

Lemma 7 (Undetectability). *The broadcasting strategy of Lemma 6, with the labeling strategy of Definition 9, for any $\beta \leq \alpha^2$, produces the distribution where a β-fraction of rounds have Pairs, and $1-\beta$ have Singles, independently.*

Proof. Observe that at the moment that $S(1), \ldots, S(h-1)$ are fixed, $S(h)$ is equal to Pair with probability exactly β. This is true not only conditioned on $S(1), \ldots, S(h-1)$, but also conditioned on any additional information known at the instant $S(h-1)$ is fixed. Therefore, for all $S(1), \ldots, S(h-1)$, the probability that $S(h)$ is Pair conditioned on $S(1), \ldots, S(h-1)$ is exactly β. Therefore, the distribution of states is independent each round, and equal to Pair with probability β. □

5.3 Reward Analysis

We provide in Appendix C a description of a Markov Chain that can be used to compute exactly the reward of our strategy. In this section, we'll aim instead to simply find a sufficiently large α so that our strategy is strictly profitable.

Recall that, by Corollary 1, it suffices to focus on Paired rounds and determine their winner. Consider the sequence $(S(h))_{h=1}^{\infty}$, and observe that by Lemma 7, each of its entries is Pair with probability β and Single otherwise. Consider now a sequence of k Pairs between two Single rounds.

Claim 2. *For a sequence of $k > 1$ consecutive Pair rounds, Attacker wins all these rounds.*

Proof. Say that height h is the first Pair. Observe that in order for each subsequent height to possibly be labeled as Pair, Attacker must have a safe block at that height. In order to have a safe block at heights $h, \ldots, h+k-1$, Attacker must also have a block at height $h+k$ before Honest has a block of height $h+k-1$, and height $h+k$ is Single. Therefore, Attacker will certainly win all these Pairs. □

We can therefore focus on *Solo Pairs*: h such that $S(h)$ is Pair, and $S(h-1)$ and $S(h+1)$ are Single.

Claim 3. *Let w denote the fraction of Solo Pairs won by the attacker. We have $w \geq \frac{2\alpha-\alpha^2-\beta}{1-\beta}$.*

Proof. Consider a Single at height $h-1$, and a Pair at height h. If Attacker has any hidden blocks, then Attacker will certainly win the pair at height h. Otherwise, $P_{h+1} = \alpha^2$ (because Attacker's next block is safe only if they create each of the next two). This means that:

- If Honest finds the next two blocks, then h is a Solo Pair, and Honest wins it. This happens with probability $(1-\alpha)^2$.
- If Attacker finds exactly one of the next two blocks, then their block of height $h+1$ is pivotal, so h is a Solo Pair and Attacker wins it. This happens with probability $2\alpha(1-\alpha)$.
- If Attacker finds both of the next blocks, then $h+1$ is safe, but is marked Single anyway with probability $1-\beta/\alpha^2$. Therefore, this is a Solo Pair that Attacker wins with probability $\alpha^2 \cdot (1-\beta/\alpha^2) = \alpha^2 - \beta$.

Altogether, we conclude that Attacker wins at least a $\frac{2\alpha-\alpha^2-\beta}{1-\beta}$ fraction of Solo Pairs. □

Corollary 3. *The fraction of pairs won by Attacker is $> (1-\alpha)$ as long as $\alpha > \frac{3-2\beta-\sqrt{5-4\beta}}{2(1-\beta)}$.*

Proof. Recall that the number of Pair rounds in a row (between two Single rounds) is distributed independently across time, and equal to i with probability $(1-\beta)\cdot\beta^i$. Therefore, we can write the expected number of Pair rounds between two Single rounds as

$$\sum_{i=0}^{\infty}(1-\beta)\cdot\beta^i\cdot i = \frac{\beta}{1-\beta}.$$

Similarly, we can write an expression for the expected number of Pair rounds won by the attacker,

$$w\cdot(1-\beta)\beta + \sum_{i=2}^{\infty}(1-\beta)\cdot\beta^i\cdot i = w\cdot\beta\cdot(1-\beta) + \frac{(2-\beta)\cdot\beta^2}{1-\beta} = \frac{w\cdot\beta\cdot(1-\beta)^2 + (2-\beta)\cdot\beta^2}{1-\beta},$$

where w denotes the fraction of Solo Pairs won.

Now, we can compute the expected fraction of Pairs won by the attacker. By linearity of expectation, the expected number of Pair rounds before we see the n^{th} Single is $n\cdot\frac{\beta}{1-\beta}$, and the expected number of Pair rounds won by the attacker is $n\cdot\frac{w\beta(1-\beta)^2+(2-\beta)\beta^2}{1-\beta}$. By the law of large numbers, we have that with probability 1:

$$\lim_{n\to\infty}(\#\text{ Pairs before } n^{th}\text{ Single})/n = \frac{\beta}{1-\beta},$$

$$\lim_{n\to\infty}(\#\text{ Pairs won by Attacker before } n^{th}\text{ Single})/n = \frac{w\cdot\beta\cdot(1-\beta)^2 + (2-\beta)\cdot\beta^2}{1-\beta}.$$

Therefore, the expected fraction of Pairs won by the attacker is:

$$\frac{\frac{(2-\beta)\cdot\beta^2+w\cdot\beta\cdot(1-\beta)^2}{1-\beta}}{\frac{\beta}{1-\beta}} = \frac{w\cdot\beta\cdot(1-\beta)^2 + (2-\beta)\cdot\beta^2}{1-\beta}.$$

Substituting in our lower bound on w from Claim 3 in the right hand side gives the following lower bound on the expected fraction of Pairs won by the attacker:

$$2\beta-\beta^2+w\cdot(1-\beta)^2 \geq 2\beta-\beta^2+(2\alpha-\alpha^2-\beta)\cdot(1-\beta).$$

And now we need to understand when $2\beta-\beta^2+(2\alpha-\alpha^2-\beta)\cdot(1-\beta)\geq 1-\alpha$. Rearranging, this inequality is equivalent to $\alpha^2\cdot(\beta-1)+\alpha\cdot(3-2\beta)+(\beta-1)\geq 0$. The left-hand-side has roots of:

$$\frac{-3+2\beta\pm\sqrt{(3-2\beta)^2-4(1-\beta)^2}}{2(\beta-1)} = \frac{3-2\beta\pm\sqrt{5-4\beta}}{2(1-\beta)}.$$

So the inequality holds iff

$$\alpha\in\left(\frac{3-2\beta-\sqrt{5-4\beta}}{2(1-\beta)}, \frac{3-2\beta+\sqrt{5-4\beta}}{2(1-\beta)}\right).$$

In particular, observe that the upper bound is at least 1 for all $\beta\in[0,1]$ (the numerator is always at least 2 and the denominator is at most 2). So the desired claim holds as long as $\alpha > \frac{3-2\beta-\sqrt{5-4\beta}}{2(1-\beta)}$. □

Proof of Theorem 4. Observe that $\frac{3-2\beta-\sqrt{5-4\beta}}{2(1-\beta)}$ is monotone decreasing in β.[18] Therefore, when $\alpha > \frac{3-\sqrt{5}}{2}$, we have $\alpha > \frac{3-2\beta-\sqrt{5-4\beta}}{2(1-\beta)}$ for all $\beta \in (0,1)$. Therefore, when $\alpha > \frac{3-\sqrt{5}}{2}$, there is a range of β arbitrarily close to 0 (and therefore a range of ℓ' arbitrarily close to 0) where our strategy is both statistically undetectable and strictly profitable. □

We can use Corollary 3 to derive a bound on how much more profit the attacker can make when allowed a target latency of $\beta > 0$ compared to a latency of zero. In particular, Lemma 4 states the reward as a function of δ (fraction of pairs won) and β (target latency): $\alpha - (1-\alpha-\delta)\cdot\beta$. The proof of Corollary 3 establishes that $\delta \geq 2\beta + \beta^2 + (2\alpha - \alpha^2 - \beta)\cdot(1-\beta)$. Plugging this into Lemma 4 states the reward – the improvement over α when allowed a latency of β – is at least $(2\beta + \beta^2 + (2\alpha - \alpha^2 - \beta)\cdot(1-\beta) - 1 + \alpha)\cdot\beta$.

We also leverage Corollary 1 to derive a bound on α that suffices for a strictly profitable strategy that is ℓ'-statistically undetectable, but not necessarily ℓ' close to 0.

Proposition 4. *Let $\alpha^* \approx 0.3586$ denote the unique real root in $(0,1)$ of $\alpha^4 - 2\alpha^3 + 3\alpha - 1$. Then for any $\alpha > \alpha^*$, there exists an ℓ' such that there is a strictly profitable strategy for the Nakamoto Consensus Game that is ℓ'-statistically undetectable.*

Proof. By Lemma 7, our proposed strategy is statistically undetectable for all $\beta \leq \alpha^2$, so we will take $\beta = \alpha^2$.[19] By Corollary 1, we have designed a strategy that is strictly profitable (and produces the desired view) as long as $\alpha > \frac{3-2\alpha^2-\sqrt{5-4\alpha^2}}{2(1-\alpha^2)}$. Expanding out the calculations, we need:

$$\begin{aligned}
\alpha &> \frac{3-2\alpha^2-\sqrt{5-4\alpha^2}}{2(1-\alpha^2)} \\
\Leftrightarrow 2\alpha(1-\alpha^2) &> 3-2\alpha^2-\sqrt{5-4\alpha^2} \\
\Leftrightarrow 3-2\alpha(1-\alpha^2)-2\alpha^2 &< \sqrt{5-4\alpha^2} \\
\Leftrightarrow 3-2\alpha+2\alpha^3-2\alpha^2 &< \sqrt{5-4\alpha^2}
\end{aligned}$$

Because we are only interested in the range $\alpha \in (0, 1/2)$, the LHS and RHS are always non-negative. Therefore we may continue with:

$$\begin{aligned}
3-2\alpha-2\alpha^2+2\alpha^3 &< \sqrt{5-4\alpha^2} \\
\Leftrightarrow (3-2\alpha-2\alpha^2+2\alpha^3)^2 &< 5-4\alpha^2 \\
\Leftrightarrow 4\alpha^6-8\alpha^5-4\alpha^4+20\alpha^3-8\alpha^2-12\alpha+9-5+4\alpha^2 &< 0 \\
\Leftrightarrow 4\alpha^6-8\alpha^5-4\alpha^4+20\alpha^3-4\alpha^2-12\alpha+4 &< 0 \\
\Leftrightarrow 4(\alpha-1)\cdot(1+\alpha)\cdot(\alpha^4-2\alpha^3+3\alpha-1) &< 0.
\end{aligned}$$

The quartic on the LHS has two real roots. One is negative, and the other is $\approx 35.86\%$. Therefore, the quartic is positive on $[.3586, 1]$, and the entire LHS is negative. □

[18] Its derivative with respect to β is $\frac{2\beta-3+\sqrt{5-4\beta}}{2\cdot\sqrt{5-4\beta}\cdot(1-\beta)^2}$. The denominator is ≥ 0 for all $\beta \in (0,1]$. The numerator is 0 at $\beta = 1$, and has derivative $2 - \frac{2}{\sqrt{4-5\beta}}$, which is positive on $(0,1)$. Therefore, the numerator is negative on the entire interval $(0,1)$.

[19] We have previously observed in the proof of Theorem 4 that the profitability threshold is monotone decreasing in β, so taking the largest viable β will optimize this.

5.4 Looking Forward: Generalizing to Main Result II

Here, we briefly note complexities extending to our second main result (the complete proof of which can be found in the full version). Recall that our second main result considers statistically undetectable Selfish Mining strategies for an ℓ-NCG with $\ell > 0$.

The key difference stems from the following. In our warmup result, Attacker is relatively free to decide which heights will be Single vs. Pair, because they will win every Pair round anyway. In our first main result, Attacker is sometimes "forced" to set an upcoming height to be Single, because they have only a pivotal block left (and having a conflict with this block is risky, and could cause the loss of many other conflicts that Attacker could instead win right now). In our second main result, Attacker is still sometimes "forced" to immediately broadcast pivotal blocks and create Single heights, but they are also sometimes "forced" to label a height as Pair (simply because both miners find a block at the same time). We now briefly elaborate how this impacts our analysis.

Because of these "Forced Pairs", our labeling strategy must change. One particular challenge is that we cannot set P_h conditioned on the entire state of information known to the Attacker at a given time, as doing so prevents any dishonest strategy from also being statistically undetectable.[20] Instead, our labeling strategy instead conditions only on the states, and *not* on the full information available to Attacker. For example, our strategy computes (e.g.) P_5 (and other related probabilities) conditioned on the fact that the first four states are ⟨Single, Single, Pair, Pair⟩, and at the moment these states are set. But, it does *not* condition on further information available to Attacker (such as whether the last two Pairs were Forced, whether Attacker has any hidden blocks, etc.). This forces our analysis to be Bayesian, and further understand the probabilities of a particular "complete world" given the current sequence of states. This is the key complexity associated with confirming statistical undetectability.

The key complexity associated with our reward calculation is the following. For our first main result, the honest strategy gets reward α, and this provides a clean target fraction of pairs to win (of $> (1-\alpha)$) in order to be strictly profitable. In an ℓ-NCG with $\ell > 0$, the honest strategy does not win an α fraction of rewards, so our benchmark is different and calculations become significantly more involved. In order to keep the calculations as tractable as possible, we use an analysis technique introduced in [Sapirshtein et al., 2016] to directly understand when withholding Attacker's first block (and attempting to Selfish Mine) might be more profitable than immediately broadcasting it (and claiming immediate reward). In particular, Proposition 5 in the full version provides a sufficient condition (as a function of α, β') for a strictly profitable and undetectable strategy to exist (and Theorem 5 follows by confirming numerically that this condition holds for $\alpha \geq 38.2\%$ and all $\beta' \in [0, 1]$).

6 CONCLUSION

We provide a statistically undetectable Selfish Mining attack. We choose to study the canonical setting of proof-of-work longest-chain protocols with a block reward, as it is the theoretically most-developed. The key takeaway from our work is that relying on risk of statistical detection to dissuade attackers is not necessarily sound — there may be ways for attackers to still profit and avoid statistical detection. Our paper leaves several directions open for future work and discussion.

On the technical front, our paper considers a particular stylized model of latency, where orphans naturally occur independently every round. It is worthwhile to further explore alternative latency

[20] We refer the reader to the technical appendices in the full version of the paper to see why. Briefly, the issue is that if the attacker finds a single block following a "Forced Pair", that block is immediately pivotal, and therefore should be broadcast. But this then means that following a Forced Pair, we can only get another Pair if we get another Forced Pair, which happens with a fixed probability set by ℓ' that is outside our control.

models, and develop an understanding of which types of "honest-with-latency" worlds can be induced by profitable deviations.

On the modeling front, our paper considers detection methods based only on the shape of the blockchain (e.g. the pattern of orphaned blocks), and not on any time-sensitive information (such as timestamps and timing of messages). On one hand, it is likely *quite* challenging (and perhaps even impossible) to design a profitable Selfish Mining strategy whose timestamps mimic that of natural latency. On the other hand, if timestamps are necessary to detect a deviant strategy, and detection of Selfish Mining via timestamps could significantly impact the value of the underlying cryptocurrency, this is potentially a significant vulnerability (because timestamps are trivial to manipulate). Indeed, the deviation discovered by [Yaish et al., 2022a] involves manipulating timestamps on Ethereum (for the purpose of extracting additional mining revenues, not to affect the underlying value of ETH).

Also on the modeling front, our strategy doesn't risk significantly impacting the underlying cryptocurrency's value, as its impact is consistent with orphans caused by latency. Still, note that while Attacker can safely disguise its behavior as latency, our Attacker still introduces extra latency into the system (and in particular, causes orphaned blocks at a higher rate). It is entirely possible that an increased orphan rate might have a smooth negative impact on the underlying cryptocurrency's value without outright tanking it. This aspect would also be important to explore, although the model might look significantly different than ours.

On the broader web3/blockchain ecosystem front, the conceptual contributions of our paper apply quite generally: deviant behavior can sometimes disguised as natural occurrences. Therefore, our work motivates further study of any domain where deviant strategies exist that are profitable when denoted in the underlying token/cryptocurrency, but are currently believed to be disincentivized due to risk of detection. In general, wherever detectable profitable deviations exist, it is important to understand whether undetectable (or "explainable-by-nature") profitable deviations exist as well.

REFERENCES

Maryam Bahrani, Pranav Garimidi, and Tim Roughgarden. 2023. Transaction Fee Mechanism Design with Active Block Producers. *arXiv preprint arXiv:2307.01686* (2023).

Maryam Bahrani, Pranav Garimidi, and Tim Roughgarden. 2024. Transaction Fee Mechanism Design in a Post-MEV World. *Cryptology ePrint Archive* (2024).

Georgios Birmpas, Elias Koutsoupias, Philip Lazos, and Francisco J. Marmolejo Cossío. 2020. Fairness and Efficiency in DAG-Based Cryptocurrencies. In *Financial Cryptography and Data Security - 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers (Lecture Notes in Computer Science, Vol. 12059)*, Joseph Bonneau and Nadia Heninger (Eds.). Springer, 79–96. https://doi.org/10.1007/978-3-030-51280-4_6

Jonah Brown-Cohen, Arvind Narayanan, Alexandros Psomas, and S. Matthew Weinberg. 2019. Formal Barriers to Longest-Chain Proof-of-Stake Protocols. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*. 459–473. https://doi.org/10.1145/3328526.3329567

Miles Carlsten, Harry A. Kalodner, S. Matthew Weinberg, and Arvind Narayanan. 2016. 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*. 154–167. https://doi.org/10.1145/2976749.2978408

Vanessa Chicarino, Célio Albuquerque, Emanuel Jesus, and Antônio Rocha. 2020. On the detection of selfish mining and stalker attacks in blockchain networks. *Annals of Telecommunications* 75 (2020), 143–152.

Hao Chung and Elaine Shi. 2023. Foundations of Transaction Fee Mechanism Design. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, Nikhil Bansal and Viswanath Nagarajan (Eds.). SIAM, 3856–3899. https://doi.org/10.1137/1.9781611977554.ch150

Ittay Eyal and Emin Gün Sirer. 2014. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*. Springer, 436–454.

Matheus V. X. Ferreira, Ye Lin Sally Hahn, S. Matthew Weinberg, and Catherine Yu. 2022. Optimal Strategic Mining Against Cryptographic Self-Selection in Proof-of-Stake. In *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, David M. Pennock, Ilya Segal, and Sven Seuken (Eds.). ACM, 89–114. https://doi.org/10.1145/3490486.3538337

Matheus V. X. Ferreira and S. Matthew Weinberg. 2021. Proof-of-Stake Mining Games with Perfect Randomness. In *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, Péter Biró, Shuchi Chawla, and Federico Echenique (Eds.). ACM, 433–453. https://doi.org/10.1145/3465456.3467636

Amos Fiat, Anna Karlin, Elias Koutsoupias, and Christos H. Papadimitriou. 2019. Energy Equilibria in Proof-of-Work Mining. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, Anna Karlin, Nicole Immorlica, and Ramesh Johari (Eds.). ACM, 489–502. https://doi.org/10.1145/3328526.3329630

Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The Bitcoin Backbone Protocol: Analysis and Applications. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 9057)*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer, 281–310. https://doi.org/10.1007/978-3-662-46803-6_10

Guy Goren and Alexander Spiegelman. 2019. Mind the Mining. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, Anna Karlin, Nicole Immorlica, and Ramesh Johari (Eds.). ACM, 475–487. https://doi.org/10.1145/3328526.3329566

Gur Huberman, Jacob Leshno, and Ciamac C. Moallemi. 2021. Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System. *Review of Economic Studies* 88 (2021), 3011–3040. Issue 6. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3025604

Harry Kalodner, Steven Goldfeder, Xiaoqi Chen, S. Matthew Weinberg, and Edward W. Felten. 2018. Arbitrum: Scalable Smart Contracts. In *Proceedings of the 27th USENIX Security Symposium (USENIX)*.

Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. 2016. Blockchain Mining Games. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*. 365–382. https://doi.org/10.1145/2940716.2940773

Ron Lavi, Or Sattath, and Aviv Zohar. 2019. Redesigning Bitcoin's fee market. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. 2950–2956. https://doi.org/10.1145/3308558.3313454

Sheng-Nan Li, Carlo Campajola, and Claudio J. Tessone. 2022. Twisted by the Pools: Detection of Selfish Anomalies in Proof-of-Work Mining. *CoRR* abs/2208.05748 (2022). https://doi.org/10.48550/arXiv.2208.05748 arXiv:2208.05748

Sheng-Nan Li, Zhao Yang, and Claudio J. Tessone. 2020a. Mining blocks in a row: A statistical study of fairness in Bitcoin mining. In *IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020, Toronto, ON, Canada, May 2-6, 2020*. IEEE, 1–4. https://doi.org/10.1109/ICBC48266.2020.9169436

Sheng-Nan Li, Zhao Yang, and Claudio J. Tessone. 2020b. Proof-of-Work cryptocurrency mining: a statistical approach to fairness. In *IEEE/CIC International Conference on Communications in China (ICCC Workshops), ICBC 2020, Toronto, ON, Canada, May 2-6, 2020*. IEEE, 156–161.

Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system.

Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. 2016. *Bitcoin and cryptocurrency technologies: a comprehensive introduction.* Princeton University Press.

Michael Neuder, Daniel J. Moroz, Rithvik Rao, and David C. Parkes. 2019. Selfish Behavior in the Tezos Proof-of-Stake Protocol. *CoRR* abs/1912.02954 (2019). arXiv:1912.02954 http://arxiv.org/abs/1912.02954

Michael Neuder, Daniel J. Moroz, Rithvik Rao, and David C. Parkes. 2020. Defending Against Malicious Reorgs in Tezos Proof-of-Stake. In *AFT '20: 2nd ACM Conference on Advances in Financial Technologies, New York, NY, USA, October 21-23, 2020*. ACM, 46–58. https://doi.org/10.1145/3419614.3423265

Tim Roughgarden. 2021. Transaction Fee Mechanism Design. In *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, Péter Biró, Shuchi Chawla, and Federico Echenique (Eds.). ACM, 792. https://doi.org/10.1145/3465456.3467591

Muhammad Saad, Laurent Njilla, Charles Kamhoua, and Aziz Mohaisen. 2019. Countering selfish mining in blockchains. In *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 360–364.

Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. 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*. 515–532. https://doi.org/10.1007/978-3-662-54970-4_30

Elaine Shi, Hao Chung, and Ke Wu. 2023. What Can Cryptography Do for Decentralized Mechanism Design?. In *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA (LIPIcs, Vol. 251)*, Yael Tauman Kalai (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 97:1–97:22. https://doi.org/10.4230/LIPIcs.ITCS.2023.97

Zhaojie Wang, Qingzhe Lv, Zhaobo Lu, Yilei Wang, and Shengjie Yue. 2021. ForkDec: accurate detection for selfish mining attacks. *Security and Communication Networks* 2021 (2021), 1–8.

Aviv Yaish, Gilad Stern, and Aviv Zohar. 2022a. Uncle Maker: (Time)Stamping Out The Competition in Ethereum. Cryptology ePrint Archive, Paper 2022/1020. https://eprint.iacr.org/2022/1020 https://eprint.iacr.org/2022/1020.

Aviv Yaish, Saar Tochner, and Aviv Zohar. 2022b. Blockchain Stretching & Squeezing: Manipulating Time for Your Best Interest. In *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, David M.

Pennock, Ilya Segal, and Sven Seuken (Eds.). ACM, 65–88. https://doi.org/10.1145/3490486.3538250
Andrew Chi-Chih Yao. 2018. An Incentive Analysis of some Bitcoin Fee Designs. *CoRR* abs/1811.02351 (2018). arXiv:1811.02351 http://arxiv.org/abs/1811.02351

A SELFISH MINING IN PRACTICE

A.1 Relevance of Selfish Mining

Strategic manipulations, like Selfish Mining, which do not directly affect consensus do not pose the same immediate threat as a double-spend attack. However, they are concerning in practice for (at least) the following two reasons.

Centralizing Force. Strategic manipulations that provide mild supralinear rewards are a centralizing force: a cohort of miners can achieve greater rewards together than separately. Or, put another way, a sufficiently large miner will achieve greater per-investment rewards than a smaller miner simply due to their size. In general, centralizing forces are undesirable in blockchain applications, as they can lead to centralization over time (which may enable more devastating attacks like double-spends).

Synergy with consensus-breaking attacks. Canonical strategic manipulations happen to have synergy with canonical attacks. Consider for example Selfish Mining and double-spending. A Selfish Miner sometimes finds themselves with a hidden chain of several blocks, and launching a double-spend attack at this time is more likely to succeed. Consider also seed manipulation and committee takeover in BFT-based protocols. Manipulating the pseudorandom seed not only helps the deviator lead more rounds, but also helps them control a greater fraction of the BFT committee. If strategic manipulations are profitable, they may additionally lower the cost of stronger consensus-breaking attacks.

Summary. The above two paragraphs give a brief summary of why strategic manipulations that mildly increase profit of deviators are relevant in practice. Although these manipulations don't directly undermine consensus, they increase the risk the risk of consensus-breaking attacks in the future (via centralizing forces) and in the present (via increased success probability).

A.2 Factors impacting Selfish Mining

It is worth distinguishing deviations from consensus protocols in the the following manner: does the deviator's profit come at the cost of other participants in consensus, or at the cost of a user of the protocol? For example, Selfish Mining steals rewards from other miners, whereas double-spending steals cryptocurrency from a user. Numerous attacks at the cost of users have been detected across multiple blockchains. To the best of our knowledge, [Yaish et al., 2022a] recently discovered the first instance of a strategic manipulation of a consensus protocol whose profits were primarily at the expense of other consensus participants. In particular, to the best of our knowledge, no evidence of Selfish Mining in practice has been discovered.[21]

Note that this holds on Bitcoin, despite periods of time (including at the time of writing) when a single mining pool controls $> 1/3$ of Bitcoin's total hashrate. This also holds on much smaller "altcoins" that are vulnerable to the same attack, despite the low cost of a single entity acquiring $> 1/3$ of the total hashrate. The following is a summary of several potential explanations for this put forth by the community:

[21]It is not entirely clear how much effort has been spent aiming to detect Selfish Mining in practice, but we are not aware of any successful detections.

(1) It is possible that no single entity has had sufficient hashpower/network connectivity for Selfish Mining to be profitable on large cryptocurrencies like Bitcoin.[22]
It is worth observing some limitations of this argument, beyond the present state of large cryptocurrencies. For example, this argument does not apply well to altcoins, as it is significantly less expensive to control a $> 1/3$ fraction of the hashrate. Moreover, it explains why Selfish Mining *has perhaps not yet occurred on Bitcoin*, but does not explain why it *forever will not occur on Bitcoin*. Indeed, recall that with sufficiently high network connectivity, the hashrate required for Selfish Mining to be profitable can drop all the way to 0 (so confirming that Selfish Mining is not a threat would require constant estimations of hashrate and network connectivity of large miners, which may be problematic in a pseudonymous protocol).

(2) There are significant engineering challenges to running a mining operation. Moreover, implementing Selfish Mining requires interacting with ASICs, and may not be as simple as its pseudocode. Therefore, the small boost to profit may not outweigh the opportunity cost of tackling more significant engineering challenges.[23]
It is also worth observing some limitations of this argument, beyond the present state of cryptocurrency mining. Indeed, the cryptocurrency mining industry is still rapidly evolving, and the opportunity cost of ignoring other engineering challenges is high. But as the industry stabilizes, miners might have fewer options to improve profit margins.

(3) If Selfish Mining were traceable to the deviator, the deviator might suffer targeted repercussions. In particular, note that Selfish Mining by a public mining pool would be traceable to the operator, because the mining pool has to ask its participants to Selfish Mine on its behalf[24] Therefore, miners may leave the pool, eliminating its ability to Selfish Mine in the first place. Similarly, the community (or perhaps a legal enforcement agency) could decide to censor all coins owned by the deviator. Note that targeted repercussions are certainly not guaranteed,[25] but the possibility may be sufficient to deter a risk-averse miner.
The limitation of this argument is simply that Selfish Mining is untraceable in absence of outside information (such as instructions from a public pool). So, while this argument applies well to public mining pools, it does not apply well to large individual miners (who can make a new public key for every block they mine).

(4) Selfish Mining is statistically detectable, and the value of the underlying cryptocurrency may be (significantly) negatively affected as a result. Therefore, while Selfish Mining may be profitable when denominated in the underlying cryptocurrency, it could be wildly unprofitable when denominated in an objective unit of value (e.g. USD).
This argument applies broadly to any miner considering any statistically detectable deviation. However, it still has limitations. For example, it relies on the miner's belief that detection will negatively impact the underlying cryptocurrency's value and inability to hedge against

[22]Note that a public mining pool is not a "single entity" — see later bullet.

[23]The authors learned of this possibility from Arvind Narayanan and Aviv Zohar, and thank them for suggesting it.

[24]Observe that in order to successfully Selfish Mine, the Selfish Miner *must not immediately broadcast blocks after creation.* Therefore, if a public mining pool wishes to Selfish Mine, they must request that members of the pool not broadcast their blocks immediately upon creation. Moreover, in order to successfully Selfish Mine, the Selfish Miner must sometimes *point to a block that has not yet been broadcast* when creating a new block. Therefore, if a public mining pool wishes to Selfish Mine, they must request that members of the pool not broadcast the existence of the block to which their own created blocks will point. Due to both of these, any members of a public mining pool will detect that the mining pool is attempting to Selfish Mine (and because the mining pool is public, it will be hard to keep this hidden).

[25]Indeed, we've previously noted that [Yaish et al., 2022a] recently identify a deviant mining pool on Ethereum. F2Pool quickly admitted to deviation, but it remains unclear if F2Pool has suffered as a result.

this,[26] or the miner's risk-aversion. After our work, another limitation is that perhaps the deviation is statistically undetectable (in which case the market cannot possibly react to its detection).

In summary, the above bullets provide answers to the following questions:
Despite theoretical profitability in a stylized model, **why might it be the case that no individual entity has Selfish Mined on Bitcoin?** Bullet One argues the possibility that no single entity ever possessed the hashrate/network connectivity to profit. Bullet Two argues that even if they did, the modest gains may not justify the engineering opportunity cost. Bullet Four further argues that even if the math works out when denominated in bitcoin, there is significant uncertainty and high risk of significant losses when denominated in USD.
Despite theoretical profitability in a stylized model, **why might it be the case that no individual entity has Selfish Mined on smaller altcoins?** Bullet Two argues that, even with sufficient resources to profitably Selfish Mine, the modest gains may not justify the engineering opportunity cost. Bullet Four further argues that even if the math works out when denominated in altcoin, there is significant uncertainty and high risk of significant losses when denominated in USD.
Despite theoretical profitability in a stylized model, **why might it be the case that no public mining pool has Selfish Mined on any cryptocurrency?** Bullet Two argues that, even with sufficient resources to profitably Selfish Mine, the modest gains may not justify the engineering opportunity cost. Bullet Three argues that there is significant uncertainty and high risk of significant retaliation (denominated in either the underlying cryptocurrency, or USD). Bullet Four further argues that even if the math works out when denominated in bitcoin, there is significant uncertainty and high risk of significant losses when denominated in USD.

They key question of interest to mechanism design researchers with an eye towards fundamental concepts not overfit to the current state of affairs is the following: given that Selfish Mining is theoretically profitable in a stylized model but has not been detected in practice, **is there a realistic concern of Selfish Mining in the future?** Prior to our work, Bullet Four is (in the authors' opinion, and anecdotally from community discussions) perhaps the strongest argument in the negative direction. Our work pushes this answer closer to the affirmative, and serves as practical motivation for the design and analysis of updated consensus protocols with stronger incentive guarantees.

Finally, note that while we have phrased this key question in the language of Selfish Mining in longest-chain proof-of-work cryptocurrencies, and our theorems hold only in this setting, the same principles apply broadly across the blockchain and web3 ecosystem. Indeed many other consensus protocols admit profitable deviations [Brown-Cohen et al., 2019, Carlsten et al., 2016, Ferreira et al., 2022, Ferreira and Weinberg, 2021, Kalodner et al., 2018, Kiayias et al., 2016, Sapirshtein et al., 2016, Yaish et al., 2022a,b], and many other aspects of the broader ecosystem currently leave (small numbers of) individual entities with the power to take significant harmful actions. However, it is commonly argued that participants in consensus protocols and powerful individuals elsewhere in the ecosystem are disincentivized from deviant behavior by identical reasoning to Bullet Four: it may be profitable when denominated in the underlying cryptocurrency/token/etc., but carry an uncertain risk of massive unprofitability when denominated in USD. Our work suggests that further care should be taken in such settings to understand whether such deviations are indeed detectable.

[26] It would be interesting for future work to estimate whether the value of ETH was impacted by [Yaish et al., 2022a]'s discovery and F2Pool's admission, although currently this remains unclear.

B BRIEF COMPARISON TO RICHER LATENCY MODELS

Here, we first briefly note that all prior work we are aware of concerning strategic block withholding consider stylized latency models (essentially, our 0-NCG) [Brown-Cohen et al., 2019, Carlsten et al., 2016, Eyal and Sirer, 2014, Ferreira et al., 2022, Ferreira and Weinberg, 2021, Kiayias et al., 2016, Neuder et al., 2019, 2020, Sapirshtein et al., 2016]. Still, it is worth a brief effort to connect this common stylized model to richer models from distributed computing, such as [Birmpas et al., 2020, Garay et al., 2015].

In comparison to [Garay et al., 2015], which is an adversarial latency model, our Attacker could certainly be implemented by their adversary, and therefore all of our results would hold in their model. But, our Attacker is using nowhere near their adversary's capability (in particular, only the "rushing" aspect of their adversary is relevant to our Attacker). For example, to "win all ties" in our model, after finding a block their adversary could wait until an honest party finds a block, and then reorder all Receive() strings to place their own block first. The adversary does not need to delay messages outside of this reordering, nor corrupt the content of any messages. To "lose all ties" in our model, their adversary does not need to take any malicious actions, and just needs to be aware of all parties' Receive() strings to time their own broadcasts.

In comparison to [Birmpas et al., 2020], which is a stochastic latency model, our model is equivalent to their special case with $q_i = 1$ for all i. The two key differences to the most general version of the their model are: (a) their model is more complex in that it allows for different miners to experience different latency, and (b) in our model, every player is always within one block of being up to date, whereas players in the [Birmpas et al., 2020] model are sometimes multiple blocks behind.

C EXACT ANALYSIS OF REWARDS USING MARKOV CHAINS

To do the award analysis using a Markov Chain, we'll simplify analysis by counting *only the Paired rounds, and who wins*. See Lemma 4 for why this suffices.

To analyze the strategy, we'll need a state to keep track of the following information:

- How many blocks does the attacker have hidden that we *know* will form a Pair?
- Is the last hidden block of the attacker Pair, Single, or Undecided? (P, S, U). If there are no hidden blocks, we leave this blank.

We will transition *every time the status of a new height is determined*. Note that several blocks may be created during a single transition. We do this for the following reason. Recall that our undetectability analysis computed a probability P_h at the moment that $S(h-1)$ was determined. It is therefore simplest if we only have transitions from the moment $S(h-1)$ is determined to the moment $S(h)$ is determined (and nothing in between). This will let us a) compute P_h the moment $S(h-1)$ is determined, and b) use P_h explicitly in the transition to where $S(h)$ is determined.

Table 1 summarises all transitions the Markov chain might take. The column q_e refers to the probability of taking this transition, which fully defines the Markov chain. We introduce three counting schemes, which will let us confirm that calculations are done correct. H_e^p and S_e^p count the number of pairs we learn are guaranteed to be won by the honest party and attacker, respectively, during this transition. H_e^b and S_e^b count the number of blocks in the longest chain (part of a pair or not) that we learn are guaranteed to be won by the honest party and attacker, respectively, during this transition. H_e^s and S_e^s count the number of solo pairs we learn are guaranteed to be won by the honest party and attacker, respectively, during this transition.

Lemma 8. *Table 2 gives the correct expressions for computing P_h in SP-Simple game.*

Proof. Recall that in the SP-Simple model, P_h is defined as the probability that Attacker creates a block of height h and that block is safe, conditioned on all information available *as of the first*

Table 1. The Markov Chain describing the attacker's strategy in the SP-Simple model, along with brief explanations of the events corresponding to each transition. The transitions out of State 0 are ommitted, as they are identical to transitions out of state iS with $i = 0$. The respective values of P_h are given in Table 2.

Transition	q_e	H_e^p	S_e^p	H_e^b	S_e^b	H_e^s	S_e^s
(iS,0)	$(1-\alpha)^{i+1}$	0	0	0	1	0	0
Honest completes all pairs plus an extra block, then Attacker finds a block. Attacker publishes all hidden blocks.							
(iS,$(i-j+1)$P)	$(1-\alpha)^j \alpha \frac{\beta}{P_h}$	0	0	0	0	0	0
Honest completes j pairs, then Attacker finds a block. The Attacker immediately decides the new block is Pair.							
(iS,$(i-j)$S)	$(1-\alpha)^j \alpha (1-\frac{\beta}{P_h})$	0	0	0	1	0	0
Honest completes j pairs, then Attacker finds a block. The Attacker immediately decides the new block is Single.							
(iU, 0)	$(1-\alpha)^i$	0	0	0	0	0	0
Honest finds the next i blocks. The last block is pivotal, and Attacker publishes it as Single.							
(iU,$(i+1-j)$U)	$(1-\alpha)^j \alpha \frac{\beta}{P_h}$	0	1	0	1	0	0
Honest finds j blocks and then Attacker finds a block. Attacker decides (in advance) that the undecided block is Pair. The new Attacker block might be pivotal.							
(iU,$(i+1-j)$P)	$(1-\alpha)^j \alpha (1-\frac{\beta}{P_h})\beta$	0	0	0	0	0	0
Honest finds j blocks and then Attacker finds a block. Attacker decides that the undecided block is Single and the new block is Pair.							
(iU,$(i+1-j)$S)	$(1-\alpha)^j \alpha (1-\frac{\beta}{P_h})(1-\beta)$	0	0	0	1	0	0
Honest finds j blocks and then Attacker finds a block. Attacker decides that the undecided block is Single and the new block is Single.							
(iP,0)	$(1-\alpha)^{i+1}$	$1-\gamma$	γ	$2-\gamma$	γ	$1-\gamma$	γ
Honest completes all pairs, triggers a race, which Honest wins. Tiebreaks for Attacker w.p. γ							
(iP,0)	$(i+1)(1-\alpha)^i \alpha$	0	1	0	2	0	1
Honest completes all pairs, triggers a race, which Attack wins.							
(iP, $(i+1-j)$U)	$(j+1)\alpha^2(1-\alpha)^j \frac{\beta}{P_h}$	0	2	0	3	0	0
Honest finds j blocks before Attacker finds 2 blocks a, b, and the first Attacker block a is marked as Pair. Guarantees win of last hidden pair and a for Attacker. b will be in longest chain.							
(iP, $(i+1-j)$P)	$(j+1)\alpha^2(1-\alpha)^j(1-\frac{\beta}{P_h})\beta$	0	1	0	2	0	1
Honest finds j blocks before Attacker finds 2 blocks a, b, with a is marked as Single and b as Pair. Guarantees win of last hidden pair by Attacker, but the fate of b is unknown.							
(iP, $(i-j)$S)	$(j+1)\alpha^2(1-\alpha)^j(1-\frac{\beta}{P_h})(1-\beta)$	0	1	0	3	0	1
Honest finds j blocks before Attacker finds 2 blocks a, b and marks both as Single. Guarantees win of last hidden pair by Attacker.							

Table 2. The values of P_h in the attacker's strategy in the SP-Simple Model, described in Table 1. Justification for these expressions is given in Lemma 8

Transition	P_h
$(i\text{S},\cdot)$	$1-(1-\alpha)^{i+1}$
$(i\text{U},\cdot)$	$1-(1-\alpha)^{i}$
$(i\text{P},\cdot)$	$1-(1-\alpha)^{i+1}-(i+1)\alpha(1-\alpha)^{i}$

moment we know $S(h')$ for all $h' < h$. Recall also that transitions in the Markov chain happen from when $S(h-1)$ is determined to when $S(h)$ is determined. When the originating state in a transition is iS or iP, $h-1$ is the height of the last hidden block, since its state is known to be S or P, and h is the new height whose state is determined via the transition. When the originating state in a transition is iU, height h refers to *the undecided block* whose state is determined via the transition.

(iS,·) or $(0,\cdot)$. Height $h-1$ is Single, so if Attacker finds the first block of height h, it will be safe. Attacker will mine the first block of height h unless Honest completes all i Pairs, and then finds an extra block, all before Attacker finds a block. Therefore, $P_h = 1-(1-\alpha)^{i+1}$.

(iU,·). The Undecided block at height h will be safe unless Honest completes all i Pairs before Attacker finds a block, making the undecided block pivotal. Therefore, $P_h = 1-(1-\alpha)^{i}$.

(iP,·). The event measured by P_h holds *unless* one of the following (mutually exclusive) events occures:

- The block at height h is mined by Honest. This happens if Honest completes the i Pairs and finds another block before Attacker finds any blocks, which happens with probability $(1-\alpha)^{i+1}$.
- The block at height h is mined by Attacker but is Pivotal; that is, Honest finds a block of height $h-1$ before attacker finds a block of height $h+1$. This happens iff among the next $i+1$ blocks to be discovered, Honest finds i and Attacker finds one. The probability of this is $(i+1)\alpha(1-\alpha)^{i}$.

Combining the two implies $P_h 1-(1-\alpha)^{i+1}-(i+1)\alpha(1-\alpha)^{i}$. □

Theorem 6 (Markov Chain Specification). *The following system of equations characterizes the stationary distribution* $\mathbf{p}$ *of the Markov chain in Table 1:*

$$p_0 = \sum_{j=0}^{\infty}(1-\alpha)^{j+1}\cdot p_{j,S} + \sum_{j=1}^{\infty}\left((1-\alpha)^{j+1} + (j+1)\alpha(1-\alpha)^j\right)\cdot p_{j,P} + \sum_{j=2}^{\infty}(1-\alpha)^j\cdot p_{j,U}$$

$$\begin{aligned} p_{i,S} = {} & \sum_{j=0}^{\infty}(1-\alpha)^j\alpha\left(1-\frac{\beta}{1-(1-\alpha)^{i+j+1}}\right)\cdot p_{i+j,S} \\ & + (j+1)\alpha^2(1-\alpha)^j\left(1-\frac{\beta}{1-(1-\alpha)^{i+j+1}-(i+j+1)\alpha(1-\alpha)^{i+j}}\right)(1-\beta)\cdot p_{i+j,P} \\ & + \alpha(1-\alpha)^j\left(1-\frac{\beta}{1-(1-\alpha)^{i+j}}\right)(1-\beta)\cdot p_{i+j,U} \qquad \text{(for } i\geq 1\text{)} \end{aligned}$$

$$\begin{aligned} p_{i,U} = {} & \sum_{j=0}^{\infty}(j+1)\alpha^2(1-\alpha)^j\frac{\beta}{1-(1-\alpha)^{i+j+1}-(i+j+1)\alpha(1-\alpha)^{i+j+1}}(1-\beta)\cdot p_{i+j+1,P} \\ & + (1-\alpha)^j\alpha\cdot\frac{\beta}{1-(1-\alpha)^{i+j+1}}\cdot p_{i+j+1,U} \qquad \text{(for } i\geq 2\text{)} \end{aligned}$$

$$\begin{aligned} p_{i,P} = {} & \sum_{j=0}^{\infty}(1-\alpha)^j\alpha\cdot\frac{\beta}{1-(1-\alpha)^{i+j-1}}\cdot p_{i+j-1,S} \\ & + (j+1)\alpha^2(1-\alpha)^j\left(1-\frac{\beta}{1-(1-\alpha)^{i+j+1}-(i+j+1)\alpha(1-\alpha)^{i+j+1}}\right)\beta\cdot p_{i+j+1,P} \\ & + (1-\alpha)^j\alpha\left(1-\frac{\beta}{1-(1-\alpha)^{i+j+1}}\right)\beta\cdot p_{i+j+1,U} \qquad \text{(for } i\geq 1\text{)} \end{aligned}$$

Theorem 7 (Reward Specification). *Let* $\{p_u\}_u$ *denote the stationary distribution of the Markov Chain described in Table 1. For any transition* e *from state* u *to state* v*, let* $p_e := p_u\cdot q_e$ *denote the fraction of transitions spent taking transition* e*. Then the following three equations each compute the expected reward of Undetectable Selfish Mining:*

$$\frac{\sum_e p_e\cdot S_e^b}{\sum_e p_e\cdot(H_e^b+S_e^b)},$$

$$\frac{\alpha-\beta+\beta\cdot\frac{\sum_e p_e\cdot S_e^p}{\sum_e p_e\cdot(H_e^p+S_e^p)}}{1-\beta}.$$

$$\frac{\alpha-\beta+\beta\cdot\left(\frac{\sum_e p_e\cdot S_e^s}{\sum_e p_e\cdot(H_e^b+S_e^s)}+\left(2-\frac{\sum_e p_e\cdot S_e^s}{\sum_e p_e\cdot(H_e^b+S_e^s)}\right)\beta-\left(1-\frac{\sum_e p_e\cdot S_e^s}{\sum_e p_e\cdot(H_e^b+S_e^s)}\right)\beta^2\right)}{1-\beta}$$

Proof. The first equation follows by the exact same logic as classical Selfish Mining analysis. The second follows by Lemma 4.

The final equation follows by the following calculations, building off Lemma 4. Observe that Undetectable Selfish Mining has the following properties:

- If there are ever multiple Pair rounds in a row, USM wins them all.
- If there is a solo Pair round, USM may win or lose it.
- The number of Pair rounds in a row (between two Single rounds) is distributed independently across time, and equal to i with probability $(1-\beta)\cdot\beta^i$.

Therefore, we can write the expected number of Pair rounds between two Single rounds as:

$$\sum_{i=0}^{\infty}(1-\beta)\beta^i \cdot i = \frac{\beta}{1-\beta}.$$

And the expected number of Pair rounds won by the attacker between two Single rounds as:

$$w \cdot (1-\beta)\beta + \sum_{i=2}^{\infty}(1-\beta)\beta^i \cdot i = \frac{(2-\beta)\beta^2}{1-\beta} + w\beta(1-\beta) = \frac{2\beta^2 - \beta^3 + w\beta - 2w\beta^2 + w\beta^3}{1-\beta}.$$

Therefore, the expected fraction of Pairs won by the attacker is:

$$\frac{\frac{2\beta^2-\beta^3+w\beta-2w\beta^2+w\beta^3}{1-\beta}}{\frac{\beta}{1-\beta}} = w + (2-w)\beta - (1-w)\beta^2. \quad \square$$

D OMITTED PROOFS

Proof of Proposition 1. We will show how to couple an ℓ-NCG with hashrates $\vec{\alpha}$ with one with hashrates $\langle \alpha_1, \frac{1-\prod_{i=2}^{n}(1-\alpha_i\cdot\ell))}{\ell}\rangle$. Observe that in the first case, the probability that only Miner 1 produces a block is $\alpha' := \frac{\alpha_1\cdot\ell\cdot\prod_{i=2}^{n}(1-\alpha_i\cdot\ell)}{1-\prod_{i=1}^{n}(1-\alpha_i\cdot\ell)}$, the probability that both Miner 1 and a Miner > 1 produces a block is $\beta' := \frac{\alpha_1\cdot\ell\cdot(1-\prod_{i=2}^{n}(1-\alpha_i\cdot\ell))}{1-\prod_{i=1}^{n}(1-\alpha_i\cdot\ell)}$, and the probability that only Miners > 1 produce a block is $\frac{(1-a_1\cdot\ell)\cdot(1-\prod_{i=2}^{n}(1-a_i\cdot\ell))}{1-\prod_{i=1}^{n}(1-\alpha_i\cdot\ell)} = \frac{1-\prod_{i=1}^{n}(1-a_i\cdot\ell)}{1-a_i\cdot\ell-\prod_{i=1}^{n}(1-\alpha_i\cdot\ell)} = 1 - \frac{a_i\cdot\ell}{1-\prod_{i=1}^{n}(1-a_i\cdot\ell)} = 1-\alpha'-\beta'$.

Consider now ℓ-NCG with hashrates $\langle \alpha_1, \frac{1-\prod_{i=2}^{n}(1-\alpha_i\cdot\ell))}{\ell}\rangle$. Then again we have that the probability that only Miner 1 produces a block is $\alpha' := \frac{\alpha_1\cdot\ell\cdot\prod_{i=2}^{n}(1-\alpha_i\cdot\ell)}{1-\prod_{i=1}^{n}(1-\alpha_i\cdot\ell)}$, the probability that both Miner 1 and a Miner > 1 produces a block is $\beta' := \frac{\alpha_1\cdot\ell\cdot(1-\prod_{i=2}^{n}(1-\alpha_i\cdot\ell))}{1-\prod_{i=1}^{n}(1-\alpha_i\cdot\ell)}$, and the probability that only Miners > 1 produce a block is $\frac{(1-a_1\cdot\ell)\cdot(1-\prod_{i=2}^{n}(1-a_i\cdot\ell))}{1-\prod_{i=1}^{n}(1-\alpha_i\cdot\ell)} = \frac{1-\prod_{i=1}^{n}(1-a_i\cdot\ell)}{1-a_i\cdot\ell-\prod_{i=1}^{n}(1-\alpha_i\cdot\ell)} = 1 - \frac{a_i\cdot\ell}{1-\prod_{i=1}^{n}(1-a_i\cdot\ell)} = 1-\alpha'-\beta'$.

Therefore, the games can be coupled so that in every round whether just Miner 1, Miner 1 and other miners, or just other miners produce a block is identical. Because s is SP-Simple, it will take the same action across both coupled games. Because s' is a longest-chain protocol that tiebreaks lexicographically or reverse lexicographically, it will also take the same action in both games (specifically, it will tiebreak for or against 1 the same way in both games). This immediately establishes that the reward of Miner 1 is the same in both games.

To see that statistical undetectability translates between the two games, observe that statistical undetectability exactly states that ℓ-NCG with strategies $s, \vec{s}_{-i}$ can be coupled with ℓ'-NCG with a longest-chain strategy and $\vec{s}_{-i}$ and hashrates $\vec{\alpha}$. By the work above, this latter game can be coupled with ℓ'-NCG with a longest-chain strategy and s' and hashrates $\langle \alpha_1, \frac{1-\prod_{i=2}^{n}(1-\alpha_i\cdot\ell))}{\ell}\rangle$ (because all longest-chain strategies are SP-Simple). Therefore, if either of the two undetectability claims hold, all four games can be coupled as desired (implying that the other undetectability claim holds as well). $\square$