SoK: Time to be Selfless?! Demystifying the Landscape of Selfish Mining Strategies and Models

Colin Finkbeiner
University of Connecticut
colin.finkbeiner@uconn.edu

Mohamed E. Najd
University of Connecticut
menajd@uconn.edu

Julia Guskind
Boston University
guskinju@bu.edu

Ghada Almashaqbeh
University of Connecticut
ghada@uconn.edu

***Abstract*—Selfish mining attacks present a serious threat to Bitcoin security, enabling a miner with less than 51% of the network hashrate to gain higher rewards than their fair share. A growing body of works has studied the impact of such attacks and presented numerous strategies under a variety of model settings. This led to a complex landscape making it hard to comprehend the state of the art and distill insights, gaps, and trade-offs.**

In this paper, we demystify the landscape of selfish mining in Bitcoin by systematizing existing studies and evaluating more realistic model adaptations. To the best of our knowledge, our work is the first of its kind. We develop a multi-dimensional systematization framework assessing prior works based on their strategy formulation and targeted models. We go on to distill a number of insights and gaps highlighting open questions and understudied areas. Among them, we find that most of the surveyed works target the block-reward setting and do not account for transaction fees, and generally consider only single attackers. To bridge this gap, we evaluate several existing strategies in the transaction-fee regime—so miner's incentives come solely from transaction fees—for both single and multi-attacker scenarios. We also extend their models to include honest-but-rational miners showing how such adaptations could garner more performant strategy variations. Finally, we touch upon defenses proposed in the literature, and discuss connections between selfish mining and relevant incentivized/fee-driven paradigms.

***Index Terms*—Bitcoin, selfish mining, transaction-fee regime.**

I. Introduction

It was a longstanding belief that Bitcoin's protocol is incentive compatible, and so the network is secure so long as the majority of the mining power is honest. However, the seminal work of Eyal and Sirer [1] invalidated this belief; they showed that selfish mining allows an attacker who controls 33% of the network hashrate to profit more than expected.

Selfish mining is a temporary block-withholding attack that exploits the longest-chain rule of Nakamoto consensus and its fork selection process. By withholding and selectively publishing locally-mined blocks to the public chain, a selfish miner can devoid the work of other miners as their blocks get abandoned. Selfish mining strategies vary based on the exact conditions that control when to withhold/publish blocks. Selfish mining presents a clear threat to blockchain security; it enables miners to earn more than their fair share of incentives, and to some extent control the blockchain content, at a lower hashrate threshold than the majority—thus lowering the security threshold of the network.

A myriad of research has emerged evaluating the impact of selfish mining attacks on Bitcoin [1]–[15]. Alongside devising new strategy variants that are more profitable than classical ones, these works also examine various model settings and parameterizations, such as accounting for broadcast latency [6], [7], [16] and varying the number of attackers [2], [5], [8], [10], [17], while others define optimal strategies for particular settings [4], [5]. Moreover, new strategy families have been formulated based on more granular attacker behaviors with respect to block withholding and publishing [2], [3], [9].

This expanding landscape makes it hard to understand the impact of selfish mining on blockchain security, especially across diverse model settings. At the same time, a careful inspection reveals that most existing works focused on particular paradigms, mainly the block reward model for single attackers, thus fracturing the results to be largely model-specific and leaving many gaps. A holistic understanding of selfish mining strategies and models is a key to incorporating the observations of these works in blockchain security modeling and analysis.

A. Contributions

To address this challenge, we develop a systematization of knowledge of selfish mining attacks in Bitcoin. To the best of our knowledge, our work is the first of its kind that not only systematizes existing works, but also empirically studies selfish mining under more realistic model adaptations. In particular, we make the following contributions.

Systematization framework. We develop a systematization framework covering two dimensions: *strategy formulation* and *model formulation*. To offer more granular insights, we introduce several sub-dimensions covering profitability notions and action update criteria for strategies, while for models, these sub-dimensions cover the number of attackers, incentive models, network configuration, and miner behaviors or threat models. We believe that our framework offers a versatile and holistic approach for evaluating (existing and future) selfish mining attacks, enabling a clear path to understanding them and distilling security impacts and trade-offs.

Analyzing existing works. Leveraging our framework, we analyze and categorize 24 selfish mining attack works. Our findings show that 14 of them introduce new strategies, and naturally most of them focus on maximizing profitability. In terms of model settings, we find that these works vary based on the number of attackers included—single or multi attackers, follow three incentive models—whether blocks rewards and transaction fees are considered, and many of them account for the impact of network configuration on the attack success. However, a few works consider rationality of honest miners beside the regular selfish vs. honest miner threat model.

Distilling insights and identifying gaps. Our analysis highlights numerous insights and calls attention to a number of gaps. For example, in context to model formulation, we find that only 3 works include transaction fees in their modeling; 2 consider both block rewards and transaction fees, and only 1 examines the transaction-fee regime (i.e., incentives come only from transaction fees). In a similar vein, we find that only 9 works consider the multi-attacker setting, while notably none of them accounts for transaction fees. These gaps point to a larger trend of selfish mining attacks being understudied outside the block-reward only model.

Extensive study of strategies in the transaction-fee regime. To complement the findings of our systematization study, we evaluate many of the existing strategies in the transaction-fee regime to assess their profitability both internally amongst each other, and externally compared to the block-reward model. We do not aim to close all the gaps we identified, but rather we focus on showing how transaction fees have a large impact on selfish mining profitability. Thus, in turn, we aim to motivate researchers to further study this setting.

Utilizing a mining simulator [18], we implement 11 existing strategies in the single and multi-attacker setting (to our knowledge, we are the first to study the multi-attacker setting in the transaction-fee regime). Our evaluation finds new (lower) profitability thresholds for many of these strategies, and highlights the dominant (most profitable) ones among the parameter space. Furthermore, we extend the selfish mining model to include honest-but-rational miners, thus fostering a mutually-beneficial relationship between selfish and honest-but-rational miners. That is, the former provides additional incentives to the latter—in the form of expected future revenue from transactions fees—to choose the selfish miner's fork on the public chain. As our evaluation shows, this increases profitability of existing strategies and leads to new strategy variants. Moreover, we study strategy composition in which attackers follow multiple strategies, and show further security threshold reduction.

Discussion and additional remarks. Showing security attacks corrects any misconceptions about the security of system designs and deployments, and motivates developing proper countermeasures and defenses. Thus, we conclude with a brief discussion of defenses against selfish mining. Moreover, we examine connections between selfish mining and incentivized mining strategies, the notion of miner extractable value (MEV), and fee-driven systems such as blockchain-based resource markets. Lastly, we briefly discuss current efforts on studying selfish mining in blockchains other than Bitcoin.

B. Related Work

To the best of our knowledge, there have been no prior work on systematizing selfish mining attacks and their models. There are surveys of defenses against selfish mining, e.g., [19], [20].

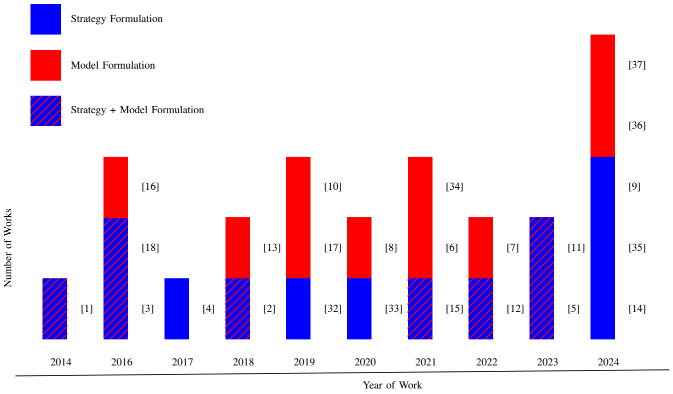


Fig. 1: Timeline of selfish mining attacks

We view these works as complementary to ours; combined they offer a holistic view of selfish mining attacks and defenses.

II. Systematization Methodology

A. Scope and Methodology

We target selfish mining attacks in Bitcoin covering papers with new results: new strategy formulations or new model parameterizations. In particular, works that: (a) target blockchains other than Bitcoin, e.g., [21]–[23], (b) reaffirm past/known results via variants of analytical modeling or evaluation frameworks, e.g., [24]–[26], (c) just mention selfish mining to justify network behavior, e.g., [27], or (d) use it to empower other mining attacks (no new selfish strategies), e.g., [28]–[31], are out of scope. Furthermore, defenses against selfish mining are out of scope; as mentioned before, there are SoKs on selfish mining defenses, but for completeness, we briefly discuss some of the works in this area in Section VII. In our search, using "selfish mining" as the search keyword, we identified 85 papers related to selfish mining attacks (these do not include defenses which account for additional 57 papers including the SoKs), which we refined based on our scope producing a list of 24 papers that we cover in our systematization study.

B. Framework

We develop a systematization framework corresponding to the features of the examined selfish mining attack studies. It covers two dimensions including attack strategies and models/settings under which these strategies have been studied. The timeline, shown in Figure 1, traces the evolution of these works across these dimensions.

Strategy formulation. This covers strategy behaviors and actions, such as when and how to build a private chain and when to publish withheld blocks. Out of the 24 works we identified, 14 of them formulate new selfish mining strategies, while the rest study existing strategies under different models. In analyzing these 14 works, we observe that strategies can be split among two dimensions: (1) *notions of profitability* and (2) *update criteria*, i.e., what triggers an action. For profitability, we find that the studied attacks can be split based on two

TABLE I: Categorization of selfish mining strategy formulation.

Category	**Dimension**	**Works**													
		[9]	[5]	[11]	[18]	[1]	[2]	[3]	[33]	[4]	[14]	[12]	[15]	[32]	[35]
Profitability	Profit maximizing	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓
	Time-to-profit minimzing								✓		✓				
Update Criteria	Lead dependent					✓	✓	✓				✓		✓	✓
	Context aware	✓	✓	✓	✓				✓	✓	✓		✓		

goals: *profit-maximizing* that aim to maximize rewards, and *time-to-profit minimizing* that aims to minimize the time it takes for a strategy to become profitable. By update criteria, we find that strategies are divided into: *lead-dependent* that are triggered solely based on updates to the lead between the height of a selfish miner's private chain and the public chain, and *context-aware* in which context from the blockchain view or the underlying protocol specifications, such as block value or mining difficulty, triggers strategy actions.

Model formulation. Selfish mining strategies are examined under various model settings. Understanding these models is critical for assessing the security impact of a particular strategy. While studying the surveyed works, we observe four categorizes that cover influencing system model factors: (1) *number of selfish miners* (or attackers) in the system, (2) *incentive model* mainly focusing on the inclusion of transaction fees when computing mining rewards, (3) *network configuration* covering parameters related to the network protocol, such as propagation delays, and (4) assumptions on *miner behavior or threat model* accounting for rationality of non-selfish miners (beside the traditional selfish vs. honest miner threat model).

We examine prior works across each of the above dimensions, and their sub-dimensions, in the next two sections.

III. Strategy Formulation

Selfish mining relies on withholding and selectively publishing locally-mined blocks. A particular strategy specifies which block to mine on, and when to publish the withheld blocks (or even abandon them and start over). Classic selfish mining [1] relies on the relative lead of the private chain, i.e., the locally withheld blocks, over the public chain's height to trigger actions. Several followup works presented variations of that. In studying these works, as shown in Table I, we classify their strategies based on profitability goals and their update criteria. We also offer fine-grained sub-categories to highlight the distinguishing factors between the various works.

A. Profitability

The goal of selfish mining is extracting additional revenue via wasting the work of other miners. Thus, a selfish miner will not attempt a particular strategy unless its revenue outperforms that of honest mining. While maximizing profitability is a typical goal [1]–[5], [9], [11], [12], [15], [18], [32], [35], Grunspan et. al [13] showed that in practice the time it takes strategies to become profitable is quite long—on the order of weeks. Accordingly, we adopt two profitability-related dimensions: profit-maximizing and time-to-profit minimizing.

Profit-maximizing. As the name implies, strategies under this category aim (for a fixed set of parameters) to maximize the expected earnings of a selfish miner. As shown in Table I, the profit-maximizing category includes the majority of selfish mining strategies. Below we discuss these strategy families, outlining their techniques and profitability thresholds. We note that profitability is meaningful within a particular model setting. Unless specified otherwise, the default for most works is the no-latency, block-rewards, and single-attacker setting, which we refer to as the *generic model*.

Classic selfish mining. This is the first selfish mining strategy to be developed [1], which is (in the worst case) profitable with 33% of the network hashrate. It occurs when a miner mines a block upon the head of the public chain, but chooses to withhold this block as part of a private chain. At this point, this private chain has a lead of 1 over the public chain. Upon publishing a new block that extends the public chain, this lead becomes 0, and so on. A classic selfish miner chooses what block to mine on and when to reveal withheld blocks according to its current lead, as follows:

- A lead > 2: A selfish miner continues to mine on its private chain. If a new block is published on the public chain, this miner reveals its oldest block (to create a fork).
- A lead of 2: If a new block is published on the public chain, the selfish miner reveals its entire chain (resulting in a longer public chain).
- A lead of 1: If a new block is published on the public chain, the selfish miner reveals its entire chain (creating a fork on the public chain).
- Having no lead, but a forked block at the top of the public chain: If the selfish miner mines the next block first (on top of its block), it immediately reveals this new block. Instead, if the opposing block is mined upon first (resulting in a negative lead for the selfish miner) the selfish miner scraps its chain, starting again to mine on top of the public chain's head.

The network security threshold (or profitability threshold) is formulated as a function of the selfish miner's connectivity rate γ (i.e., the ratio of honest miners that choose to mine on the selfish miner's block). Specifically, for a fixed γ, classic selfish mining with hashrate α is profitable when $\alpha > \frac{1-\gamma}{3-2\gamma}$.

> *Insight 1:* There is a natural inverse relationship between γ and the network security threshold. Higher γ allows the selfish miner to profit at a lower hashrate.

Stubborn selfish mining. While classic selfish mining is already profitable, subsequent profit-maximizing strategies attempted to further decrease the profitability threshold via action adaptations. Among them, stubborn selfish mining [3] allows selfish miners to continue mining on their private chains for a longer duration than that of classic selfish mining. At its core, this family can be seen as a greedier variation, thus riskier as more effort is put into the private chain. Stubborn selfish mining consists of three sub-strategies:

- Stubborn-k-trail mining T_k: A selfish miner continues to build on a private chain with a negative lead until this lead is trailing by k blocks. At that time, it discards its private chain and starts over.
- Stubborn lead mining L: Upon a lead ≥ 0, when a new block is published on the public chain, the selfish miner publishes its oldest block. Compared to classic selfish mining, a stubborn selfish miner chooses to preemptively fork while still having some lead.
- Stubborn fork mining F: During a fork containing a selfish miner's block, if the selfish miner is the first to mine the next block it withholds this block (resulting in a lead of 1) and continues to selfishly mine.

In the generic model, stubborn mining is more profitable than classic selfish mining for the majority of the parameter space, and choosing the best sub-strategy varies largely within this space. For example, for a hashrate between 33% and 45% and $\gamma \in [0, 0.3]$ stubborn-1-trail is the most profitable (among stubborn mining strategies) outperforming classic selfish mining threshold by 1.4%. While for $\gamma = 1$, the combination of stubborn lead and stubborn-1-trail is the most profitable being up to 25% more profitable than classic selfish mining.

Insight 2: Neither classic or stubborn selfish mining was dominantly profitable across the entire parameter space [3]; dominance depends on choice of parameters.

Publish-N selfish mining. In the setting of multi non-colluding selfish miners, publish-N [2] attempts to mitigate the downsides of the increased competition between these miners—the lost rewards due to this infighting. This class presents a risk-averse variant forgoing building long private chains to later be wasted by an opposing prevailing selfish miner. Publish-N acts as a truncated version of classic selfish mining; it allows a lead of up to $N-1$, and upon reaching a lead of N blocks, the oldest block of the private chain is published.

Insight 3: The choice of N in publish-N denotes risk-aversion; $N = 1$ is equivalent to honest mining and $N = \infty$ is equivalent to classic selfish mining.

While a more risk-averse strategy, and in principle less profitable than classic selfish mining, in the multi-attacker setting (among many scenario) publish-N is more profitable than classic selfish mining due to its ability to account for increased competition. We note that [2] did not show a security threshold with respect to relative revenue (relative number of valid blocks for which a miner produced), instead it framed performance in terms of a miner's relative stale block rate (number of stale blocks it produced relative to the overall number of stale blocks in the system). Thus, a relative stale block rate lower than 1 denotes profitability. The findings show that in the two-attacker setting under symmetric hashrate between attackers and $\gamma = 0.5$, publish-3 is profitable when each attacker controls 20% of the hashrate compared to around 22% hashrate under classic selfish mining.

Gap 1: A security threshold for publish-N across varying number of attackers is still an open question.

Partial selfish mining. Assuming the presence of rational miners fosters new strategies. Specifically, this allows selfish miners to collude with rational miners to establish a selfish mining pool to extend this selfish miner's private chain. Thus, this mimics classic selfish mining under a larger hashrate. Those rational miners are willing to collude only if this collusion is more profitable than mining honestly.

In this light, Yu et al. [12] presented partial selfish mining, a strategy class that exploits this relation. In turn, this encompasses additional actions. In particular, a selfish miner has to convince rational miners that a valid private chain exists, and that they will act fairly while colluding. i.e., cooperate in sharing future heads of the private chain to avoid wasting each other's work. To achieve this, [12] employs zero-knowledge proofs to prove the existence of a valid private chain. Furthermore, the selfish miner deploys a collateralized smart contract to ensure rewarding colluding rational miners for producing valid blocks that extend the private chain. Their findings show that this strategy can be mutually profitable for both rational and selfish miners. Notably, assuming 50% of the miners to be rational, partial selfish mining is more profitable than honest mining at a 20% hashrate.

Bribery selfish mining. Classic selfish mining can become more profitable if it can attract rational miners to build on its forked block at the top of the public chain. To this light, Gao et al. [32] introduced bribery selfish mining (BSM) where some fraction of the block rewards are offered as bribes to other miners in exchange for its fork adoption. Bribes are distributed using some external mechanisms, i.e., out-of-band payments. While not showing impact on the security threshold, their findings show that BSM provides additional revenue compared to classic selfish mining. Notably, if $\gamma = 0.5$, a bribery selfish miner, with a hashrate of 0.3, can expect approximately 1.64% more earnings than classic selfish mining (when 30% of the network is bribed with 0.02 of the block rewards).

Another recent work [35] showed how bribery can be combined with a stubborn mining-inspired strategy under what is called lead-hide bribery selfish mining (LHBSM). Here, upon having a lead > 2, the selfish miner creates a pseudo-trailing position—by first publishing a block and then letting the public chain surpass it in height. By bribing miners to mine on its fork of the chain, the selfish miner can introduce a 3-way fork: the public chain's block, the bribed miners' block, and its own next block. A selfish miner can then publish another block from its private chain to guarantee the inclusion of its fork. In comparison to BSM, LHBSM is more profitable. For example, given a hashrate > 0.37, LHBSM becomes more profitable than BSM (when 10% of the network is bribed with 0.02 of the block rewards).

Insight 4: Having rational miners empowers selfish miners; collusion with these miners establishes a selfish mining pool, thus effectively increasing the selfish miner's hashrate, while bribing them increases the chances of adopting the selfish miner's fork on the public chain.

Improved selfish mining. In the transaction-fee regime, Carlsten et al. [18] introduced improved selfish mining. This class utilizes the fact that without block rewards, mining rewards are not fixed but vary from block to block based on the included transactions. As such, a selfish miner will conditionally selfishly mine depending on the block value—immediately publishing high-value blocks for short term rewards, while withholding lower-value blocks to selfishly mine upon.

This class is thus parameterized by a cutoff point determining the block value to decide whether to selfishly or honestly mine. Optimal parametrization of this cutoff in accordance with the attacker's hashrate shows that selfish mining is barely more profitable than honest mining when the hash rate is below 25%. Just at a hash rate of 25%, profitability begins to diverge from that of honest mining. This is because as attacker's hashrate decreases, the cutoff value approaches 0, and thus the selfish miner will mine honestly most of the time. It was also shown that when $\gamma = 0$, where the security threshold of classic selfish mining is 33% (which is the same in transaction-fee regime as found by [18]), improved selfish mining succeeds in mining 38% of the blocks adopted by the public chain.

WeRLman strategies. A recent work [11] introduced WeRLMan, a framework for formalizing strategies employing deep reinforcement learning under block-rewards and volatile transaction fees. By exploring the strategy space, i.e., determining what action to perform based on parameters such as hashrate and knowledge of the chain, WeRLMan found that semi-frequent bumps in fees can drastically downgrade blockchain security. Notably, under high fee variability, the security threshold could be as low as 23% and that still may further degrade over time as block rewards become smaller.

Insight 5: Inclusion of transaction fees changes the impact of selfish mining, and this depends on the characteristics of these fees in terms of value and volatility.

Undetectable selfish mining. Due to the difference in behavior from honest mining, selfish mining could be detectable. The class of undetectable selfish mining strategies [9] trades off profitability for statistical undetectability.[1] Since a key indicator of detection is how wasted blocks are produced, the selfish miner tries to mimic the behavior of honest miners, which is likely to succeed under increased network latency. That is, it balances revealing additional blocks to ensure that not too many blocks on the public chain are wasted in succession, rather than just depending on the lead a selfish miner has. These strategies are found to be statistically undetectable at the expense of higher security threshold, which rises to 38.2%.

Insight 6: In contrast to stubborn selfish mining, which is a risky variation of classic selfish mining, undetectable selfish mining is a risk-averse version.

Optimal selfish mining. Instead of developing a fixed strategy, this class followed a problem optimization approach. It selects the best action (e.g., reveal a block, and whether to selfish mine and upon which block) based on the chain history and other parameters such as the selfish miner's hashrate and connectivity rate (and in the case of multi-attackers, any available information about these attackers—their number, hashrates, and strategies).

Sapirshtein et al. [4] formulate selfish mining as a single-player decision problem, and use a numerical solver to solve for the optimal actions for specific parameters. Their findings show a marginally lower worst-case profitability threshold compared to classic selfish mining, and higher profitability when exceeding this threshold. Interestingly, their results mirror those found via the combination of stubborn mining strategies in [3], being at most 1.4% more profitable than their stubborn mining strategy counterparts. Similar results were additionally found by a subsequent work [15], which utilizes deep reinforcement learning to identify optimality.

For multiple attackers, no longer does an attacker know the entire space; other competing selfish miners maintain private states. Consequently, optimal strategies formulated in the generic model do not translate directly. The multi-attacker case was first explored in [10] while assuming that a private chain may only exist up to a fixed length. In doing so, they find that in the two-attacker setting when each attacker's hashrate is within the range $20\% - 27\%$, a Nash equilibrium exists both between honest miners and the two selfish miners, with the latter being more profitable. A later work [5] removed this assumption, adapting the formulation of [4] by viewing the problem as a partially observable Markov decision process and solving for the optimal set of actions against classic selfish miners. They find that the profitability threshold is greatly reduced. In the case of two attackers, when $\gamma = 0$ and the opposing classic selfish miner has a hashrate of 34%, the optimal attacker only requires a hashrate of 2% to be profitable.

Time-to-profit minimizing. This category includes strategies that aim to minimize the time to profitability. As mentioned before, it was observed that selfish mining might take on the order of weeks to become profitable [13]. The key reason is that while a selfish miner attempts to waste other miners' efforts by excluding their valid blocks, it also risks short-term profits by the possibility that selfishly-mined blocks may be excluded as well. However, in the long term, such block exclusions lower the difficulty threshold. That is, in Bitcoin the network difficulty is adjusted every 2016 blocks (roughly two weeks). As less blocks are produced, the difficulty adjustment algorithm (DAA) will produce a lower difficulty target than expected.

[1] Semi-selfish mining [38] aims to counter detection, however it was later shown that is impossible for this strategy to be undetectable [39].

As a result, a selfish miner may earn more during this, lower difficulty, period.

Intermittent selfish mining. This class [33] utilizes a dynamic approach alternating between classic selfish mining and honest mining every other difficulty adjustment period. It relies on intermittently lowering the difficulty for mining blocks (via selfish mining) before taking full advantage of the easier mining difficulty to mine as many blocks as they can (via honest mining). Impressively, this approach is profitable within just one difficulty adjustment period. Compared to profit-maximizing strategies, intermittent selfish mining is less profitable (though in a shorter time horizon). In the worst case, where $\gamma = 0$, intermittent selfish mining has a security threshold of 37% (while classic selfish mining has a 33% threshold).

Further research [14] extrapolated this space under various DAAs (including those that factor in excluded blocks within the difficult adjustment). This variant, known as *smart intermittent mining*, not only swaps strategies (selfish and honest) between difficulty adjustment periods, but also does that within a given period. When alternating between classic selfish and honest mining on a 50% split within a given period under Bitcoin's original DAA, smart intermittent selfish mining is found to be more profitable given a hashrate of 27% and $\gamma = 0.5$ in approximately 11 weeks. This is compared to 14 weeks for optimal selfish mining under the same parameters.

In addition, when evaluating these strategies under a DAA that factors in excluded blocks, it is found that they are profitable for a hashrate of around 25% when $\gamma = 0.5$. This refutes a previous claim made by [13] that the inclusion of stale blocks within a DAA makes selfish mining unprofitable.

Gap 2: There are no works aim to simultaneously maximize profitability while minimizing time-to-profitability.

Gap 3: Performance of strategies, beyond classic selfish mining, in the intermittent setting is an open question.

Gap 4: In general, temporal composability of strategies across families (so a selfish miner alternates between different selfish strategies over time) is yet to be studied.

B. Strategy Update Criteria

Selfish mining strategies rely on different criteria to trigger their actions. For example, where to mine, when to withhold a block, and when to publish a withheld block to the public chain. As shown in Table I, the update criteria can be categorized into two classes: (1) *lead-dependent* relying only on the relative lead of the private chain, and (2) *context-aware* relying on additional context from the blockchain or its protocol.

Lead-dependent strategies. These strategies trigger actions upon updates to the lead of a selfish miner's private chain over the public chain. This category includes classic, stubborn, publish-N, bribery, and partial selfish mining. As noted, early mining strategies rely on having a lead ≥ 0. This is natural since it usually gives advantage to a selfish miner (with a potential of having a longer private chain) to have its chain adopted by the network due to the longest branch rule. Having a negative lead is adopted by stubborn-k-trail mining; although it is counter-intuitive and riskier, it was shown in [3] that this strategy is more profitable than classic selfish mining, e.g., for an attacker with hashrate above 33%, it outperforms classic selfish mining by upwards of 13%. Still, trail lead is naturally bounded; a selfish miner would not continue if the gap becomes so big between its private chain and the public chain.

Context-aware strategies. On the other hand, some strategies rely on additional information about the blockchain content, and/or its protocol specifications, when making decisions. This category includes intermittent, improved, and optimal selfish mining, as well as the WeRLman's strategies.

In intermittent selfish mining, swapping between selfish and honest mining is tied to how mining is happening in Bitcoin—what difficulty level the network is currently operating on, and how this difficulty is adjusted. For improved selfish mining, the context is the current block value, given its conditional nature to decide upon whether to selfishly mine. Finally, for optimal and WeRLman's strategies, the context is the blockchain state (information about the current block), as well as the history of past strategy actions and their impact on profitability seen so far. Context-aware strategies add an opportunistic nature to selfish mining, i.e., exploit any profit-improving opportunities.

Insight 7: Context-aware update criteria enables a more adaptive selfish mining than lead-dependent due to taking advantage of the underlying model settings.

IV. Model Formulation

The generic model (no-latency, block-rewards, single-attacker) has been the foundation for many of the works discussed so far. While impactful in understanding Bitcoin security, the system model in practice differs from this idealistic setting. Such differences include the possibility of having multiple selfish miners, the effect of network propagation delay on participants' view of the blockchain and block adoption, the incentive model in terms of miner revenue sources, and how miners react to changing the incentive model.

While analyzing prior work, we observe that a more realistic system model highlights not only how strategy performance varies between model settings, but also its ability to foster new strategies. We identify four main categories of model formulation (based on the additional factors considered over the generic model): (1) number of attackers (non-colluding selfish miners), (2) incentive model or mining reward sources, (3) network configuration, and (4) miner behavior (or threat model in terms of whether other miner behaviors, in addition to the selfish vs. honest behavior, are included).

A. Number of Attackers

The number of selfish miners plays a key role in the performance of an individual mining strategy. This is because

TABLE II: Selfish mining works by number of attackers.

# of Attackers	Profit-maximizing	Time-to-profit Minimizing
1	[1], [3], [4], [6], [7], [9], [11]–[13], [15], [16], [18], [32], [34], [35]	[14], [33]
2	[2], [5], [6], [8], [10], [15], [17], [36], [37]	
3+	[2], [5], [6], [8], [10], [15]	

these multiple attackers are basically competing with each other, making the effectiveness of a particular strategy questionable under the partial view of the system (i.e., no knowledge of other selfish miners' withheld blocks or decisions).

Table II classifies prior works based on the number of attackers considered. Some works cover evaluations of prior strategies originally examined in the generic model, such as classic and stubborn-trail selfish mining. While others introduce new strategies stemmed from having multiple attackers, such as publish-N and optimal selfish mining. As shown, the majority (14 works) considered only single attackers.

In terms of evaluating prior strategies under multiple attackers, existing studies, however, mainly focused on classic selfish mining [6], [8], [17], [36], [37]. Bai et al. [17] found that multiple attackers lower the profitability threshold necessary for an individual attacker. For example, for $\gamma = 0.5$, each attacker needs a hashrate of 21.48% to be profitable (compared to 25% in the single attacker setting). Their analytical model was later tightened by Wang et al. [36]; in comparison to simulation-based results, the model from [36] saw an average profitability difference of just 2.4% compared to 7.35% for the model from [17].

In other instances, existing strategies have been examined alongside new strategies specific to the multi-attacker setting. Such is the case in [2], which evaluates publish-N against classic and stubborn-trail selfish mining. In the two-attacker setting, they show that publish-3 outperforms both classic and stubborn-1-trail selfish mining at higher connectivity rates. They also show that strategy comparisons in the multi-attacker setting are often complex, having to consider both the examined strategy and strategies of the opposing attacking miners.

A counterpoint to the increased effectiveness is that for a strategy to be appealing, it now must be profitable for all attackers. An attacker who is not profiting would switch to honest mining instead. This will impact the once profitable miner (to the point of becoming unprofitable) since its profitability may have relied on having another active selfish miner. Hence, profitability in the multi-attacker setting could be increasingly volatile and highly dependent on the joint profitability of all attackers. Zhang et al. [8] confirmed this relation, and Wang [37] further analyzed this relationship among pairs of miners with different hashrates. Notably, as the number of attackers increases, while the profitability threshold for an individual attacker decreases, so does the parameter space where selfish mining is jointly profitable for all attackers.

Finally, optimal profit-maximizing strategies under a certain number of attackers have been studied. This was first formulated under a simplified model by [10], then later, [5] established an optimal policy when competing against classic selfish miners. The deep reinforcement learning based approach [15] affirmed that the optimal strategy from [4], which is for single attackers, is not optimal in the multi-attacker setting. Moreover, their results suggest—though does not prove–that when there are ≥ 3 attackers in the block-reward-only setting, there may not exist a profitable Nash equilibrium. All these optimal strategies are context-aware as they rely on knowing the number and strategies of opposing attackers.

Gap 5: Optimal selfish mining for the multi-attacker setting is currently limited to comparisons against only classic selfish mining.

Gap 6: The performance of time-to-profit minimizing strategies in the multi-attacker setting is an open question.

B. Miner Incentive Model

The inclusion of transaction fees (either in conjunction with block-rewards, or replacing it) is one of the most understudied areas in the evaluated selfish mining works. In Table III we highlight this gap, systematizing works across the various incentive models they adopted. Notably, only three works [11], [13], [18] considered transaction fees.

In [18], block rewards were replaced with transaction fees, so they study the transaction-fee regime. They evaluate classic selfish mining against improved selfish mining that takes advantage of the fees. Their findings highlight that classic selfish mining has a profitability hashrate threshold just marginally lower than that in the generic model, while improved selfish mining is more profitable than classic selfish mining—affirming the ability of varying the incentive model to foster new strategies.

In the setting where transactions fees are included alongside block rewards, [11] examines potential selfish miner's actions while assuming that these fees are volatile with infrequent jumps in block value. They highlight a relation between the expanding significance of transaction fees (i.e., transaction fees become a large part of miners' revenue) and blockchain insecurity, with security continuing to degrade as block rewards get smaller (due to Bitcoin block reward inflation policy). On the other hand, [13] utilizes transaction fees in analyzing time-to-profitability of classic selfish mining (this work did not aim to minimize this time, just analyzing it). They found a varying window of time-to-profitability ranging from 2 weeks to 100 weeks depending on fee parametrization.

Gap 7: There is no comprehensive assessment of the impact of transaction fee inclusion on many existing selfish mining strategies. Also, optimal selfish mining in the transaction-fee regime has not been studied yet.

TABLE III: Classification of selfish mining strategies based on the incentive model.

Strategies / Model Setting	Block-reward Model		Block Reward + Tx Fees Model		Transaction-fee Regime	
	Single Attacker	Multi-attacker	Single Attacker	Multi-attacker	Single Attacker	Multi-attacker
Classic selfish mining	[1], [3], [6], [7], [16], [34]	[2], [6], [8], [10], [17], [36], [37]	[13]		[18], This work	This work
Stubborn selfish mining	[3]	[2]			This work	This work
Publish-N		[2]			This work	This work
Other strategies	[4], [9], [12], [14], [15], [32], [33], [35]	[5], [15]	[11]		This work	This work

C. Network Configuration

Most prior work studied a simplified model of Bitcoin network, namely, abstracted away propagation delay by assuming no latency. However, latency can play a crucial role in selfish mining performance as shown by [4], [6], [7], [16].

Varying approaches, such as simulation [6], stochastic modeling [7] and analytical modeling [4], [16], [34], have been used in such evaluations. However, they have largely been limited to classic selfish mining, except for [4] which developed an optimal solution. Also, they were limited to the single-attacker scenario except [6]. Interestingly, the results show that propagation delay can in fact make selfish mining attacks more profitable, and lower their profitability hashrate threshold. The reason is that non-zero propagation delay provides additional time for selfish miners to selfishly mine, further benefiting their strategies. Finally, as mentioned before, undetectable selfish mining relied on mimicking the behavior of honest miners under non-zero latency to prevent detection.

Gap 8: Studying the impact of propagation delay on selfish mining strategies, other than classic selfish mining, in both the single and multi-attacker settings, is an understudied area.

D. Miner Behavior (or Threat Model)

Selfish mining is inherently reliant on wasting the computation of opposing miners. As such, it is highly affected by the assumptions regarding the opposing miner's behavior—whether they are honest or, in the multi-attacker setting, what selfish mining strategies they adopt.

Insight 8: Based on the limited existing studies, and our own examination in Section VI, miners' threat model highly impacts the effectiveness of existing selfish mining strategies and may foster new strategy variants.

While a critical component, prior works are largely limited with respect to their threat model. In the single attacker setting, all works (except [12], [32], [35]) assume only the presence of honest miners (beside the selfish one). In practice, whereas not all miners will choose to mine selfishly, due to incentives, many may still wish to adapt their mining behavior in the most profitable way. That is, [12] showed that rational miners may choose to collude with a selfish miner and help in extending its private chain (rather than selfishly mine on their own), while [32], [35] included rational-honest miners who would choose the selfish miners' fork on the public chain to collect a promised bribe. These works show that indeed such extended threat model promotes profitability of selfish mining attacks.

Adding transaction fees may further promote this relation. It has already been found that for other (non-selfish mining) attacks transaction-fees can act as a helpful tool for attracting rational miners [18], [40]. Though it is yet to be validated for selfish mining attacks.

Gap 9: Further studies are still needed to examine how transaction fees can be utilized to sway rational miners, and thus, improving profitability of selfish mining.

In the multi-attacker setting, the threat model additionally includes the behavior of the opposing attacking miners in terms of their strategies. This impacts competition, and in turn, attack effectiveness. For example, in the two-attacker setting, the performance of classic selfish mining opposed to classic selfish mining and honest miners will differ from that of an opposing publish-3 selfish miner and an honest miner. While not extensive, as discussed in Section IV-A, a few works [2], [5], [15] have confirmed this relation.

Gap 10: Further studies of the impact of miner rationality in the multi-attacker setting are still needed.

V. Selfish Mining in the Transaction-fee Regime

Among the notable gaps in prior work is that most of them targeted the block reward model. This is a result of viewing transaction fees as negligible compared to block rewards. However, recent years witnessed huge spikes in transaction fee values, and due to Bitcoin's deflationary policy, continuing to halve its block rewards approximately every 4 years, block rewards will in turn become negligible.

The few works that accounted for transaction fees— [18] studied the transaction-fee regime, while [11] combined fees with block rewards—confirmed that this inclusion influences the profitability of selfish mining. However, both works were limited to the single-attacker setting. So it is important to study this regime and understand its impact on blockchain security, and examine how existing results would change when accounting for both multiple attackers and transaction fees.

We attempt to bridge this gap by evaluating existing selfish mining strategies, and variations thereof, in the transaction-fee regime for both single and multiple attackers. We do not aim to close all gaps we identified previously, which amounts to a separate work on its own. But rather we want to provide a more inclusive view of selfish mining by accounting for this regime, and thus motivate researchers to further examine this understudied area. In this section, we contextualize our system model and the strategies we examine, while in the next section we present the results (and insights) of our evaluations.

A. System Model

We model the Bitcoin blockchain under the scenario that block-rewards have been entirely phased out. Therefore, miners are compensated entirely via the fees from transactions included in the mined block. Our primary focus is assessing the profitability of selfish mining strategies in terms of their relative revenue. For a miner i running strategy s, with hashrate h, its relative revenue $Rev_h(s)$ refers to the total value (i.e., transactions fees) of blocks produced by miner i divided by the total value of all blocks in the chain.

We define a strategy s with hashrate h to be profitable if its relative revenue is higher than that of the honest mining strategy H with the same hashrate, i.e., $Rev_h(s) > Rev_h(H)$. Moreover, we denote the profitability threshold for a strategy s to be the minimal h such that $Rev_h(s) > Rev_h(H)$.

Following [18], the only work that studied the transaction-fee regime, the system is modeled as a game of a sequence of rounds, wherein each round, a constant amount of new transactions, and thus transaction fees, are added to the network. We assume no transaction backlog, so a newly mined block will empty the queue. Hence, the value of a block is the sum of transaction fees per round multiplied by the time (in rounds) since its parent block was mined. We also assume that each miner shares the same view of transactions, i.e., all transactions are coming from a shared public mempool. A miner publishes the newly mined block according to its specified strategy. After a series of rounds, we compute the number of blocks included in the public chain, determining the number of blocks produced by each miner and so its total earnings. Our game modeling involves three types of miners:

- *Honest miners* who strictly follow the protocol, and immediately publish newly mined blocks.
- *Honest-but-rational miners* who immediately publish newly mined blocks, but if there is a fork, they choose to mine on the block that provides the highest utility, i.e., one that leaves the largest transaction fees to be included in the next block.[2] We note that this is a different behavior from the one in [12], [32], [35], where the additional utility is an out-of-band payment/bribery offered to rational (selfish or honest) miners.
- *Selfish miners* who employ some selfish mining strategy.

We evaluate the profitability of various selfish mining strategies under two settings for our game:

Single-attacker setting. There are at most three (collective) parties: an honest miner with hashrate β, an honest-but-rational miner with hashrate κ, and a selfish miner with hashrate α, where $\beta + \kappa + \alpha = 1$. We refer to each party based on their hashrate fraction. Upon encountering a fork, where two blocks are published simultaneously, honest miners mine on the first block they receive. We define γ as the fraction of honest miners (in terms of their collective hashrate β) who build on a selfish miner's block (so they received this block first). For honest-but-rational miners, they will choose the block that leaves the most transaction-fees behind, so we define ω as the fraction of their hashrate that builds on the selfish miner's block.[3]

Multi-attacker setting. Here, there are $n > 1$ non-colluding selfish miners competing. As above, we denote each party by its hashrate fraction. However, we further define α_i as the hashrate of the i^{th} selfish miner, such that $\alpha = \sum_{i=1}^{n} \alpha_i$ and $\beta + \kappa + \alpha = 1$. With multiple selfish miners, a fork may contain more than two branches (i.e., $k > 2$ attackers reveal their blocks concurrently). As such, it is necessary to further consider the hashrate of honest miner's building off of each selfish miner's block in a fork. As before, we denote the overall fraction of honest miners (in terms of their hashrate) building off selfish miners' blocks upon a fork as γ. Since we have multiple attackers, and a fork may contain k selfishly mined blocks, honest miners could be distributed among them. To account for that, we denote θ_i to be the fraction of honest miners' hashrate building on the i^{th} selfish miner's block, where $i \in \{1, \ldots, k\}$ and $\sum_{i=1}^{k} \theta_i = 1$. Lastly, we use ω to denote the fraction of honest-but-rational miners hashrate mining on a selfish block.[4]

B. Selfish Mining Strategies: Existing and New Variants

We present the existing strategies that we evaluate in the transaction-fee regime and in the presence of honest-but-rational miners, followed by a new variant that we devise based on this regime, in addition to a set of composed strategies.

Existing strategies. We focus our examination on the larger class from previous sections; profit-maximizing strategies, which include: classic selfish mining (denoted as S), stubborn selfish mining—stubborn-lead (L), stubborn-fork (F), and stubborn-k-trail selfish mining (T_k), and publish-N selfish mining (P_n). Although classic selfish mining has been analyzed within the transaction-fee regime, we reevaluate it to provide a baseline for our benchmarks.

New strategy variant—Incentivized trailing selfish mining. The presence of honest-but-rational miners introduces a new dimension for evaluating the impact of selfish mining. That is, an honest-but-rational miner will select which block to build on based on what maximizes its utility. A selfish miner, then, can improve the likelihood of its block to be included on the public chain by offering incentives to these honest-but-rational miners. This is demonstrated in Figure 2; having honest-but-rational miners extends the finite state machine (FSM) of classic selfish mining by introducing new transitions (i.e., the action of a miner mining a new block) connecting the states (which represent the lead of the selfish miner's private chain). Similar to [1], we denote the state where a selfish miner has no lead but does have a block contained in the a fork as $0'$.

Under this model, and compared to classic selfish mining, a selfish miner may have some number of miners mining upon

[2] If an honest-but-rational miner's own block is one of the fork options, the value of this block is included in its utility evaluation.

[3] It is expected that ω is either 0 or 1 as one branch is typically more profitable than the other. In case of a tie, ω will be some value within [0,1].

[4] Again, this will typically be 0 or 1. If multiple blocks of a fork have the same value, honest-but-rational miners decide between them with uniform probability. Since we believe that such value-tie situation is rare, we omit formalizing the partial distribution of ω among the forks.

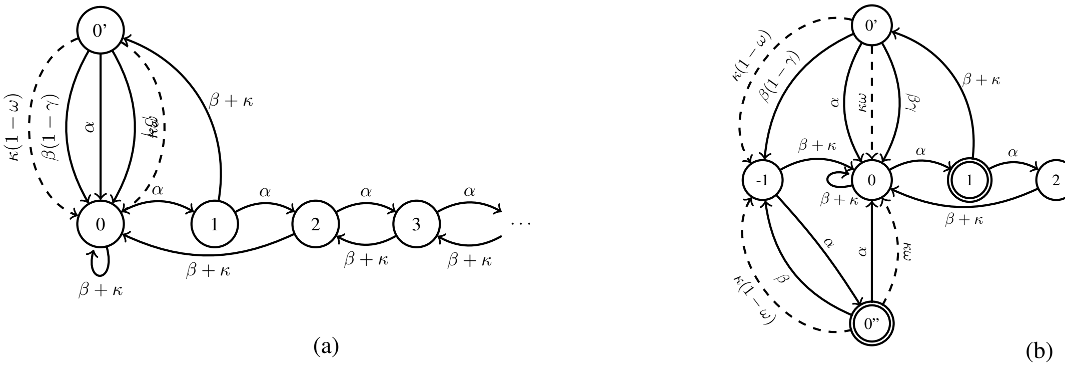


Fig. 2: FSMs for (a) classic selfish mining and (b) incentivized trailing selfish mining both in the presence of honest-but-rational miners. Dashed lines represent a new transition introduced by having honest-but-rational miners. For readability, we denote the state transition $\beta + \kappa$ by a single line instead of two parallel lines.

its block during a fork—the fraction γ of honest miners who received this selfishly-mined block first—but with little ability to adjust this parameter. This is amplified for trailing selfish mining strategies; upon a fork from a lead-trailing position, no honest miners are expected to mine upon a selfish miner's block (in the no-latency setting). On the other hand, by incentivizing honest-but-rational miners, a selfish miner might be able to improve the adoption of this block. Based on this observation, we introduce the following strategy variant.

Incentivized k-trailing selfish mining $I_{(f,k)}$. Inspired by a double-spending attack strategy [40] and an undercutting technique for fork selection [18], the $I_{(f,k)}$ class allows a k-trailing selfish miner to incentivize honest-but-rational miners to choose its fork (from a trailing position) via the availability of future transaction fees. That is, collecting these future fees is conditioned on choosing the trailing fork. Upon publishing a block from a trailing position resulting in a fork, the selfish miner will release a transaction with fee f that becomes available to future miners contingent on the inclusion of its own block. Honest-but-rational miners then will be incentivized to mine upon the selfish block with the hope of collecting f.

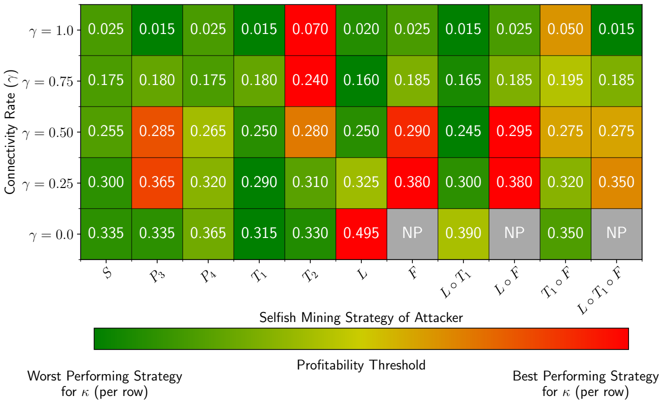
To accomplish this, the selfish miner must take preemptive action, which is issuing some transaction that facilitates the subsequent (incentivizing) transaction. In Figure 2b, we outline the actions of this strategy class for incentivized stubborn 1-trailing selfish mining. The resulting FSM extends that of stubborn-1 trail mining, additionally specifying states where upon entering, a transaction needs to be released with a double border, i.e., some external action needs to take place at this state. Alongside the original model of stubborn-1-trailing selfish mining, we specify a state -1 to have a leading of -1 over the public chain and $0''$ to be that of a fork resulting from a trailing position. Upon entering state 1, a selfish miner releases its facilitatory transaction, whereupon entering state $0''$, the transaction with an additional fee f is released.

Beside the release timeline of these transactions, their specifications also matter. Upon entering state 1, the selfish miner issues tx_A with a fee that is significant enough for its inclusion in the next block on the public chain (the selfish miner does not include this transaction in its own future blocks). tx_A transfers an X amount of currency between two addresses owned by the selfish miner. Next, upon entering state $0''$, the selfish miner issues transaction tx_B with fee f from the same address that issued tx_A, attempting to send X coins to some third address also owned by the selfish miner. As a result tx_B will only be valid within the selfish miner's chain.

This strategy allows for flexible incentives; the selfish miner can choose f as it wishes. Additionally, transactions fees are not lost from the incentivized trailing miner's own block of the fork, and paying incentives to honest-but-rational miners is contingent on them building off its own block.

Composed strategies. In our evaluations, we denote the composition of strategies with $\circ$. This refers to a strategy variant combining the action criteria of the composed strategies, enabling us to provide additional comparisons in the transaction-fee regime. This is inspired by [3] who showed that stubborn trailing, fork, and lead selfish mining strategies actions can be combined with each other. Similarly, we note that the class $I_{f,k}$ can be combined with lead and fork strategies. Accordingly, we examine the following compositions.

$L \circ T_1$ allows a stubborn-lead selfish miner to continue mining on a private chain with a negative lead of 1. $L \circ T_1 \circ F$ adds the fork behavior of stubborn fork selfish mining; during a fork occurring at a selfish miner's block, if the selfish miner mines the next block first (on top of its forking block) it withholds this block and begins to selfishly mine. $L \circ F$ combines the lead behavior of stubborn lead selfish mining (i.e., always publish its oldest block when a new block is mined on the public chain) with the fork withholding property of F. $T_1 \circ F$ combines this fork property with the negative lead capability of stubborn trail mining. Lastly, $I_{(f,1)} \circ L$, $I_{(f,1)} \circ F$, and $I_{(f,1)} \circ L \circ F$ are similar to $L \circ T_1$, $T_1 \circ F$, and $L \circ T_1 \circ F$, respectively, but now $I_{(f,1)}$ is used instead of T_1, i.e., allowing negative lead with honest-but-rational miner incentivizing capability.



Connectivity Rate (γ)	S	P_3	P_4	T_1	T_2	L	F	$L \circ T_1$	$L \circ F$	$T_1 \circ F$	$L \circ T_1 \circ F$
$\gamma = 1.0$	0.025	0.015	0.025	0.015	0.070	0.020	0.025	0.015	0.025	0.050	0.015
$\gamma = 0.75$	0.175	0.180	0.175	0.180	0.240	0.160	0.185	0.165	0.185	0.195	0.185
$\gamma = 0.50$	0.255	0.285	0.265	0.250	0.280	0.250	0.290	0.245	0.295	0.275	0.275
$\gamma = 0.25$	0.300	0.365	0.320	0.290	0.310	0.325	0.380	0.300	0.380	0.320	0.350
$\gamma = 0.0$	0.335	0.335	0.365	0.315	0.330	0.495	NP	0.390	NP	0.350	NP

Selfish Mining Strategy of Attacker
Profitability Threshold
Worst Performing Strategy for κ (per row)
Best Performing Strategy for κ (per row)

Fig. 3: Profitability threshold of lead-dependent strategies.

VI. Experimental Evaluation

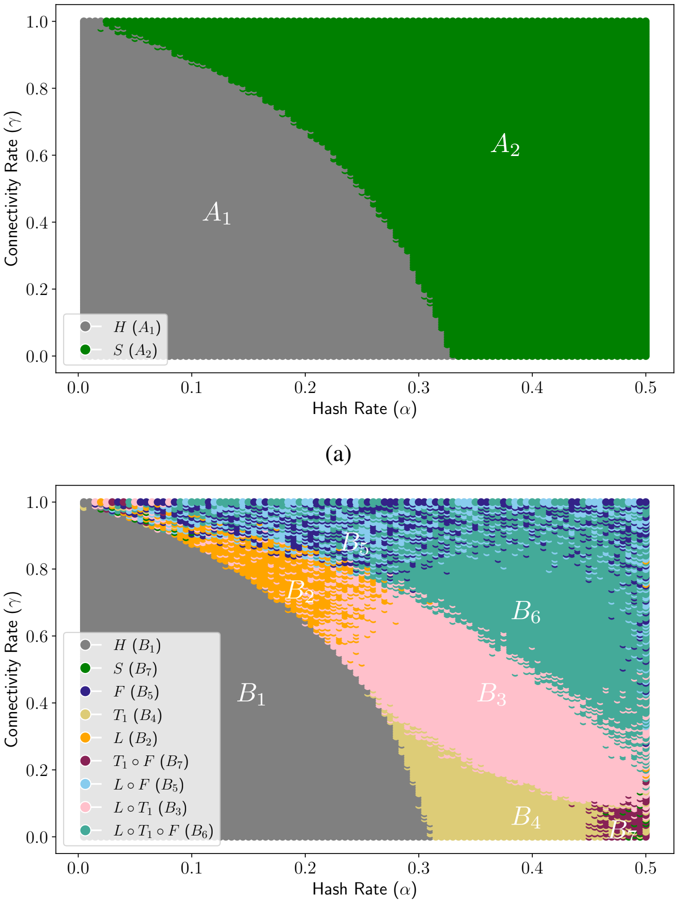
Our evaluation of the transaction-fee regime, for both the single and multi-attacker settings, is split among two different model settings; with and without honest-but-rational miners. The former has the goal of bridging the gap in understanding the performance of existing strategies within this regime. While the latter examines the impact on profitability in this regime, for both existing strategies and the new variant presented in the previous section, when including honest-but-rational miners.

A. Implementation

We extended the mining simulator from [18] by implementing the set of strategies from Section V-B, and including an honest-but-rational actor (our code can be found at [41]). Our simulator is round-based, a property inherited from the original simulator, where each round represents 1 second during which each miner attempts to mine a block with a success rate based on its hashrate. To model Bitcoin, we set the simulator to mine a block on average every 10 minutes (600 rounds). To highlight strategy performance rather than network setting, we assume no propagation delays. At the end of a game, we compute the revenue of selfish miners as explained in Section V-A. For each experiment, we perform 100 runs, each consisting of $10,000$ blocks, with 95% confidence interval while taking the lower bound of this interval when computing the revenue.

B. Single Attacker Setting

Without honest-but-rational miners. We evaluate a range of existing lead-dependent strategies within this setting: classic selfish mining, publish-N, and 7 variants of the composable class of stubborn selfish mining strategies. We set $\gamma \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and compute the respective profitable hashrate threshold for each strategy, i.e., when the selfish mining strategy becomes more profitable than honest mining. Figure 3 shows a row-specific heatmap of the best and worst performing strategy for a specific γ for hashrates $h \in (0, 0.5)$ with increments of 0.005. In the figure, dark red highlights the worst threshold in a row and dark green highlights the best. Additionally, we denote a strategy to be not profitable, NP, if it is unprofitable under any hashrate (all subsequent figures use the same notation/color code).



(a)

(b)

Fig. 4: Dominant lead-dependent strategies by parameterization (γ, α). For (a), selfish mining versus honest mining is evaluated, in (b) this is extended to all lead-dependent strategies.

> *Result 1:* The security threshold of stubborn selfish mining is found to be roughly 2% lower in the transaction-fee regime than within the block-reward only model. Lowering the threshold from 33% to 31.5% when using stubborn-1-trailing selfish mining with $\gamma = 0$.

To further understand the performance of these strategies within the transaction-fee regime, we analyze their performance across the full parameter space. Specifically, we evaluate the dominant strategy—the one with the highest revenue—for each pair of γ, α for all $\gamma \in [0, 1]$ and $\alpha \in (0, 0.5)$ (with 0.005 increments). It should be noted that any non-honest strategy that is dominant, is also inherently profitable compared to honest mining under the same pair of parameters.

Figure 4a shows the dominant strategy between an honest and a classic selfish miner, and this is extended in Figure 4b to show dominance across all evaluated lead-dependent strategies. Where additional strategies examined, they tend to outperform

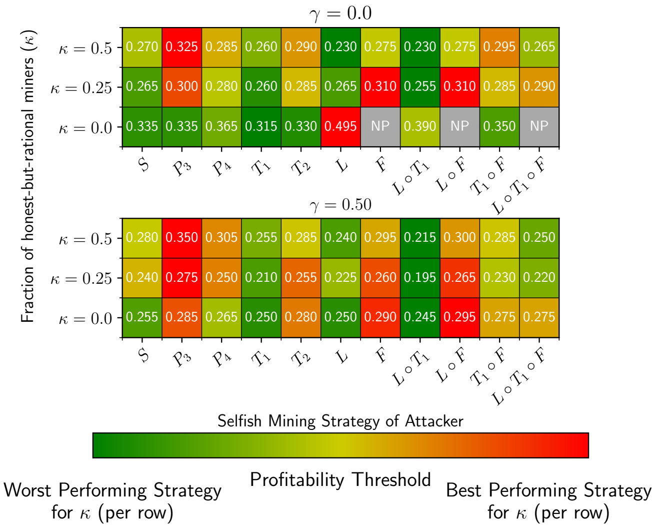


Fig. 5: Profitability threshold of lead-dependent strategies in the presence of honest-but-rational miners.

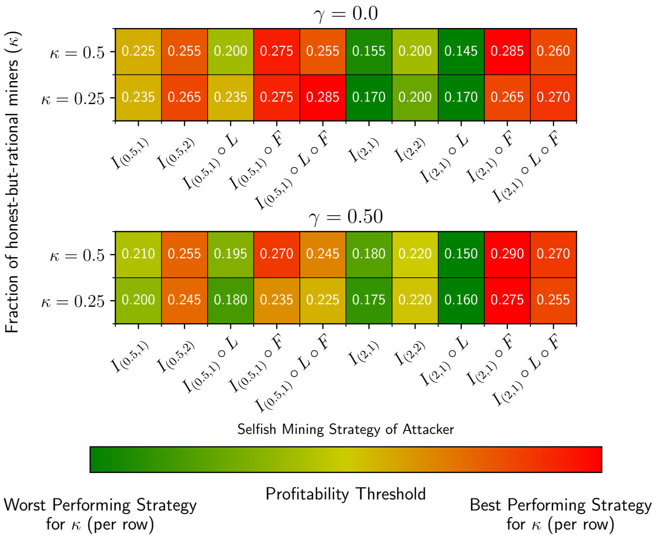


Fig. 6: Profitability threshold of $I_{(f,k)}$.

classic selfish mining (S) for regions $B_2 - B_6$, and the dominant strategy is not S. Whereas for region B_7, while $T_1 \circ F$ is still the majority dominant strategy, few select points show that S is more performant. Moreover, we see that no single strategy is dominant over the whole parameter space; dominance varies for different parameter regions (see regions B_2, B_3, B_4, B_6, B_7 in Figure 4b). Even for some parameter regions there is no single dominant strategy, such as region B_5.

Interestingly, our results show that publish-N is not dominant under any parameter configuration. We believe this is because publish-N is basically a truncated version of classic selfish mining and thus more risk averse; a feature beneficial in the multi-attacker setting (as we show later), but a limiting one in the single-attacker setting since there is no competition.

Result 2: Agreeing with the block-reward model results [3], stubborn selfish mining in the transaction-fee regime has no single dominant lead-dependent strategy.

Result 3: For regions where selfish mining is more profitable than honest mining, examined lead-dependent strategies tend to outperform classic selfish mining (except under select parameters in B_7 as shown in Figure 4).

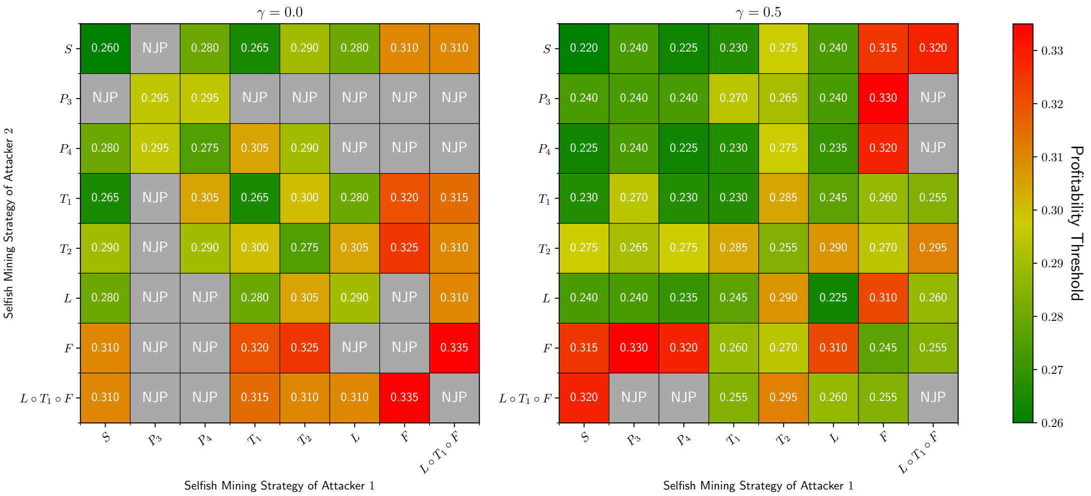
With honest-but-rational miners: existing strategies. We examine existing strategies in the presence of honest-but-rational miners. Our results are shown in Figure 5 for $\kappa \in \{0, 0.25, 0.5\}$ and $\gamma \in \{0, 0.5\}$.

Result 4: Having honest-but-rational miners further improves profitability of lead-dependent strategies. Notably, assuming 25% honest-but-rational miners (i.e., $\kappa = 0.25$) and $\gamma = 0$, the profitability threshold of classic selfish mining is lower by 7.5%.

Interestingly, in Figure 5, we see that the profitability threshold reduction is not directly related to an increase in κ. Under $\kappa = 0.5$, the threshold is equal to or higher than that of $\kappa = 0.25$ for all strategies. We believe that this is due to honest-but-rational miners choosing to maximize their utility, i.e., they additionally include the revenue garnered from the inclusion of their own block in the fork selection process. As a result, when honest-but-rational miners control a large hashrate (i.e., a large κ), so does the likelihood that one of their own blocks is part of the fork, effecting their utility and mining choice decisions. Across choices of κ, we find a key reason for the lower profitability of these strategies; coming from a leading position, a withheld block of a selfish miner is mined earlier than an honest miner. As such, upon a fork containing a selfish miner's block, it is in the honest-but-rational miners best interest to mine upon the selfish miner's block—being mined earlier, this block leaves additional transactions on the table to be included in future blocks.

With honest-but-rational miners: new strategy variant. Finally, we examine the incentivized trailing selfish mining strategy class $I_{(f,k)}$ in the presence of honest-but-rational miners. We evaluate 10 variants, across varying incentive value f, trailing positions $k \in \{1, 2\}$, and compositions with other stubborn selfish mining strategies. For an incentivized selfish mining strategy $I_{(f,k)}$, we note the provided incentive f as a factor of the expected block value, and evaluate it for $f \in \{0.5, 2.0\}$.

Figure 6 shows the profitability threshold for this strategy class where $\kappa \in \{0.25, 0.5\}$ (our profitability threshold evaluation includes the additional payment of incentive f in its determination). Compared to Figure 5, we find that $I_{(f,k)}$ results in lower profitability thresholds. Notably, when $\gamma = 0$ and $\kappa = 0.5$, we find that $I_{(2,1)} \circ L$ is profitable at a hashrate of just 14.5%. We attribute this to the ability of incentivized-trailing-selfish mining to conditionally incentivize honest-but-



$\gamma = 0.0$

	S	P_3	P_4	T_1	T_2	L	F	$L \circ T_1 \circ F$
S	0.260	NJP	0.280	0.265	0.290	0.280	0.310	0.310
P_3	NJP	0.295	0.295	NJP	NJP	NJP	NJP	NJP
P_4	0.280	0.295	0.275	0.305	0.290	NJP	NJP	NJP
T_1	0.265	NJP	0.305	0.265	0.300	0.280	0.320	0.315
T_2	0.290	NJP	0.290	0.300	0.275	0.305	0.325	0.310
L	0.280	NJP	NJP	0.280	0.305	0.290	NJP	0.310
F	0.310	NJP	NJP	0.320	0.325	NJP	NJP	0.335
$L \circ T_1 \circ F$	0.310	NJP	NJP	0.315	0.310	0.310	0.335	NJP

$\gamma = 0.5$

	S	P_3	P_4	T_1	T_2	L	F	$L \circ T_1 \circ F$
S	0.220	0.240	0.225	0.230	0.275	0.240	0.315	0.320
P_3	0.240	0.240	0.240	0.270	0.265	0.240	0.330	NJP
P_4	0.225	0.240	0.225	0.230	0.275	0.235	0.320	NJP
T_1	0.230	0.270	0.230	0.230	0.285	0.245	0.260	0.255
T_2	0.275	0.265	0.275	0.285	0.255	0.290	0.270	0.295
L	0.240	0.240	0.235	0.245	0.290	0.225	0.310	0.260
F	0.315	0.330	0.320	0.260	0.270	0.310	0.245	0.255
$L \circ T_1 \circ F$	0.320	NJP	NJP	0.255	0.295	0.260	0.255	NJP

Fig. 7: Profitability threshold of an individual attacker in the two-attacker setting.

rational miners upon fork resulting from a trailing position of this miner due the projection of collecting future profits (i.e., transaction fees that the selfish miner left behind).

Result 5: The class $I_{(f,k)}$ outperforms trailing selfish mining in the presence of honest-but-rational miners. In some cases producing large drops, e.g., we see a 7.5% drop in the profitability threshold for $L \circ T_1$ vs. $I_{(2,1)} \circ L$.

Insight 9: The class $I_{(f,k)}$ further improves the mutually-beneficial relationship between honest-but-rational and selfish miners; by incentivizing the former during fork selection—from both a leading and trailing position—the latter could improve the likelihood of having their forked blocks adopted in the public chain.

C. Multi-attacker Setting

Without honest-but-rational miners. We explore existing strategies in the multi-attacker setting, specifically, for two non-colluding attackers. Our evaluations cover 64 combinations in terms of which strategy each attacker is employing. We use the notation (a, b) to represent each combination, i.e., attacker 1 employs strategy a and attacker 2 employs strategy b. In line with previous studies on the multi-attacker setting [8], we examine the profitability threshold when it is profitable for both attackers. We note that our evaluations provide the first assessment of selfish mining strategies in the multi-attacker setting under the transaction-fee regime.

Figure 7 shows the joint profitability threshold for 64 various strategy combinations for $\gamma = \{0, 0.5\}$ and a cumulative hashrate $\alpha \in (0, 0.7)$ with increments of 0.01 such that both attackers have equal hashrates ($\alpha_1 = \alpha_2 = \alpha/2$). Additionally, we assume in the case of a fork containing both attackers, that $\theta_1 = \theta_2 = 0.5$ and denote the joint profitability threshold of the combination by the individual profitability threshold of one such attacker due to its symmetric nature. Also, for all hashrates, when one or more of the strategies underlying the combination are unprofitable, we denote it as not-jointly-profitable (NJP).

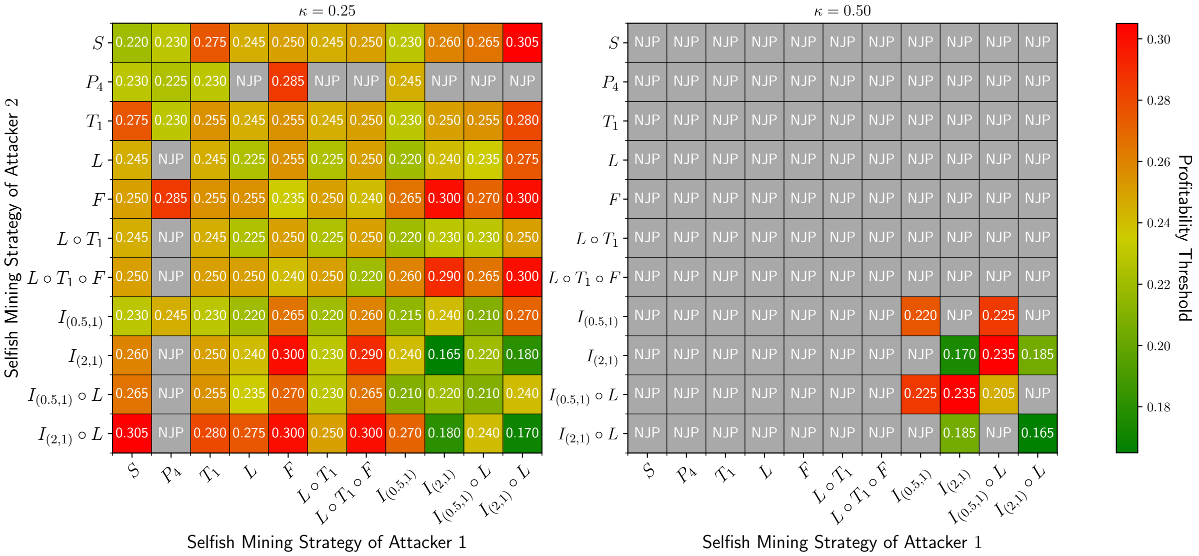
As observed from the figure, in general the riskier mining strategies, that tend to perform better in the single-attacker model (i.e., T_1 and L, as seen in Figure 3), perform worse in the two-attacker setting. That is, strategies such as T_1 and L that choose to further profit by attempting to waste additional resources at the increased risk of their blocks not being included perform worse in the competitive multi-attacker setting. On the other hand, we find that more risk-averse strategies tend to perform better. For example, we see that classic selfish mining and publish-N have better profitability thresholds.

Result 6: In the two-attacker setting with equal hashrates among attackers, we observe an inverse relationship between the perceived risk of a lead-dependent mining strategy and its profitability.

Similar to the trend observed in the block-reward model (for single vs. multiple attackers), in the two-attacker setting the profitability threshold of an individual attacker could be improved, i.e., it is lower than the one in the single-attacker setting. For example, having two attackers running classic selfish mining with $\gamma = 0.5$, the profitability threshold for a single attacker changes from 25% to 22%.

With honest-but-rational miners. We additionally explore the performance of new and existing strategies in the two-attacker setting, but now in the presence of honest-but-rational miners. Our evaluations explore additional 121 strategy combinations. As before, we assume both attackers have equal hashrates, that $\theta_1 = \theta_2 = 0.5$. In addition, in the case that a fork occurs containing two selfish miners' blocks of equal block value and both maximize the honest-but-rational miner utility, it selects a block to mine upon from these two blocks uniformly at random.

Figure 8 left shows our results when $\gamma = 0$, $\kappa = 0.25$, and



$\kappa = 0.25$

	S	P_4	T_1	L	F	$L \circ T_1$	$L \circ T_1 \circ F$	$I_{(0.5,1)}$	$I_{(2,1)}$	$I_{(0.5,1)} \circ L$	$I_{(2,1)} \circ L$
S	0.220	0.230	0.275	0.245	0.250	0.245	0.250	0.230	0.260	0.265	0.305
P_4	0.230	0.225	0.230	NJP	0.285	NJP	NJP	0.245	NJP	NJP	NJP
T_1	0.275	0.230	0.255	0.245	0.255	0.245	0.250	0.230	0.250	0.255	0.280
L	0.245	NJP	0.245	0.225	0.255	0.225	0.250	0.220	0.240	0.235	0.275
F	0.250	0.285	0.255	0.255	0.235	0.250	0.240	0.265	0.300	0.270	0.300
$L \circ T_1$	0.245	NJP	0.245	0.225	0.250	0.225	0.250	0.220	0.230	0.230	0.250
$L \circ T_1 \circ F$	0.250	NJP	0.250	0.250	0.240	0.250	0.220	0.260	0.290	0.265	0.300
$I_{(0.5,1)}$	0.230	0.245	0.230	0.220	0.265	0.220	0.260	0.215	0.240	0.210	0.270
$I_{(2,1)}$	0.260	NJP	0.250	0.240	0.300	0.230	0.290	0.240	0.165	0.220	0.180
$I_{(0.5,1)} \circ L$	0.265	NJP	0.255	0.235	0.270	0.230	0.265	0.210	0.220	0.210	0.240
$I_{(2,1)} \circ L$	0.305	NJP	0.280	0.275	0.300	0.250	0.300	0.270	0.180	0.240	0.170

$\kappa = 0.50$

	S	P_4	T_1	L	F	$L \circ T_1$	$L \circ T_1 \circ F$	$I_{(0.5,1)}$	$I_{(2,1)}$	$I_{(0.5,1)} \circ L$	$I_{(2,1)} \circ L$
S	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP
P_4	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP
T_1	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP
L	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP
F	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP
$L \circ T_1$	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP
$L \circ T_1 \circ F$	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP
$I_{(0.5,1)}$	NJP	NJP	NJP	NJP	NJP	NJP	NJP	0.220	NJP	0.225	NJP
$I_{(2,1)}$	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	0.170	0.235	0.185
$I_{(0.5,1)} \circ L$	NJP	NJP	NJP	NJP	NJP	NJP	NJP	0.225	0.235	0.205	NJP
$I_{(2,1)} \circ L$	NJP	NJP	NJP	NJP	NJP	NJP	NJP	NJP	0.185	NJP	0.165

Fig. 8: Profitability threshold for an individual attacker in the two-attacker setting in the presence of honest-but-rational miners.

$\alpha \in (0, 0.75)$. We observe further reductions to the profitability threshold as compared to the single-attacker setting (Figure 5). For example, when $\gamma = 0$ and $\kappa = 0.25$, the profitability threshold of an individual attacker utilizing classic selfish mining is reduced from 26.5% to 22%.

Result 7: Notably, we find that our class $I_{(f,k)}$ is particularly successful in the two-attacker setting under the transaction-fee regime, with $(I_{(2,1)}, I_{(2,1)})$ achieving a profitability threshold of 16.5% when $\kappa = 0.25, \gamma = 0$—the lowest among this parameter space.

Insight 10: The presence of honest-but-rational miners in the multi-attacker setting lowers the profitability threshold of all examined strategies compared to the multi-attacker setting without honest-but-rational miners.

Figure 8 right shows the joint profitability threshold for $\gamma = 0.0$, $\kappa = 0.5$, and $\alpha \in (0, 0.5)$. With $\kappa = 0.5$, the multi-attacker setting is inherently limited, where in the symmetric setting each attacker can only have at most 25% of the hashrate.

Result 8: We find that except for a small number of combinations—for strategies from the $I_{(f,k)}$ class—most strategy combinations are not jointly profitable.

We justify this behavior as follows. In the single-attacker setting with honest-but-rational miners, we find that strategies tend to be more profitable at $\kappa = 0.25$ than $\kappa = 0.50$ (i.e., they require lower profitability thresholds), and thus we would expect similar behavior in the multi-attacker setting. In addition, when $\kappa = 0.5$, the maximum hashrate for each attacker is only 25%. Comparing this to the setting where $\kappa = 0.25$, we see a maximum hashrate of an attacker to be 0.375, and where a majority of pairs have security threshold above 25%. As a result, as the setting of $\kappa = 0.5$ is expected to be worse than 0.25, and most are already out of the feasible range for such an attacker, they are found to be NJP within this model.

Result 9: For the strategy combinations that are profitable, we find that they again highlight the performance of the $I_{(f,k)}$ class. For example, $(I_{(2,1)}, I_{(2,1)})$ offers an individual profitability threshold for each attacker of 17% (for $\gamma = 0$ and $\kappa = 0.5$). This shows that having honest-but-rational miners promote selfish mining profitability in the multi-attacker setting.

Lastly, we note that the transaction inclusion rule in our model is more restrictive than in the real world; there is not a single view of the mempool and not all transactions are included in a newly mined block. In practice, miners implement various policies for transaction inclusion. Still our evaluations demonstrate how the incentive and threat models promote the impact of selfish mining on blockchain security.

VII. Discussion and Additional Remarks

We conclude with discussions on the implications of selfish mining. In particular, we discuss the connections of transaction fee inclusion to miner extractable value (MEV), as well as fee-driven systems, namely, decentralized (or blockchain-based) resource markets. Furthermore, a study of security attacks would be incomplete without examining countermeasures. Thus, we discuss two relevant topics: detectability of selfish mining and some of the defense mechanisms proposed in the literature. Lastly, we briefly discuss selfish mining studies targeting systems other than Bitcoin.

A. Connections to Incentivized Mining Strategies and MEV

In recent years, a number of works investigated the vulnerability of blockchains to attacks relying on incentivized mining behavior; allowing an attacker to incentivize the inclusion/exclusion or reordering of blocks and transactions. This is highlighted in miner extractable value (MEV), in which a miner can extract additional profits via these actions. Many of the underlying techniques are security attacks themselves, such as front-running, back-running and sandwich attacks usually seen in trading systems, e.g., automated market makers [42].

In the setting of mining this is additionally true; [18] presented a mining strategy known as undercutting which attempts to reorder the head of the chain to introduce additional unnecessary forks.[5] By including less transaction fees in its block, the undercutting miner can incentivize other miners to mine upon this block with the hope of collecting the fees that were left behind in their future blocks. In a way, incentivized selfish mining can be viewed as an MEV technique.

In the context of selfish mining, strategies that attempt to outwardly incentivize other miners have seen little examination. As mentioned previously, beside this work, only three other studies exist [12], [32], [35]. Carlsten et al. [18] have discussed how undercutting may be a useful technique in conjunction with selfish mining (though without presenting a concrete strategy). Thus, we believe that studying the dynamics between transaction fees and the conjunction of selfish mining and MEV is an impactful direction to explore.

B. Connections to Decentralized Resource Markets

In many blockchain-based systems, such as resource markets [43], service fees represent a large part of the earnings. It is customary in these systems that miners also play the role of servers offering services (such as file storage or content distribution) on top of the currency exchange medium [44], [45]. A miner (server) collects service fees given the following condition: the service contracts, and later the transactions that contain proofs of service delivery, are published on the blockchain. Additional revenue could come from resolving disputes and vetting cheating claims against parties in the system, which again need to be published on the blockchain in order to collect their rewards.

Consequently, publishing a block could be controlled by which service contracts and service-related transactions are included. Under such a scenario, we expect that the transaction-fee regime, and the presence of honest-but-rational miners, to be even more impactful. The incentivized selfish mining strategy variant we introduced, and our evaluations, help in drawing insights on the effect of selfish mining in these fee-driven systems and the expected security thresholds for their blockchains. That is, a selfish miner now may not need to publish additional transactions and dispense the fee f, or maybe would need a lower fee value. Instead, it can sway honest-but-rational miners based on the inclusion of their service contracts and service payments, in the selfishly-mined blocks, and thus, encourage them to build on these blocks even if they come from a trailing position compared to the public chain.

C. Detectability of Selfish Mining

Selfish mining strategies can often be detected due to the difference between their behavior and that of honest mining as shown in [46]–[48]. At their core, these works rely on the belief that a high frequency of excluded (orphaned) or stale blocks may be as an indicator of selfish mining. While detection does not represent a direct defense against selfish mining, knowing that selfish miners are present may garner distrust and loss in value to the underlying blockchain; in effect causing incentive value loss for (selfish and honest) miners.

Most prior work on selfish mining detection are theoretical, and until recently no works have examined them in practice. A recent empirical study [49] studied that for Bitcoin, Ethereum, Litecoin, Bitcoin Cash and Monacoin. Notably, a behavior closely resembling behavior of selfish miners has been identified on Monacoin, as well as some degree of abnormal mining behavior among all examined chains.[6]

On the other hand, and as we discussed before, Bahrani et al. [9] presented a stealthy selfish mining strategy to counter detection. In fact, this work presents a framework that enables various selfish mining strategies to become undetectable. This in turn makes reactive mechanisms, that act only when selfish mining is detected, ineffective. As a result, there is a need for proactive defense mechanisms that deter miners from attempting selfish mining in the first place.

D. Defenses Against Selfish Mining

A number of works have developed defenses against selfish mining [51]–[57] (comprehensive surveys can be found in [19], [20]). In this section, we discuss some of these defenses with a focus on the network protocol and miner behavior changes that they present, which impact adoption in practice. We classify these solutions into counter strategies (miner behavior-related) and network changes (encompassing solutions that modify the underlying network protocol).

Counter strategies. Upon detection of a selfish miner, counter-strategy miners adapt their mining behavior to a strategy that reduces the selfish miner's profits. While such strategies do not stop selfish mining, they attempt to penalize the detected selfish miner—hoping to make its behavior unprofitable, so it will go back to mining honestly.

Lee et al. [51] introduced detective mining; it relies on the observation that today most miners are part of large public mining pools that share information between each other. A detective miner can attempt to join such selfish pools to learn what block they are mining upon, subsequently mining upon this block to compete against the pool.

Gal et al. [52] introduced piggybacking, where upon detecting a selfish miner, the piggybacking miners also withhold their blocks for a longer period than that of selfish mining. Maintaining this competition for a long time, a significantly large counter-miner would be expected (with high probability) to build a chain longer than the public chain which has endured

[5] Though an adversarial mining strategy, this would not be classified as selfish mining as blocks are not withheld.

[6] On Monacoin, the time of this abnormal behavior aligns with what is believed to be a selfish mining attack happened between May 13-15, 2018 [50].

waste due to selfish mining. As a result, this chain will replace the selfish one, causing losses for the selfish miner.

Such counter-strategies may limit the profitability of selfish mining, however they come with their own strong assumptions such as presence of a miner with a large hashrate, or the ability to join (and spy on) a selfish mining pool, thus limiting their performance. Also, all they assume that selfish mining can be detected, which makes their effectivity questionable under undetectable selfish mining strategies.

Network changes. A majority of the defense solutions we examine belong to this category [1], [53]–[58]. These works introduce changes to the protocol/mining procedure, and are classified as soft and hard changes based on whether they result in soft or hard forks, respectively.

Soft changes. The first mitigation solution was presented in the first selfish mining work [1]. It modified the fork selection rule; instead of choosing the first block received, a miners chooses one of the received blocks uniformly at random. While this solution may mitigate some of the effects of highly connected selfish miners, it unfortunately further empowers selfish miners with low connectivity. Moreover, it still allows selfish mining to be profitable for many reasonable hashrates. In general, while soft network changes may be easier to adapt, an inherent downside of them is that they are not enforced by the network—miners can continue to use the old software. This limits their impact especially that blockchain participants are usually incentive-driven.

Hard changes. These attempt to solve the problem above and ensure that the majority of the miners adopt the defense mechanism. Early approaches aimed to ensure freshness [53], [57], i.e., blocks are published within a short period after being mined. Such approach shows promise in mitigating selfish mining—though not in its entirety—but introduces additional complexities that may lead to possible security vulnerabilities.

Zhang et al. [54] proposed having miners publish intermediate blocks (valid blocks that meet an easier difficulty target) to be then used as the new head for the mining of subsequent blocks (either normal blocks or intermediate ones). These blocks do not provide a reward or bring transaction on-chain, but are only used to minimize the time span between blocks—diminishing the window during which selfish mining can occur. Another work [55] utilized weighted fork resolving policy—in a similar way to the inclusion of uncle blocks in proof-of-work Ethereum. This allows resolving forks in a way that potentially favors honest forks over selfish ones that tend to have smaller weight and not-so-fresh blocks.

Others resorted to designing new consensus protocols, e.g., Fruitchain [58] that aims to enforce reward fairness so that miners obtain rewards in proportion to their hashrate. While [13] advocated for including valid stale/orphaned blocks when adjusting the difficulty to avoid producing easier difficulty targets that may benefit selfish miners. However, [14] showed that even under this modified DAA algorithm, several selfish mining attacks may still be profitable.

Finally, when including transaction fees, Xiongfei et al. [56] proposed limiting fee volatility (as it has been found impactful for improved selfish mining and undercutting [18]). This is done by capping the number of transactions with high fees that may go into a block (and enforcing that as part of the mining protocol). Nonetheless, such technique requires configuring several parameters, like what constitutes a high fee, and whether the cap would change as the transaction fee distribution changes over the years, etc. So it adds to the complexity of the consensus/mining protocol.

E. Studying Selfish Mining in Other Blockchains

It is observed that most selfish mining works targeted Bitcoin; which is natural given how Nakamoto-style consensus works and the popularity of Bitcoin. While not to the same extent, selfish mining has been examined across other systems finding numerous profitable selfish mining strategies. For Ethereum, selfish mining has been studied for both its older proof-of-work version, e.g., [21], [22], [59]–[61] (contextualizing the effect of uncle rewards on classic selfish mining, stubborn mining, and bribery selfish mining), and the more recent proof-of-stake (PoS) version [23] (formulating a new strategy class specific to its network protocol). Selfish mining strategies have additionally been studied for other protocols, including the longest chain PoS protocols [62], such as the Emmy protocol in Tezoz [63] and Ouroboros protocol in Cardano [64], [65]. For Filecoin, [66] examined three selfish mining strategies against Filecoin's consensus layer [67]. These works represent a high level adaptation to "selfish proposing" attacks—in systems where new blocks are proposed rather than mined.

While relying on often specific protocol features to formulate an attack, still each of these works utilizes temporary-block withholding. Nonetheless, their modeling settings and evaluation results vary based on the target protocol specifications. Keller et al. [68] have attempted to bridge this gap by formulating a generic Markov decision process for selfish mining attacks on DAG-based protocols. Although they mention that this generic model could be adopted for other protocols, they show results only for Bitcoin. Thus, more work is still needed to understand selfish mining at a generic level.

VIII. Conclusion

We presented a systematization framework categorizing existing selfish mining attack works in Bitcoin according to their strategy and model formulations. In doing so, we unravel the often entangled developments of selfish mining attacks—often fragmented across varying modeling environments. To further contextualize the landscape, we evaluated existing and new strategy variants in the transaction-fee regime, across both the single and multi-attacker setting, showing new security thresholds and relationships. Finally, we discuss detectability of selfish mining and prior works on the defense side, as well as connections to MEV, fee-driven blockchain-based systems, and selfish mining in systems other than Bitcoin.

Acknowledgments

The work of C.F. and G.A. is supported by NSF under Grant No. CNS-2226932.

References

[1] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International Conference on Financial Cryptography and Data Security*, 2014.

[2] H. Liu, N. Ruan, R. Du, and W. Jia, "On the strategy and behavior of bitcoin mining with n-attackers," in *Asia Conference on Computer and Communications Security (AsiaCCS)*, 2018.

[3] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.

[4] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security*, 2017.

[5] Q. Bai, Y. Xu, N. Liu, and X. Wang, "Blockchain mining with multiple selfish miners," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3116–3131, 2023.

[6] Q. Xia, W. Dou, T. Xi, J. Zeng, F. Zhang, J. Wei, and G. Liang, "The impact analysis of multiple miners and propagation delay on selfish mining," in *IEEE Annual Computers, Software, and Applications Conference (COMPSAC)*, 2021.

[7] C. Schwarz-Schilling, S.-N. Li, and C. J. Tessone, "Stochastic modelling of selfish mining in proof-of-work protocols," *Journal of Cybersecurity and Privacy*, vol. 2, no. 2, pp. 292–310, 2022.

[8] S. Zhang, K. Zhang, and B. Kemme, "Analysing the benefit of selfish mining with multiple players," in *IEEE International Conference on Blockchain (Blockchain)*, 2020.

[9] M. Bahrani and S. M. Weinberg, "Undetectable selfish mining," in *ACM Conference on Economics and Computation*, 2024.

[10] F. J. Marmolejo-Cossío, E. Brigham, B. Sela, and J. Katz, "Competing (semi-) selfish miners in bitcoin," in *ACM Conference on Advances in Financial Technologies*, 2019.

[11] R. Bar-Zur, A. Abu-Hanna, I. Eyal, and A. Tamar, "Werlman: To tackle whale (transactions), go deep (rl)," in *IEEE Symposium on Security and Privacy (SP)*, 2023.

[12] J. Yu, S. Gao, R. Song, Z. Cai, and B. Xiao, "Partial selfish mining for more profits," *arXiv preprint arXiv:2207.13478*, 2022.

[13] C. Grunspan and R. Pérez-Marco, "On profitability of selfish mining," *arXiv preprint arXiv:1805.08281*, 2018.

[14] R. Sarenche, R. Zhang, S. Nikova, and B. Preneel, "Time-averaged analysis of selfish mining in bitcoin," *Cryptology ePrint Archive*, 2024.

[15] C. Hou, M. Zhou, Y. Ji, P. Daian, F. Tramèr, G. Fanti, and A. Juels, "Squirrl: Automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning," in *Network and Distributed Systems Security (NDSS) Symposium*, 2021.

[16] J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," *Performance evaluation*, vol. 104, pp. 23–41, 2016.

[17] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong, "A deep dive into blockchain selfish mining," in *IEEE international conference on communications (ICC)*, 2019.

[18] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *ACM SIGSAC conference on computer and communications security (CCS)*, 2016.

[19] K. Nicolas, Y. Wang, and G. C. Giakos, "Comprehensive overview of selfish mining and double spending attack countermeasures," in *IEEE Sarnoff Symposium*, 2019.

[20] N. Madhushanie, S. Vidanagamachchi, and N. Arachchilage, "Selfish mining attack in blockchain: a systematic literature review," *International Journal of Information Security*, pp. 1–19, 2024.

[21] C. Feng and J. Niu, "Selfish mining in ethereum," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2019.

[22] F. Ritz and A. Zugenmaier, "The impact of uncle rewards on selfish mining in ethereum," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2018.

[23] J. Neu, E. N. Tas, and D. Tse, "Two more attacks on proof-of-stake ghost/ethereum," in *ACM Workshop on Developments in Consensus*, 2022.

[24] B. Jebari, K. Ibrahimi, M. Jouhari, and M. Ghogho, "Analysis of blockchain selfish mining: a stochastic game approach," in *IEEE International Conference on Communications*, 2022.

[25] C. Grunspan and R. Pérez-Marco, "Selfish mining and dyck words in bitcoin and ethereum networks," *Blockchain Economics, Security and Protocols*, 2019.

[26] Y. Zhang, M. Liu, J. Guo, Z. Wang, Y. Wang, T. Liang, and S. K. Singh, "Optimal revenue analysis of the stubborn mining based on markov decision process," in *International Conference on Machine Learning for Cyber Security*, 2022.

[27] I. G. A. K. Gemeliarana and R. F. Sari, "Evaluation of proof of work (pow) blockchains security network on selfish mining," in *International seminar on research of information technology and intelligent systems (ISRITI)*, 2018.

[28] X. Dong, F. Wu, A. Faree, D. Guo, Y. Shen, and J. Ma, "Selfholding: A combined attack model using selfish mining with block withholding attack," *Computers & Security*, vol. 87, 2019.

[29] X. Dong and S. Gao, "Genselfholding: fusing selfish mining and block withholding attacks on bitcoin revisited," *Journal of Networking and Network Applications*, vol. 2, no. 1, pp. 23–35, 2022.

[30] Q. Wang, T. Xia, D. Wang, Y. Ren, G. Miao, and K.-K. R. Choo, "Sdos: Selfish mining-based denial-of-service attack," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3335–3349, 2022.

[31] J. Shang, T. Lu, and P. Zhao, "Sim: Achieving high profit through integration of selfish strategy into innocent mining," *IEEE Transactions on Network and Service Management*, 2024.

[32] S. Gao, Z. Li, Z. Peng, and B. Xiao, "Power adjusting and bribery racing: Novel mining attacks in the bitcoin system," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

[33] K. A. Negy, P. R. Rizun, and E. G. Sirer, "Selfish mining re-examined," in *Financial Cryptography and Data Security*, 2020.

[34] S. G. Motlagh, J. Mišić, and V. B. Mišić, "The impact of selfish mining on bitcoin network performance," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 724–735, 2021.

[35] W. Liang, S. Jiang, W. Li, and Y. Wang, "Leading hide bribery stubborn mining attack," in *ACM International Symposium on Blockchain and Secure Critical Infrastructure*, 2024.

[36] S.-W. Wang and S.-S. Tzeng, "An accurate analytical model for a proof-of-work blockchain with multiple selfish miners," in *IEEE International Conference on Communications*, 2024.

[37] S.-W. Wang, "Analysis of earned rewards in a blockchain with two selfish miners," in *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2024.

[38] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.

[39] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 10 576–10 597, 2022.

[40] K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in *Financial Cryptography and Data Security Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA*, 2017.

[41] "Our source code," https://github.com/CSSL-UConn/mining-simulator.

[42] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *IEEE Symposium on Security and Privacy (SP)*, 2020.

[43] G. Almashaqbeh, "Rethinking service systems: A path towards secure and equitable resource markets," *USENIX ;login: Magazine*, 2021.

[44] "Filecoin," https://filecoin.io/.

[45] "Livepeer," https://livepeer.com/.

[46] Z. Wang, Q. Lv, Z. Lu, Y. Wang, and S. Yue, "Forkdec: accurate detection for selfish mining attacks," *Security and Communication Networks*, vol. 2021, no. 1, 2021.

[47] V. Chicarino, C. Albuquerque, E. Jesus, and A. Rocha, "On the detection of selfish mining and stalker attacks in blockchain networks," *Annals of Telecommunications*, vol. 75, pp. 143–152, 2020.

[48] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in *International Conference on Computing, Networking and Communications (ICNC)*, 2019.

[49] S.-N. Li, C. Campajola, and C. J. Tessone, "Statistical detection of selfish mining in proof-of-work blockchain systems," *Scientific Reports*, vol. 14, no. 1, p. 6251, 2024.

[50] D. Gutteridge, "Japanese cryptocurrency monacoin hit by selfish mining attack," May 2018. [Online]. Available: https://finance.yahoo.com/news/japanese-cryptocurrency-monacoin-hit-selfish-205031219.html?

[51] S. Lee and S. Kim, "Detective mining: Selfish mining becomes unrealistic under mining pool environment," *Cryptology ePrint Archive*, 2019.

[52] J. Gal and M. B. Szabo, "Majority is not needed: A counterstrategy to selfish mining," *arXiv preprint arXiv:2304.06313*, 2023.

[53] E. Heilman, “One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner,” in *Financial Cryptography and Data Security Workshops, BITCOIN and WAHC*, 2014.

[54] R. Zhang and B. Preneel, “Broadcasting intermediate blocks as a defense mechanism against selfish-mine in bitcoin,” *Cryptology ePrint Archive*, 2015.

[55] ——, “Publish or perish: A backward-compatible defense against selfish mining in bitcoin,” in *Topics in Cryptology–CT-RSA*, 2017.

[56] X. Zhao and Y.-W. Si, “Dynamic transaction storage strategies for a sustainable blockchain,” in *IEEE International Conference on Services Computing (SCC)*, 2021.

[57] S. Solat and M. Potop-Butucaru, “Brief announcement: Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin,” in *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, 2017.

[58] R. Pass and E. Shi, “Fruitchains: A fair blockchain,” in *ACM symposium on principles of distributed computing*, 2017.

[59] Y. Liu, Y. Hei, T. Xu, and J. Liu, “An evaluation of uncle block mechanism effect on ethereum selfish and stubborn mining combined with an eclipse attack,” *IEEE Access*, vol. 8, pp. 17 489–17 499, 2020.

[60] C. Feng and J. Niu, “Selfish mining in ethereum,” in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1306–1316.

[61] Y. Wang, Z. Wang, M. Zhao, X. Han, H. Zhou, X. Wang, and A. S. V. Koe, “Bsm-ether: Bribery selfish mining in blockchain-based healthcare systems,” *Information Sciences*, vol. 601, pp. 1–17, 2022.

[62] R. Sarenche, S. Nikova, and B. Preneel, “Deep selfish proposing in longest-chain proof-of-stake protocols,” in *Financial Cryptography and Data Security*, 2024.

[63] L. M. Goodman, “Tezos—a self-amending crypto-ledger white paper,” 2014. [Online]. Available: https://www.tezos.com/static/papers/white_paper.pdf

[64] A. Kiayias, A. Russell, B. David, and R. Oliynykov, “Ouroboros: A provably secure proof-of-stake blockchain protocol,” in *Annual international cryptology conference*. Springer, 2017, pp. 357–388.

[65] B. David, P. Gaži, A. Kiayias, and A. Russell, “Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain,” in *Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part II 37*. Springer, 2018, pp. 66–98.

[66] T. Cao and X. Li, “Temporary block withholding attacks on filecoin’s expected consensus,” in *International Symposium on Research in Attacks, Intrusions and Defenses*, 2023.

[67] “Filecoin, expected consensus,” https://spec.filecoin.io/algorithms/expected_consensus/.

[68] P. Keller and G. Bissias, “Generic selfish mining mdp for dag protocols,” *arXiv preprint arXiv:2309.11924*, 2023.