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 when mining honestly. 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 with conclusions that are often exclusive to certain model assumptions. This growing complexity makes it hard to comprehend the state of the art and distill its impact and trade-offs.

In this paper, we demystify the landscape of selfish mining by presenting a systematization framework of existing studies and evaluating their strategies under 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 clarifying open questions and understudied areas. Among them, we find that most studies target the block-reward setting and do not account for transaction fees, and generally study the single attacker case. To bridge this gap, we evaluate many of the existing strategies in the no-block-reward setting-so miner's incentives come solely from transaction fees-for both the single and multi-attacker scenarios. We also extend their models to include a realistic honest-butrational miners showing how such adaptations could garner moreperformant 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 and effectively control the blockchain.

Selfish mining is a temporary block-withholding attack. It aims to waste other miners' computation by withholding and selectively publishing locally-mined blocks to the public chain, and hence, devoiding the work of other miners as their blocks get abandoned. Selfish mining strategies vary based on the exact conditions under which a selfish miner withholds/publishes blocks. In general, these strategies exploit the longest-chain rule of Nakamoto-style blockchains and its fork selection process. Selfish mining presents a clear threat to blockchain security; it enables selfish 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 [1]–[20]. Alongside devising new strategy variants that are more performant (i.e., profitable) than classical ones [1], these works also examine various model settings and parameterizations, such as accounting for broadcast latency [7], [8], [21] and varying the number of attackers [3], [6], [9], [11], [22], while others define optimal strategies for particular settings [5], [6]. Moreover, new strategy families have been formulated based on more granular attacker behavior with respect to block withholding and publishing [3], [4], [10], [15]. Finally, a few works have studied selfish mining for systems other than Bitcoin, including Ethereum [18], [19], [23], Filecoin [17], and Cardano [15].

This expanding landscape makes it hard to understand the impact of selfish mining strategies on blockchain security and their effectiveness. These strategies are analyzed across diverse model settings spanning multiple blockchain specifications, number of attackers, and incentive formulations. At the same time, a careful inspection reveals that most of existing works focused on particular paradigms, mainly in the block reward model for single attackers, thus fracturing the results to be largely model-specific. This in effect complicates the process of assessing the current state of the art. Due to their impact on blockchain security, 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 studies. 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. On the one hand, our work makes it easier to understand the landscape of selfish mining and clearly distill the impact on blockchain security. On the second hand, it bridges many of the gaps in this landscape by studying new strategy variants under model adaptations that existing works did not cover. In particular, we make the following contributions.

Systematization framework. We develop a two-category systematization framework for prior selfish mining works 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 various miner behaviors or threat models. We believe that our framework offers a versatile and holistic approach for the evaluation of (existing and future) selfish mining works, enabling a clear path to understanding them and distilling security impacts and trade-offs.

Analyzing existing works. Leveraging our systematization framework, we analyze and categorize 21 selfish mining attack works. Our findings show that 13 works include new strategy formulations, where naturally most of them focus on maximizing profitability. Whereas, in terms of model settings, we find that these works vary based on the number of attackers included—single or multi attackers, generally follow three incentive models—whether blocks rewards and transaction fees are covered, and many of them account for the network configuration impact on the attack success. However, a few of these works considered rationality of honest miners beside the regular selfish vs. honest miner threat model.

Distilling insights and identifying gaps. Our analysis highlights a number of overarching insights and calls attention to a number of open questions. For example, in context to model formulation, we find that only 3 of the 21 works include transaction-fees in their modeling; 2 works consider both block rewards and transaction fees, and only 1 study examines the no block-reward model (i.e., incentives come only from transaction fees, known as the transaction-fee regime). In a similar vein, we find that only 7 works consider the multi-attacker setting—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, and aid in closing some of the gaps, 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. In particular, utilizing a mining simulator, we implement and evaluate 11 existing strategies across several different models, including the single and multi-attacker setting (to our knowledge, this is the first study of the multi-attacker setting in the transaction-fee regime). Our assessment finds new (lower) profitability thresholds for many existing strategies, and highlights the dominant (most profitable) strategies among regions of the parameter space.

Furthermore, we extend the selfish mining model to include honest-but-rational miners. In doing so, we find a new mutually-beneficial relationship between selfish miners and honest-but-rational ones during the fork selection process. Specifically, this may facilitate existing selfish mining strategies to become more profitable while providing additional incentives to honest-but-rational miners in the form of expected future revenue from transactions fees. In addition, we show how this new environment can foster strategy variants that can further benefit from this newfound relationship. For example, the 33%

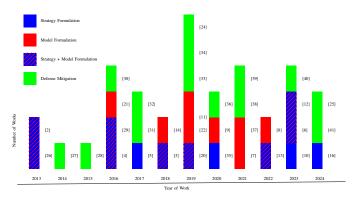


Fig. 1: Timeline of selfish mining attacks and defenses.

profitability threshold of classic selfish mining is reduced to 26.5% (under 25% honest-but-rational miners). Moreover, we study strategy composition in which attackers follow multiple strategies, and show further security threshold reduction.

Discussion and additional remarks. Showing security attacks is important as it corrects any misconceptions on the security state of systems and current practices. It also serves as a motivation for developing proper security countermeasures and defenses. Thus, we conclude with examining whether selfish mining can be detected when it takes place, and the current state 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 systems 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. On the side of defenses, [24], [25] provided surveys of selfish mining countermeasures studying their features and limitations. In contrast, we focus on the attack side of the selfish mining landscape, categorizing these attacks according to the underlying characteristics that define their strategies and distinct model settings under which they are examined. Thus, we view [24], [25] as complementary to our work; combined they offer a holistic view of selfish mining attacks and defenses.

II. SYSTEMATIZATION FRAMEWORK

We develop a systematization framework corresponding to the features of the examined selfish mining attacks. Generally, our framework covers two dimensions including the attack strategies, and the models/settings under which these strategies have been studied. The timeline, shown in Figure 1, traces the evolution of these works across these dimensions (to provide a holistic view, the figure also traces the evolution of defenses to show the contrast between the attack and defense arena).

Strategy Formulation. This encompasses the conceptualization of selfish mining strategies characterized by their behaviors and actions, such as when and what block to mine on and when to reveal the withheld blocks. Examining 13 selfish mining

TABLE I: Categorization of selfish mining strategy formulation.

Category	Dimension	Works												
		[2]	[10]	[6]	[12]	[29]	[1]	[3]	[4]	[35]	[5]	[16]	[13]	[20]
Profitability	Profit maximizing	✓	\checkmark		\checkmark		\checkmark	\checkmark						
	Time-to-profit minimzing									✓		✓		
Update	Lead dependent	✓					✓	✓	✓				✓	
Criteria	Context aware		✓	\checkmark	✓	✓				✓	\checkmark	✓		✓

works that formulate selfish mining strategies, we observe that these strategies can be split into two categories: (1) the *notions* of profitability and (2) the update criteria, i.e., what triggers an action. For profitability, we find that the studied attacks can be split based on two goal: the natural profit-maximizing goal in terms of collecting more 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 can be split 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 that rely on additional context either from the blockchain view or the underlying protocol specifications to trigger actions (e.g., block value or mining difficulty).

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 of 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 basically 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). The seminal 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 key 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 mining honestly. While maximizing profitability is a typical goal [1], [2], [4]–[6], [6], [13], [17]–[20], [23], [29], Grunspan et. al [14] showed that in practice the time it takes strategies to become profitable is quite long—on the order of weeks. Accordingly, we adopt these two profitability-related dimensions.

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. Classic selfish mining [1] falls under this category, which is (in the worst case) profitable with 33% of the total network hashrate. Below we compare and contrast this to the development of other strategy classes, discussing their techniques and profitability thresholds.²

Insight 1: Following the footsteps of classic selfish mining, and as shown in Table I, the profit-maximizing category includes the majority of formulated selfish mining strategies so far.

Classic selfish mining. At its core, classic selfish mining 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 on top of the public chain head, this lead becomes 0, and so on. A classic selfish miner chooses what block to mine on and when to reveal its private block according to its current lead. Its actions can be characterized as follows:

- Having a lead > 2: Here, a selfish miner will continue to mine on its private chain. If a new block is published to the top of the public chain, the selfish miner will choose to reveal its oldest block (thus, it creates a fork).
- Having a lead of 2: If a new block is published on the public chain, the selfish miner will reveal its entire chain (resulting in a longer public chain).
- Having a lead of 1: If a new block is published on the public chain, the selfish miner will reveal its entire chain (resulting in 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 will immediately reveal this new block. Instead, if the opposing block is mined upon first (resulting in a negative lead for the selfish miner) the selfish miner will scrap its chain, starting again to mine on top of the public chain's head.

¹The rest of the works do not develop new strategies, but study existing ones under different models.

²Profitability 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.

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

Insight 2: There is a natural inverse relationship between the connectivity rate, γ , of a selfish miner and the network security threshold. Higher γ allows the selfish miner to profit at a lower hashrate, which lowers the network security threshold.

Stubborn selfish mining. While classic selfish mining is already profitable, subsequent profit-maximizing strategies have attempted to further decrease the profitability threshold via action adaptations. Among them, stubborn selfish mining [4] represents strategies in which selfish miners continue to mine 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 in 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 will discard its private chain and start over.
- Stubborn lead mining L: Upon a lead ≥ 0 , when a new block is published on the public chain, the selfish miner will publish the oldest block of its private chain. 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 will withhold publishing this block (resulting in a lead of 1) and continues to selfishly mine.

These strategies can be further combined to present even more nuanced risk scenarios. In the generic model (no-latency, block-reward only, single-attacker), 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 was found to be the most profitable (among stubborn mining strategies) outperforming classic selfish mining threshold by 1.4%, while for $\gamma=1$, it is found that the combination of stubborn lead and stubborn-1-trail is most profitable being up to 25% more profitable than classic selfish mining.

Insight 3: Based on [4], neither classic or stubborn selfish mining was dominantly profitable across the entire parameter space. Instead, the most profitable strategy was dependent on the choice of parameters.

<u>Publish-N selfish mining.</u> In the setting of multi non-colluding selfish miners, the publish-N strategy class [3] 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 in the sense that it allows a classic selfish miner to have a lead up to N-1, and upon reaching a lead of N blocks, the oldest block of its private chain is published.

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 [3] 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.

Insight 4: The choice of N in publish-N denotes the risk-aversion compared to classic selfish mining, where N=1 is equivalent to honest mining, and $N=\infty$ is equivalent to classic selfish mining.

Partial selfish mining. Assuming the presence of rational miners, who choose to mine where the most profit is promised, fosters new strategies. Specifically, this allows selfish miners collude with rational miners. In this light, Yu et al. [13] presented partial selfish mining, a strategy class that incentivizes rational miners to help the selfish miner in extending its private-chain prior to its publication.

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 with each other to avoid wasting each other's work). In effect, this mimics classic selfish mining under a larger hashrate.

To achieve this, [13] employs zero-knowledge proofs to prove the existence of a valid private chain. Furthermore, the selfish miner deploys a collateralized smart contract to ensure that rational miners will act fairly, i.e., rewarding them for producing valid blocks that extend the private chain. Their findings show that this strategy can be mutually beneficial for both rational miners and selfish miners with both outperforming honest miners. Notably, assuming 50% of the miners to be rational, a partial selfish mining strategy is more profitable than honest mining at a 20% hashrate.

Insight 5: Collusion with rational miners establishes a selfish mining pool, thus empowering the selfish miner's hashrate and improves its profitability.

Improved selfish mining. In the setting of no block-reward (i.e., only transaction fees), Carlsten at. al [29] introduced the improved selfish mining class. 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 (i.e., 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 due to the act that as attacker's hashrate lowers, the cutoff value approaches 0, and thus the improved selfish miner will mine honestly most of the time. It was shown that when $\gamma=0$, where the security threshold of classic selfish mining is 33% (as also found in the no-block-reward model [29]), improved selfish mining succeeds in mining 38% of the blocks adopted by the public chain.

WeRLman strategies. A recent work [12] introduced WeRL-Man, a framework for formalizing strategies that utilizes deep reinforcement learning under block-rewards and volatile transaction fees. By exploring the strategy space, i.e., determines what action to take place based on parameters such as its hashrate and knowledge of the chain, WeRLMan found that semi-frequent bumps 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 6: 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.

<u>Undetectable selfish mining.</u> Due to the difference in behavior from honest mining, selfish mining could be detectable. The class of undetectable selfish mining strategies [10] trades off profitability for statistical undetectability. Since a key indicator of detection is how wasted blocks are produced, the selfish miner tries to mimic the behavior of honest miners. That is, it relies on a careful balance between revealing additional blocks while ensuring that not too many blocks on the public chain would be wasted in succession, rather than just depending the lead a selfish miner has. These strategies are found to be statistically undetectable at the expense of their worst case security threshold, which rises to 38.2%.

Insight 7: In contrast to stubborn selfish mining which can be portrayed as a risky variation of classic selfish mining, undetectable selfish mining can be viewed as a more risk-averse version. Its behavior is more likely to pass as honest under increased network latency.

Optimal selfish mining. Instead of developing a new 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 history of the chain and specific parameters such as the selfish miner's hashrate (and in the case of multi-attackers, any available information about these attackers—their number, hashrates, and strategies). In particular, [5] formulated selfish mining as a single-player decision problem, and utilized a numerical solver to solve for the optimal actions for specific parameters, i.e., connectivity rate and selfish miner hashrate.

The findings of [5] show a marginally lower worst-case profitability threshold compared to classic selfish mining, and higher profitability when exceeding this threshold. Interestingly, the profitability results yielded from this approach closely mirror the results found via the combination of stubborn mining strategies in [4], being at most 1.4% more profitable than their stubborn mining strategy counterparts. Similar results were additionally found by a subsequent work [20], 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, the optimal strategies formulated in the generic model do not translate directly. The multi-attacker case was first explored in [11] 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 attackers 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 [6] removed this assumption, adapting the formulation of [5] 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. The key reason is that while a selfish miner attempts to waste other miners' efforts by excluding their valid blocks, it also risks the short term profit by the possibility that the selfishly-mined blocks may be excluded as well. However, in the long term, these exclusions of blocks lowers the difficulty threshold. That is, in Bitcoin the difficulty of block production is adjusted every 2016 blocks (roughly two weeks), as less blocks are produced, the difficulty adjustment algorithm will produce a lower difficulty target than

what is expected to be. As a result, a selfish miner may earn more during this, lower difficulty, next period.

Intermittent selfish mining. Intermittent selfish mining [35] utilizes a variable approach swapping 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 two difficulty adjustment periods. Compared to profit-maximizing strategies, intermittent selfish mining is less profitable (though in a shorter time-horizon) than profit-maximizing strategies. 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 [16] extrapolated this space under various difficulty adjustment algorithms (DDA) (including those that factor in excluded blocks within the difficult adjustment). These variants, known as *smart intermittent mining*, not only swaps strategies (selfish and honest) between difficulty adjustment periods, but also does that within a given period. Examining smart intermittent selfish mining—where classic selfish mining is alternated with honest selfish mining 50% within a given period—under Bitcoin's original DDA, it is found that this strategy is profitable given a hashrate of 27% and $\gamma=0.5$ in approximately 11 weeks compared to the 14 weeks necessary for optimal selfish mining under the same parameters.

In addition, when evaluating these strategies under a DDA that factors in excluded blocks, it was found that they are profitable for a hashrate of around 25% when $\gamma=0.5$. This refutes a previous claim [14] that the inclusion of stale blocks within a DDA makes selfish mining attacks unprofitable.

Gap 2: There are no works on strategies that attempt to maximize profitability when still minimizing the time-to-profitability. Optimizing across these two dimensions may foster new variants with reduced security thresholds.

Gap 3: The performance of strategy classes, 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 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, we find the update criteria can largely be categorized into two classes: (1) *lead-dependent* relying just on the relative lead of the private chain (2) *context-aware* relying on additional context from the blockchain or its protocol.

Lead-dependent strategies. Lead-dependent 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, 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 [4] that this strategy is more profitable than classic selfish mining, e.g., for an attacker with hashrate above 33%, it outperform 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, to trigger their actions. This category includes intermittent, improved, and optimal selfish mining, as well as the WeRLman's strategies.

Insight 8: All attempts to formulate optimally-profitable strategies adopted the context-aware update criteria.

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), the choice of parameters (hashrate and connectivity rate), and the history of past strategy actions and its impact on profitability seen so far. Context-aware strategies add an opportunistic nature to selfish mining in the sense of utilizing any profit-improving opportunities.

Insight 9: Context-aware update criteria allows for for a more adaptive selfish mining than lead-dependent. It allows 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 ar. While impactful in advancing the understanding of 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 value.

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

TABLE II: Selfish mining works by number of attackers.

# of Attackers	Profitability Dimension								
	Profit	Time-to-Profit							
	Maximizing	Minimizing							
1	[1], [2], [4], [5], [8], [10], [12]–[14], [20], [21], [29], [37]	[16], [35]							
2	[3], [6], [7], [9], [11], [20], [22]								
3+	[3], [6], [7], [9], [11], [20]								

formulation (based on the considered additional factors 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 miner behaviors, other than honest, in addition to the selfish 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 these multiple attackers are basically competing against 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).

In Table II we highlight the number of attackers examined in prior works according to our two dimensions of strategy profitability. Existing works include evaluations of strategies originally examined in the generic model, such as classic and stubborn selfish mining.³ In addition, they include the formalization of new strategies stemmed from having multiple attackers, such as publish-N and optimal selfish mining.

Insight 10: The majority (16 of 21) of selfish mining attacks has been examined only for single attackers.

For strategies that have previously been examined within another context, evaluations within the multi-attacker setting further contextualize their performance and its relationship to the model setting. For classic selfish mining, multiple works [7], [9], [22] explored that. Bai et al. [22] found that multiple attackers lower the profitability threshold necessary for an individual attacker. Specifically, when $\gamma=0.5$, each attacker only needs a hashrate of 21.48% to be profitable (compared to 25% in the single attacker setting).

To this effect, 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, choose to switch to honest mining. This will alter the profitability of the once profitable miner (to the point where it may not be profitable anymore), as its profitability may have relied on having another party who is also selfishly mining. In other words, profitability in the multi-attacker setting could be increasingly volatile and highly dependent on the joint profitability of all selfish miners. Zhang et al. [9] confirmed this relation. 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 among all attackers.

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

Finally, as in the single-attacker setting, there have been questions regarding the optimal profit-maximizing strategy under a certain number of attacker. This was first formulated under a simplified model by [6] and later found by [6] an optimal policy when competing against classic selfish miners. The deep reinforcement learning based approach [20] affirmed that the optimal strategy present in [5] 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.

Gap 5: Optimal selfish mining for the multi-attacker setting are context-aware relying on the number and strategies of opposing attackers—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 we find in the evaluated selfish mining works. In Table III we highlight this gap, systematizing works across the various model types that include transaction fees. Notably, only three works [12], [14], [29] considered transaction fees.

In [29], block rewards were replaced with only transaction fees, i.e., transaction-fee regime, to evaluate classic selfish mining against improved selfish mining (the latter 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, [12] examines possibles strategies while assuming that these fees are volatile with infrequent jumps in block value. Their findings highlight a relation between expanding significance of transaction-fees (i.e., as transaction-fees make up more and more of the total rewards) and blockchain insecurity, with security continuing to degrade as block rewards get smaller (due to Bitcoin block reward inflation policy).

On the other hand, [14] first utilizes transaction fees in its analysis of the time-to-profitability of classic selfish mining

³Such examinations are limited [3]; only compare stubborn-trail mining to classic and publish-N classes.

TABLE III: Classification of selfish mining strategies by model setting.

Strategies / Model Setting	Block-reward	Model	Block Reward +	Tx Fees Model	No-Block-Reward Model		
	Single Attacker	Multi-Attacker	Single Attacker	Multi-Attacker	Single Attacker	Multi-Attacker	
Classic selfish mining	[1], [2], [4], [7], [8], [21], [37]	[3], [7], [9], [11], [22]	[14]		[29], This work	This work	
Stubborn selfish mining	[4]	[3]	1		This work	This work	
Publish-N		[3]	1		This work	This work	
Other strategies	[5], [10], [13], [16], [20], [35]	[6], [20]	[12]		This work	This work	

(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 (whether profitability or time to become profitable) on many existing selfish mining strategies. Also, optimal selfish mining in the transaction-fee regime is an open question.

C. Network Configuration

Most prior works studied a simplified model of Bitcoin network, namely, abstracted away propagation delay byt assuming no latency. However, latency can play a crucial role in the performance of selfish mining strategies as shown by a select number of works [5], [7], [8], [21].

Varying approaches such as simulations [7], stochastic [8] modeling and analysis [5], [21], [37] have been used in such evaluations. But they have largely been limited to classic selfish mining, except for [5] which developed an optimal solution. Interestingly, results show that propagation delay can in fact both make selfish mining attacks more profitable, and lower the security threshold (i.e., profitability threshold) for a selfish mining strategy. While most of these works focus on the singleattacker setting this was additionally found true in [7] that examines classic selfish mining under latency in the multiattacker setting. The key idea behind these findings is that non-zero propagation delay provides additional time for selfish miners to selfishly mine, further benefiting their strategies. Finally, while not additionally examining its effects on selfish mining, undetectable selfish mining utilizes the behavior of honest miners under non-zero latency to prevent detection.

Insight 11: Non-zero propagation delay has a direct effect on a selfish mining strategy's profitability, providing additional time for a selfish miner to selfishly mine.

Gap 8: The impact of propagation delay has almost exclusively examined classic selfish mining in the single-attacker setting. Studying this impact for other strategies and model settings is an open question.

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 opposing honest miners or, in the multi-attacker setting, opposing selfish miners in terms of the strategies they are using.

Insight 12: 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 formulations.

While a critical component, explorations of variant mining behavior are largely limited with respect to their threat model. In the single attacker setting, all works (except [13]) assumes 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. This may include rational miners, i.e., miners who are willing to selectively selfishly mine, and honest-but-rational miners, i.e., those who are willing to select the most profitable branch (for them) of the public chain. The inclusion of rational miners has been explored in [13] (block-reward model, no transaction fees); it showed indeed that this promotes profitability as mentioned before.

Gap 9: None of the existing works have studied the inclusion of honest-but-rational miners.

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 by manipulating the block value [29], [42]. Though it is yet to be validated for selfish mining attacks.

Gap 10: How transaction-fees can be utilized to sway rational and honest-but-rational miners in the selfish mining is an open question.

In the multi-attacker setting, the threat model additionally includes the behavior of the opposing attacking miners in terms of their strategies. This impact 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 has confirmed this relation.

Gap 11: Further studies of the impact of miner rationality in the multi-attacker setting are 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. The few works that accounted for transaction fees confirmed that this inclusion influences the profitability of selfish mining and can foster new strategies. Studying selfish mining in the transaction-fee regime, where miner's revenue comes solely from transaction fees, received much less attention (only one work [29] did that). This is a result of viewing transaction fees as negligible compared to block mining 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. So it is important to study this regime and understand its impact on blockchain security.

As discussed previously, [12], [29] studied the impact of transaction fees (with and without block rewards), but only in the single-attacker model. At the same time, considering multiple attackers has been limited to the block reward model [3], [6], [7], [9], [11], [22]. So a natural question is how these results would change when accounting for 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 the single and multi-attacker setting. The goal is to provide a more inclusive view of selfish mining impact to complement our systematization of knowledge study. In this section, we contextualize the system model of this regime 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 an 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 [29], 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 there is no backlog of transactions, so a newly mined block will empty the queue. Thus the block value of this block is the sum of transaction fees per round multiplied by the time (in rounds) since the 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*: Miners who strictly follow the protocol, and immediately publish newly mined blocks.
- Honest-but-rational miners: Miners who immediately
 publish newly mined blocks, but in the case of a fork,
 they choose to mine on the block that provides the highest
 utility, i.e., the block that leaves the most remaining
 transaction fees to be included in the next block.⁴
- Selfish miners: An attacker employing some selfish mining strategy where mined blocks are temporary withheld and will be later revealed strategically.

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. To model this, we define γ as the fraction of honest miners (in terms of their collective hashrate β) who build on a selfish miner's block given a fork (so they received the selfish miner's 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 build on the selfish miner's block.⁵

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 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 a selfish miner's block 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.⁶

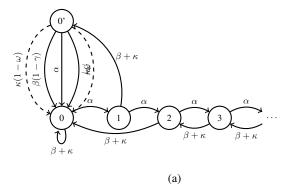
B. Selfish Mining Strategies: Existing and New Variations

We present the existing strategies that we evaluate for the first time in the transaction-fee regime and in the presence of

⁴If 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.

⁵It is expected that ω is either 0 or 1 as one branch is typically more profitable. However, in the case that the expected profit to mine on either branch is the same ω may be some value within this range.

⁶Again, this will typically be 0 or 1. However, there is a negligible chance multiple blocks of a fork have the same value, honest-but-rational miners decide between them with uniform probability. Due to its rarity, we omit a partial formalization similar to θ_i .



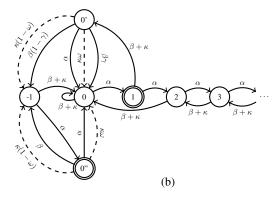


Fig. 2: FSMs for (a) classic selfish mining and (b) incentivized trailing selfish mining both in the presence of honest-but-rational miners. Dashed lines represent a new transition introduced by having honest-but-rational miners. For readability, we denote the state transition $\beta + \kappa$ by a single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

honest-but-rational miners, followed by a new variant that we devise based on this regime.

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. Lastly, we denote the composition of strategies with \circ separating the composed strategy notations.

New strategy variant—Incentivized trailing selfish mining. The presence of honest-but-rational miners introduces a new

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 its block during a fork—the fraction of honest miners who received this selfishly-mined block first and this fraction is represented by γ —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 [42], the $I_{(f,k)}$ class allows a k-trailing-selfish miner to incentivize honest-but-rational miners (during a fork from a trailing position) via the availability of future transaction fees, such that this availability is conditioned on the inclusion of the trailing-selfish miner's block of a fork. That is, upon publishing a block from a trailing position resulting in a fork, the miner will additionally release a transaction with fee f that is to be 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 early-on 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 to incentivize honest-but-rational miners with an additional fee f is released.

Beside the release timeline of these transactions, the details of how they are specified matter to the attack. 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 attempt to 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 currency 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.

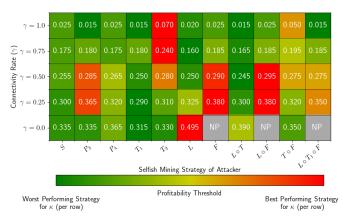


Fig. 3: Lead-dependent strategies profitability threshold.

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.

VI. EXPERIMENTAL EVALUATION

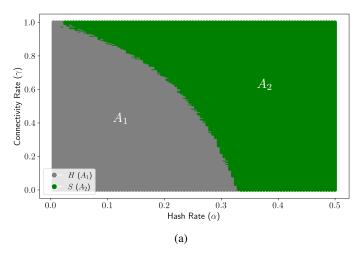
Our evaluation of the transaction-fee regime, for both the single and multi-attacker setting, is split among two different model settings; with and without honest-but-rational miners. The former has the goal of bridging the gap in understanding the performance of existing strategies within this regime.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

A. Implementation

We extended the mining simulator from [29] by implementing the set of strategies from Section V-B, and including an honest-but-rational actor. 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 delay in the publication of blocks. 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 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 mining honestly. Figure 3 shows a row-specific heatmap of the best and worst



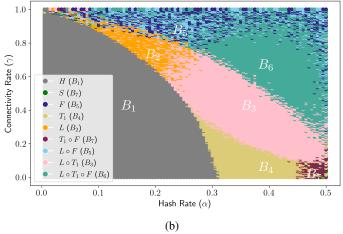


Fig. 4: Dominant (most profitable) lead-dependent mining strategy by parameterization (γ, α) . For (a), selfish mining versus honest mining is evaluated, in (b) this is extended to all lead-dependent mining strategies.

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 is not profitable under any hashrate (all subsequent figures use the same notation/color code).

Result 1: The security threshold of stubborn selfish mining strategies 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 where $\gamma=0$.

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 λ, 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

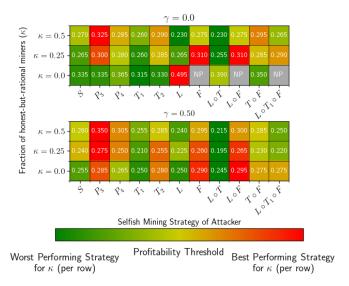


Fig. 5: Profitability threshold of lead-dependent selfish mining strategies in the presence of honest-but-rational miners.

honest mining under the same pair of parameters.

Figure 4a shows the dominant strategy between an honest and 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 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 leading. 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 [4], 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.^a

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 $\gamma \in \{0, 0.5\}$).

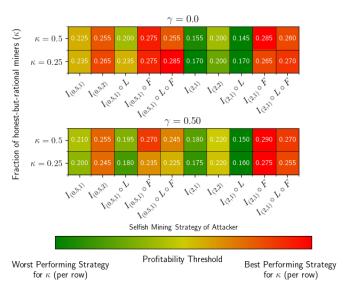


Fig. 6: Profitability threshold of $I_{(f,k)}$.

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% (when $\gamma=0$, upon a fork, no honest miners will mine upon a selfish miner's block).

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. That is, as honest-but-rational miners additionally include the revenue garnered from the inclusion of its 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 general profitability of these strategies, is that 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 block of a selfish miner, it is in the honest-but-rational miners best interest to mine upon the selfish miners block - being mined earlier it leaves additional transactions on the table to be included by this miner.

Insight 13: The presence of honest-but-rational miners ignites a mutually-beneficial relationship between them and selfish miners. An honest-but-rational miner may collect future revenue for the inclusion of a selfish miner's block whereas, a selfish miner can increase the chance of having its blocks be included in the public chain.

With honest-but-rational miners: new strategy variant. Finally, we examine the incentivized trailing selfish mining

^aExcept under select parameters in B_7 as shown in Figure 4.

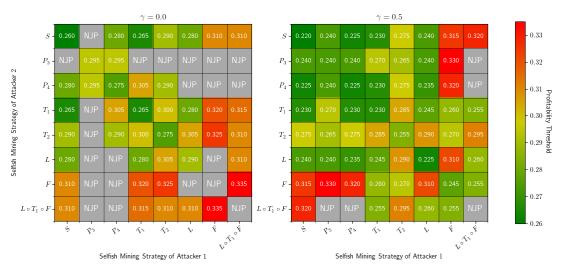


Fig. 7: Profitability threshold of an individual attacker in the 2-attacker setting.

 $I_{(f,k)}$ in the presence of honest-but-rational miners. We evaluate 10 variants, across varying incentives f, trailing positions $(k \in \{1,2\})$, and combinations 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 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-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 14: The class $I_{(f,k)}$ further improves the mutually-beneficial relationship between honest-butrational 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 2 non-colluding attackers. Our evaluations cover 64 combinations (e.g., attacker 1 employs S, and attacker 2 employs T_k). In

line with previous studies of the multi-attacker setting [9], we examine the profitability threshold when it is profitable for both attackers. Thus, when it is not mutually profitable, a selfish miner who is not achieving any profits may resort to honest mining, and consequently possibly affecting the profitability of the other once-profiting selfish miner.

Insight 15: 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 not profitable, we denote it as not-jointly-profitable (NJP).

As observed from the figure, in general the more risky mining strategies—that tend perform better in the single-attacker model (i.e., T_1 and L, as seen in Figure 3)—perform worse in the 2-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.

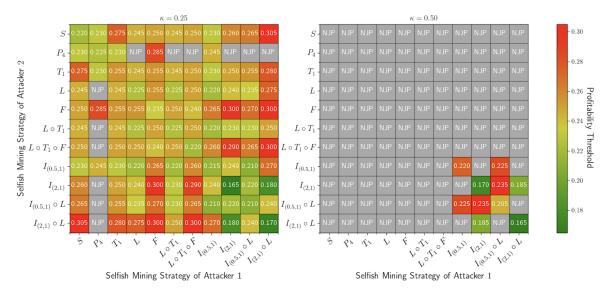


Fig. 8: Profitability threshold for an individual attacker in the 2-attacker setting in the presence of honest-but-rational miners

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' block of equal block value, if such a choice maximizes the honest-but-rational miner utility, it selects a block to mine upon uniformly at random.

Figure 8 shows our results when $\gamma=0$, $\kappa=0.25$, and $\alpha\in(0,0.75)$. We observe further reductions to the profitability threshold, as compared to the single-attacker setting (Figure 5), in the transaction-fee regime. For example, when $\gamma=0$, $\kappa=0.25$, it reduces the profitability threshold of an individual attacker utilizing classic selfish mining from 26.5% to 22%.

Result 7: Notably, we find that our class $I_{(f,k)}$ is particularly successful in this setting, with $(I_{(2,1)},I_{(2,1)})$ having a profitability threshold of 16.5% when $\kappa=0.25, \gamma=0$.

Insight 16: 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 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 hash rate 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 a 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, in the 2-attacker setting, $(I_{(2,1)},I_{(2,1)})$ have an individual profitability threshold of 17% (for $\gamma=0$ and $\kappa=0.5$). In the single attacker setting, when $\kappa=0.5$, $I_{(2,1)}$ has a security threshold of 15.5%. This indicates how selfish miners utilize their ability to incentivize honest-but-rational miners in the fork selection process.

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 the inclusion of transaction fees to miner extractable value (MEV), as well as an emerging system class, namely, decentralized (or blockchain-based) service systems. Furthermore, a study of security attacks would be incomplete without examining countermeasures. Thus, we discuss two relevant topics: detectability of selfish mining and defense mechanisms proposed so far in the literature. Lastly, we examine the state of selfish mining studies targeting systems other than Bitcoin.

A. Connections to Incentivized Mining Strategies and MEV

In recent years, a number of works have emerged showcasing the vulnerability of blockchains to attacks relying on incentivizing mining behavior; allowing an attacker to incentivize the inclusion/exclusion, or ordering of blocks and transactions. This is highlighted in miner extractable value (MEV), in which a miner can extract additional profits via the inclusion, exclusion, and ordering of transactions. Many of the used techniques are security attacks themselves, such as front-running, backrunning and sandwich attacks usually seen in trading systems such as automated market makers [43].

Even before MEV, incentives has long been a tool (and a motivation) for attacks; double-spending is one such example [42]. In the setting of mining strategies this is additionally true; [29] presented a mining strategy known as undercutting which attempts to reorder the head of the chain to introduce additional unnecessary forks. In doing so, via the inclusion of less transaction-fees within the block, the undercutting miner can incentivize other miners to mine upon their block by leaving additional transaction, and thus their fees, on the table potentially for those miners.

In the context of selfish mining, strategies that attempt to outwardly incentivize other miners have seen little examination, having only partial selfish mining [13] that attracts rational miners to collaborate and extend the selfish miner's private chain. Moreover, while presenting no such strategies, [29] discuss how undercutting may be a useful technique in conjunction with selfish mining. In our evaluation, we have confirmed the utility of incentives in promoting the success and profitability of selfish mining; in a way it is a form of undercutting. Furthermore, incentivized selfish mining can be viewed as an MEV technique for miners.

B. Connections to Decentralized Service Systems

In many blockchain-based systems, such as resource markets [44], 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 [45], [46]. A miner (server) collects service fees given the following condition: having the service contracts, and later the transactions that contain proofs of service delivery, be 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 variants of incentivized selfish mining strategies we introduced, and their 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. 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. A select group of previous works have evaluated techniques to detect selfish mining [34], [36], [39]. At their core, these works rely on the belief that a high frequency of excluded blocks or (stale) blocks may be as an indicator that selfish mining is occurring. Specifically, that chains of two or more stale blocks or an overall increase rate of stale block rates can be used to detect selfish mining.

While detection does not represent a direct defense against selfish mining, the knowledge of the presence of selfish miners may garner distrust and loss in value to the underlying blockchain; in effect causing incentive value loss for selfish miners. Furthermore, though these detection methods were found in theory, until recently no works have examined them in practice. A recent empirical study [41] 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.⁸

Whereas these detection methods seem to be practical, as discussed earlier, Bahrani et al. [10] presented a stealthy strategy to counter that. In fact, this work enables various selfish mining strategies to be formulated to become undetectable. This in turn makes reactive mechanisms, that act only when selfish mining is detected, ineffective. This is turn motivates the need for proactive defense mechanisms that aim to deter miners from attempting selfish mining in the first place.

⁷Though an adversarial mining strategy, this would not be classified as selfish mining as blocks are not withheld.

⁸On Monacoin, the time of this abnormal behavior aligns with what is a believed selfish mining attack, that was announced to have occurred between May 13-15, 2018 [47].

D. Defenses Against Selfish Mining

A number of works have developed selfish mining defense solutions [27], [28], [30], [31], [33], [38], [40]. Our goal is not to survey these solutions in details (surveys can be found in [24], [25]). Instead we focus on the network protocol and miner behavior changes that they present, which impact adoption in practice. In this light, these solutions can be classified into counter strategies (miner behavior-related) and network changes (those that rely on modifying the network protocol).

Counter strategies. Upon detection of a selfish miner, a counter-strategy miner adapts there mining behavior to a strategy that reduces profits for that selfish miner. 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. [33] 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 in attempt to learn what block they are mining upon, subsequently mining upon this block to competing against the pool.

Gal et al. [40] introduced piggybacking, where upon detecting a selfish miner, the piggybacking miners also withhold there blocks for an expected longer period of time than selfish mining. Maintaining this competition for a long period, a significantly large counter-miner can be expected (with high probability) to build a chain longer than the public chain which has endured 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 assumptions such as presence of a miner with a large hashrate, or the ability to join (and spy on) a selfish mining pool. This limits the effectivity of these defenses.

Network changes. Network changes make up a majority of the defense solutions [1], [27], [28], [30]–[32], [38]. 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 proposed an edit to the fork selection rule; instead of choosing the first block received, choose one of the received blocks uniformly at random. While this solution may mitigate some of the effects of highly connected 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. 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 [27], [30], 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. [28] proposed having miners publish intermediate blocks (valid blocks that meet an easier difficulty target) that are then used as the new head for the mining of subsequent blocks (either future normal blocks or intermediate ones). These blocks do not provide a reward and does not bring transaction on-chain, but are only used to minimize the spans between subsequent blocks being produced—diminishing the window where selfish mining can occur. Another work [31] utilized the idea of a weighted fork resolving policy—in a similar way to the inclusion of uncle blocks in the proof-of-work version of 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 protocols. For example, Fruitchain [32] aims to enforce the reward fairness property, i.e., miners obtain rewards in proportion to their hashrate. While [14] advocated for including valid stale/orphaned blocks in the difficulty adjustment algorithm as a way to avoid producing easier difficulty targets that may benefit selfish miners. However, [16] showed that even under this modified algorithm, several selfish mining attacks may still be profitable.

Finally, when including transaction fees, Xiongfei et al. [38] suggest limiting fee volatility (as it has been found impactful for improved selfish mining and undercutting [29]). This is done by capping the number of transaction 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 protocol.

E. Studying Selfish Mining in Other Systems

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-ofwork version [18], [19] (contextualizing the effect of uncle rewards on classic selfish mining), and the more recent proofof-stake (PoS) version [23] (the formulation of a new strategy class specific to the network protocol). Selfish mining strategies have additionally been formulated specific to other protocols, including for longest chain PoS protocols [15], such as Tezoz and Cardano, and for Filecoin [45]. These new strategies represent a high level adaptation to "selfish proposing" attacks (in systems where new blocks are proposed rather than mined (i.e., PoS) (Filecoin appends blocks through a somewhat similar system though not PoS).

While relying on often specific protocol features to formulate an attack, still each of these works utilizes temporary-block withholding. Nonetheless, modeling and obtained results vary from protocol to protocol based on their specifications. Keller et al. [48] have attempted to bridge this gap formulating a generic Markov decision process for selfish mining attacks on DAG-based protocols, and mentioned that it could be adopted for other protocols, however they show results only for Bitcoin. Thus, more work is still needed to understand selfish mining strategies at a generic level.

VIII. CONCLUSION

We presented a systematization framework categorizing existing selfish mining works according to their strategy and model formulation. 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 the 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," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.
- [2] L. Bahack, "Theoretical bitcoin attacks with less than half of the computational power (draft)," 2013.
- [3] H. Liu, N. Ruan, R. Du, and W. Jia, "On the strategy and behavior of bitcoin mining with n-attackers," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 357–368
- [4] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016, pp. 305–320.
- [5] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security*, J. Grossklags and B. Preneel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 515–532.
- [6] 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.
- [7] 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 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2021, pp. 694–703.
- [8] 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.
- [9] S. Zhang, K. Zhang, and B. Kemme, "Analysing the benefit of selfish mining with multiple players," in 2020 IEEE International Conference on Blockchain (Blockchain). IEEE, 2020, pp. 36–44.
- [10] M. Bahrani and S. M. Weinberg, "Undetectable selfish mining," arXiv preprint arXiv:2309.06847, 2023.
- [11] F. J. Marmolejo-Cossío, E. Brigham, B. Sela, and J. Katz, "Competing (semi-) selfish miners in bitcoin," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 89–109.
- [12] R. Bar-Zur, A. Abu-Hanna, I. Eyal, and A. Tamar, "Werlman: To tackle whale (transactions), go deep (rl)," in 2023 IEEE Symposium on Security and Privacy (SP). IEEE Computer Society, 2023, pp. 93–110.
- [13] J. Yu, S. Gao, R. Song, Z. Cai, and B. Xiao, "Partial selfish mining for more profits," arXiv preprint arXiv:2207.13478, 2022.

- [14] C. Grunspan and R. Pérez-Marco, "On profitability of selfish mining," arXiv preprint arXiv:1805.08281, 2018.
- [15] R. Sarenche, S. Nikova, and B. Preneel, "Deep selfish proposing in longest-chain proof-of-stake protocols," *Cryptology ePrint Archive*, 2024.
- [16] R. Sarenche, R. Zhang, S. Nikova, and B. Preneel, "Time-averaged analysis of selfish mining in bitcoin," Cryptology ePrint Archive, 2024.
- [17] T. Cao and X. Li, "Temporary block withholding attacks on filecoin's expected consensus," in *Proceedings of the 26th International Symposium* on Research in Attacks, Intrusions and Defenses, 2023, pp. 109–122.
- [18] F. Ritz and A. Zugenmaier, "The impact of uncle rewards on selfish mining in ethereum," in 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, 2018, pp. 50–57.
- [19] 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.
- [20] 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."
- [21] 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.
- [22] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong, "A deep dive into blockchain selfish mining," in *ICC 2019-2019 IEEE international* conference on communications (ICC). IEEE, 2019, pp. 1–6.
- [23] J. Neu, E. N. Tas, and D. Tse, "Two more attacks on proof-of-stake ghost/ethereum," in *Proceedings of the 2022 ACM Workshop on Developments in Consensus*, 2022, pp. 43–52.
- [24] K. Nicolas, Y. Wang, and G. C. Giakos, "Comprehensive overview of selfish mining and double spending attack countermeasures," in 2019 IEEE 40th Sarnoff Symposium. IEEE, 2019, pp. 1–6.
- [25] 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.
- [26] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," arXiv preprint arXiv:1311.0243, 2013.
- [27] E. Heilman, "One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner," in Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers 18. Springer, 2014, pp. 161–162.
- [28] R. Zhang and B. Preneel, "Broadcasting intermediate blocks as a defense mechanism against selfish-mine in bitcoin," *Cryptology ePrint Archive*, 2015.
- [29] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of* the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 154–167.
- [30] S. Solat and M. Potop-Butucaru, "Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin," arXiv preprint arXiv:1605.02435, 2016.
- [31] R. Zhang and B. Preneel, "Publish or perish: A backward-compatible defense against selfish mining in bitcoin," in *Topics in Cryptology–CT-RSA 2017: The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14–17, 2017, Proceedings.* Springer, 2017, pp. 277–292.
- [32] R. Pass and E. Shi, "Fruitchains: A fair blockchain," in *Proceedings of the ACM symposium on principles of distributed computing*, 2017, pp. 315–324.
- [33] S. Lee and S. Kim, "Detective mining: Selfish mining becomes unrealistic under mining pool environment," *Cryptology ePrint Archive*, 2019.
- [34] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in 2019 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2019, pp. 360–364.
- [35] K. A. Negy, P. R. Rizun, and E. G. Sirer, "Selfish mining re-examined," in Financial Cryptography and Data Security, J. Bonneau and N. Heninger, Eds. Cham: Springer International Publishing, 2020, pp. 61–78.
- [36] 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.
- [37] 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.

- [38] X. Zhao and Y.-W. Si, "Dynamic transaction storage strategies for a sustainable blockchain," in 2021 IEEE International Conference on Services Computing (SCC). IEEE, 2021, pp. 309-318.
- [39] 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, p. 5959698, 2021.
- [40] J. Gal and M. B. Szabo, "Majority is not needed: A counterstrategy to selfish mining," arXiv preprint arXiv:2304.06313, 2023.
- [41] S.-N. Li, C. Campajola, and C. J. Tessone, "Statistical detection of selfish mining in proof-of-work blockchain systems," Sci. Rep., vol. 14, no. 1, p. 6251, Mar. 2024.
- [42] K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21. Springer, 2017, pp. 264–279.
- [43] 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 2020 IEEE Symposium on Security and Privacy (SP). IEEE, 2020, pp. 910-927.
- [44] G. Almashaqbeh, "Rethinking service systems: A path towards secure and equitable resource markets," USENIX ;login: Magazine, 2021.
- [45] "Filecoin," https://filecoin.io/. [46] "Livepeer," https://livepeer.com/.
- [47] 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?
- [48] P. Keller and G. Bissias, "Generic selfish mining mdp for dag protocols," arXiv preprint arXiv:2309.11924, 2023.