**Exploring Profitable Miner Strategies: A Comparative Analysis Beyond Honest Mining in Bitcoin**

First Author, Second Author, and Third Author

firstauthor@virginia.edu, secondauthor@vt.edu, thirdauthor@gmu.edu

***Abstract* – In recent years, there has been a surge in the popularity of cryptocurrencies because of the trust, transparency, security, and accessibility they provide in contrast to centralized financial systems. The most well-known cryptocurrency is Bitcoin, introduced in 2009 as a solution to the problems associated with centralized management of transactions that has dominated the entire cryptocurrency market. As of March 2024, there are around 8985 active cryptocurrencies, with 420 million cryptocurrency users across the globe and approximately 18000 businesses have integrated cryptocurrency payment methods, reflecting the growing acceptance of digital assets in the commercial sector [[1]](#footnote-1). In decentralized cryptocurrency, for instance, in Bitcoin, the transactions are verified by bitcoin users who possess the required hardware and computing power. This process is solved based on a difficult mathematical puzzle called Proof of Work (PoW) and all the miners compete among themselves to solve the PoW puzzle and the miner who solves the puzzle receives a reward in the form of bitcoins. The miners follow different mining strategies with the intention of maximizing their rewards. For a long time, the most profitable mining strategy was believed to be the honest mining strategy. But it was proved later to gain more mining rewards by selfish mining deviating from honest mining by withholding the newly solved blocks without publishing them to the network. However recent research shows that selfish mining attacks may not be optimal and proposed different versions of selfish mining allowing the attacker to earn potentially higher rewards. Moreover, authors formulated the mining problem as a Markov Decision Process (MDP) which can be solved to derive an optimal mining strategy. Later, authors extended this model for other cryptocurrencies such as Ethereum. In a more recent study, authors have adopted the MDP model, employing Reinforcement Learning to achieve an optimal mining strategy. This paper outlines the findings of our Systematic Literature Review, which delves into the latest research and advancements in the domain of Bitcoin and Ethereum mining strategies. Our objective is to investigate and assess the diverse mining strategies that are more profitable than honest mining.**

***Index Terms* - Blockchain, Proof-of-Work, Profitable Mining, Selfish Mining, Reinforcement Learning, Markov Decision Process**

1. **Introduction**

[[2]](#footnote-2)Cryptocurrency is a digital or virtual form of currency that utilizes cryptography for security and operates on decentralized networks, meaning they are not controlled by a central governing body such as a government or financial institution. The market leader and original cryptocurrency in the cryptocurrency market is Bitcoin which makes up 48.6% of the total value of the crypto market as of 2024. Bitcoin cryptocurrency was introduced in the 2009 global economic crisis to overcome the problems of centralized transaction management providing several benefits such as improved trust, security, transparency among member organizations by improving the traceability of data shared across a business network and delivering cost savings through new efficiencies. [1], [2]. This makes blockchain more secure and less prone to fraud, tampering or general system failure than keeping them in a single centralized location [3]-[5].

Bitcoin cryptocurrency was introduced in the 2008 global economic crisis to overcome the problems of centralized transaction management providing several benefits such as improved trust, security, transparency among member organizations by improving the traceability of data shared across a business network and delivering cost savings through new efficiencies. For years, Bitcoin dominated the crypto industry. Today, investors can choose from thousands of different cryptocurrencies. Bitcoin constitutes approximately fifty percent of the total cryptocurrency market capitalization. As of February 2024, the global cryptocurrency market cap is $2.09 trillion whereas bitcoin’s market cap of $1.02 trillion accounts for around 50% of that total.

All the transactions are verified by bitcoin users who have the required hardware and computing power. These entities are called miners. This mining process is solved based on a difficult cryptographic puzzle called proof of work (PoW). Bitcoin uses the proof of work (PoW) as the underlying consensus algorithm for bitcoin mining [3],[4],[5].  Therefore, every miner attempts to solve the mathematical puzzle competing among themselves to validate the transaction and to create a new block in the chain. As soon as the miners successfully created a valid block, he gets rewarded. Miners must allocate necessary hardware and computing resources to tackle the PoW challenge. The goal of a miner is to earn the maximum possible reward. To tackle this problem, miners can follow various strategies to solve the PoW challenge. Honest mining is one of the most popular and earliest strategies in which the miner promptly broadcasts newly solved blocks across the network. This approach assumes that participating honestly in the network, by immediately sharing valid blocks, will yield the most profitable outcomes for miners in the long term.

Honest miners are participants in a cryptocurrency network who engage in the mining process according to the rules and protocols established by the network. They contribute their computational power to validate and secure transactions, add new blocks to the blockchain, and maintain the network's integrity. Honest miners follow the consensus rules of the network, compete fairly for block rewards, and do not engage in malicious activities such as double-spending, selfish mining, or other forms of manipulation. Therefore a miner with control over α fraction of the network’s computational resources should ideally receive only α fraction of the mining reward. However, a malicious attacker can deploy diverse tactics to unfairly acquire a larger portion of the mining reward. These tactics aim to provide a specific group of malicious miners with an unfair advantage, ensuring they receive a disproportionate share of mining rewards compared to their legitimate contribution of computational power. This unfair advantage enables them to achieve greater expected revenue from mining activities than what would be equitable based on their actual computational resources.. These tactics are known as *mining attacks*.

Selfish mining is a strategy in cryptocurrency mining where miners withhold newly solved blocks instead of immediately broadcasting them to the network [6]. Because of the self-adjusting behavior of Bitcoin to ensure that on average only one block is added to the blockchain every 10 minutes [7], this tactic allows the selfish miner to gain a competitive advantage by secretly working on the next block while other miners continue to work on the current one. Once the selfish miner finds the next block, they release both blocks, causing other miners' work on the current block to become wasted. This enables the selfish miner to earn a larger share of the block rewards compared to other miners, ultimately increasing their profits at the expense of the overall network integrity. Selfish mining has proved to be more profitable than honest mining, allowing the miner to collect higher rewards. For example, with computing power ratio α = ¼, the rewards obtained by selfish mining can be up to 1/3 fraction of the total rewards [8]. Based on this observation, [9] further proposed various selfish mining strategies with even higher rewards. However different studies show that selfish mining may not be optimal under certain strategies [8],[9],[12]. Despite the many versions of selfish mining, the optimal (i.e., most-profitable) mining strategy remained elusive until [9].

The authors of [9] formulated the mining problem as a general Markov Decision Process (MDP) with a large state-action space. The objective of the mining MDP, however, is not a linear function of the rewards as in standard MDPs. Thus, the mining MDP cannot be solved using a standard MDP solver. To solve the problem, [9] first transformed the mining MDP with the non-linear objective to a family of MDPs with linear objectives, and then employed a standard MDP solver over the family of MDPs to iteratively search for the optimal mining strategy.

This paper aims to unveil the findings from our Systematic Literature Review, which delves into the evolution of mining strategies with the aim of earning higher profit than others while attempting to answer the following questions.

RQ1: What are the extant mining strategies that yield greater profitability compared to honest mining?

RQ2: What are the limitations of these mining techniques?

RQ3: What are the possible research gaps and areas of improvement that can be further studied?

1. **Background**

In this section, we briefly recap the blockchain preliminaries, operation of the consensus algorithm namely proof-of-work (PoW) and mining process.

* 1. **Proof of Work and Mining**

Blockchain is a decentralized, distributed ledger technology that enables secure and transparent recording of transactions across a network of computers. At its core, a blockchain is a chain of blocks, each containing a list of transactions. These blocks are linked together using cryptographic techniques, forming a chronological chain that is immutable and resistant to tampering.

Bitcoin mining is the process of validating and adding new transactions to the Bitcoin blockchain, as well as the mechanism through which new bitcoins are created. It is a crucial component of the decentralized consensus mechanism that underpins the Bitcoin network. The structure of a block in the Bitcoin blockchain typically consists of components such as Block Header, Transactions, Block Size, Block Height, Block Hash and Block Reward. The block header contains metadata about the block and is used to validate its integrity. It includes Version, Previous Block Hash, Merkle Root, Timestamp, Nonce, Difficulty Target. It’s important to note that blocks in the blockchain are linked together using cryptographic hashing and referencing the hash of the previous block within the block header. (See Fig. 1)

The Proof of Work (PoW) algorithm is the consensus mechanism used in blockchain networks, including Bitcoin, to achieve agreement on the state of the blockchain and to validate transactions. It requires participants, known as miners, to solve complex mathematical puzzles to create new blocks and add them to the blockchain. To create a new block, miners gather a set of pending transactions from the network and combine them into a block. This block also contains metadata such as a timestamp and a reference to the previous block's hash. Miners select a random number called a nonce (number used once) and include it in the block header. The nonce is a 32-bit (4-byte) field that miners can vary to find a valid block hash. Miners hash the block header along with the nonce using a cryptographic hash function, such as SHA-256. This produces a hash value, which is essentially a random string of numbers and letters. The hash value is compared to a target value set by the network. This process can be denoted mathematically as follows.

Where n is the nonce value, p is the hash value of the previous block, m is the Merkle root of all the included transactions in the block, and D is the target. The target represents the difficulty level of the puzzle and is adjusted periodically to ensure that new blocks are mined, on average, every 10 minutes in the case of Bitcoin. If the hash value meets the difficulty target (i.e., it is lower than the target), the miner has found a valid solution to the puzzle. This is known as “proof of work" because the miner has demonstrated that they have expended computational effort (work) to find a valid hash. The miner broadcasts the new block to the network, along with the nonce and hash value. Other nodes in the network verify the validity of the block by independently hashing the block header with the nonce and comparing the resulting hash to the target. If the block is valid, it is accepted by the network and added to the blockchain. The miner who successfully mines a new block is rewarded with a certain number of newly created bitcoins, known as the block reward. Additionally, they may collect transaction fees associated with the transactions included in the block.

What happens when two miners publish different valid blocks referencing the same preceding block simultaneously? If it happens, a temporary situation known as a blockchain *fork* occurs. This scenario can happen due to the decentralized and asynchronous nature of blockchain networks, where multiple miners may find valid solutions to the cryptographic puzzle required to create a new block simultaneously. When two miners independently create valid blocks that reference the same preceding block, the blockchain temporarily forks into two separate branches. Each branch contains a different valid block at the same height. Both blocks are broadcast to the network, and nodes in the network receive and propagate both blocks. As a result, different parts of the network may initially see different versions of the blockchain, leading to a temporary lack of consensus on which block is the "correct" one. Miners and nodes in the network will continue to build upon the block they received first, extending their respective branches of the blockchain. This leads to a race to find the next block, with miners attempting to create longer chains to establish dominance. Eventually, one of the branches will become longer than the other as miners add more blocks to it. When this happens, nodes in the network recognize the longer chain as the valid one according to the blockchain's consensus rules. The shorter branch is then discarded, and the network returns to a single, unified chain. The blocks that were part of the shorter branch (i.e., not included in the longer chain) become orphaned or stale blocks. Transactions included in these orphaned blocks are returned to the memory pool and can be included in future blocks.

**How Pools Work**

1. **Methodology**

The study employed a systematic literature review approach to fulfill its objectives and gain a thorough understanding of potential experimental work in the possibility of applying Artificial Intelligence to achieve the optimal mining strategy in Bitcoin networks. Following the updated 2020 Preferred Reporting Items for Systematic Reviews (PRISMA) guidelines [4], we established a protocol outlining the research questions, information sources, search strategy, selection criteria, data extraction, and analysis. This ensured transparency, replicability, and scientific adequacy in the systematic review process.

* 1. **Formulation of research questions**

RQ1: What are the extant mining strategies that yield greater profitability compared to honest mining?

RQ2: What are the limitations of these mining techniques?

RQ3: What are the possible research gaps and areas of improvement that can be further studied?

* 1. **Data Sources and Search Strategies**

The systematic literature review conducted in this study involved an in-depth exploration of published articles spanning the period from 2018 to 2024 across a wider range of electronic databases: Scopus, IEEE Xplore Digital Library, Springer Link and Google Scholar. These databases were selected due to their globally acknowledged impact indices, which encompass a wide array of peer-reviewed scientific and scholarly literature from various scientific domains and disciplines worldwide.

We determined search strings by identifying associated key terms within blockchain mining. This was based on our subject knowledge and previous most-cited research papers and journals.

We established two generic search terms in association with informatics and employed Boolean operators as follows to encompass all literature focusing on profitable mining strategies and to capture information on Bitcoin and Ethereum cryptocurrencies.

*(Bitcoin OR Ethereum) AND profitable AND mining*

Drawing upon subject knowledge, we identified various types of attacks or mining strategies that deviate from standard Bitcoin and Ethereum protocols. Subsequently, we formulated search terms tailored to capture these strategies effectively as given below.

* selfish mining
* double spending
* 51% attack
* time drift attack
* eclipse attack
* pool hopping
* grinding attack
* stubborn mining
* finney attack
* majority attack
* race attack
* siphoning attack

Subsequently, we combined these key values into the generic query and executed iterative queries across digital libraries for each mining strategy.

* 1. **Selection of studies**

These inclusion criteria help ensure that the systematic review focuses on relevant, high-quality research that contributes meaningfully to understanding profitable mining strategies deviating from Honest mining.

* Empirical studies employing quantitative, qualitative, or mixed-methods research designs, including experimental studies, case studies, surveys, and observational studies.
* Articles specifically addressing the development, implementation, evaluation, or impact of different mining strategies on Bitcoin and Ethereum.
* Articles published within the last ten years (2014-2024) to capture recent advancements and developments in the field.
* Only the articles published in English are considered.
* Articles with a clear description of research methods, data collection techniques, analysis procedures, and consideration of potential sources of bias.
* Articles available through academic databases, institutional repositories, or other accessible sources for data extraction and analysis.
  1. **Data Extraction and Quality Assessment**
  2. **Constitution of the Corpus of Analysis**
  3. **Characteristics of included Studies**

1. **Results**

In this section, we outline the outcomes derived from the systematic review process described earlier. These findings are structured based on the research inquiries outlined in Section 3.1 that directed our investigation and subsequent analysis.

* 1. **RQ1: What are the extant mining strategies that yield greater profitability compared to honest mining?**

Authors of [6] formulated a strategy called Selfish Mining that can be used by a minority pool to obtain more revenue than the pool’s ratio of the total mining power. Under selfish mining the authors of [6] emphasized the strategy of a pool to keep deliberately forking the chain by keeping its discovered blocks private. Meanwhile, honest nodes persist in mining on the public chain, while the pool focuses on its separate, private branch. By consistently uncovering more blocks, the pool establishes a greater lead on the public chain, maintaining the secrecy of these new blocks. Once the public chain nears the length of the pool's private branch, the selfish miners disclose blocks from their private chain to the public. Consequently, this approach induces honest miners adhering to the Bitcoin protocol to expend resources on solving cryptographic puzzles that ultimately serve no purpose. Their analysis [6] indicates that although both honest and selfish participants waste resources to some extent, the honest miners incur a comparatively higher waste, and the rewards for the selfish pool surpass its share of the network's mining capacity. This situation provides the selfish pool with a competitive edge and motivates rational miners to join it. [6] showed that once a selfish mining pool reaches a certain threshold, rational miners will preferentially join selfish miners to earn higher revenues compared to other pools.

. Authors of [9] formalized a bitcoin mining model incorporating the relevant features of mining and system parameters. Suppose the system contains a set of miners where each denotes the mining power of miner such that,

Without loss of generality, let us assume that these miners are divided into two groups as selfish miners (a minority pool those who keep their mined blocks private) and honest miners (A majority those who adheres to the standard bitcoin protocol and mines on the long public branch). Suppose that the pool controls fraction of the computing power of the whole network and be the communication capability of the pool: The fraction of the honest miners that will first receive a block publish by the pool if the pool and one honest miner choose to release their blocks approximately at the same time. The main idea of selfish mining [9] is described as follows.

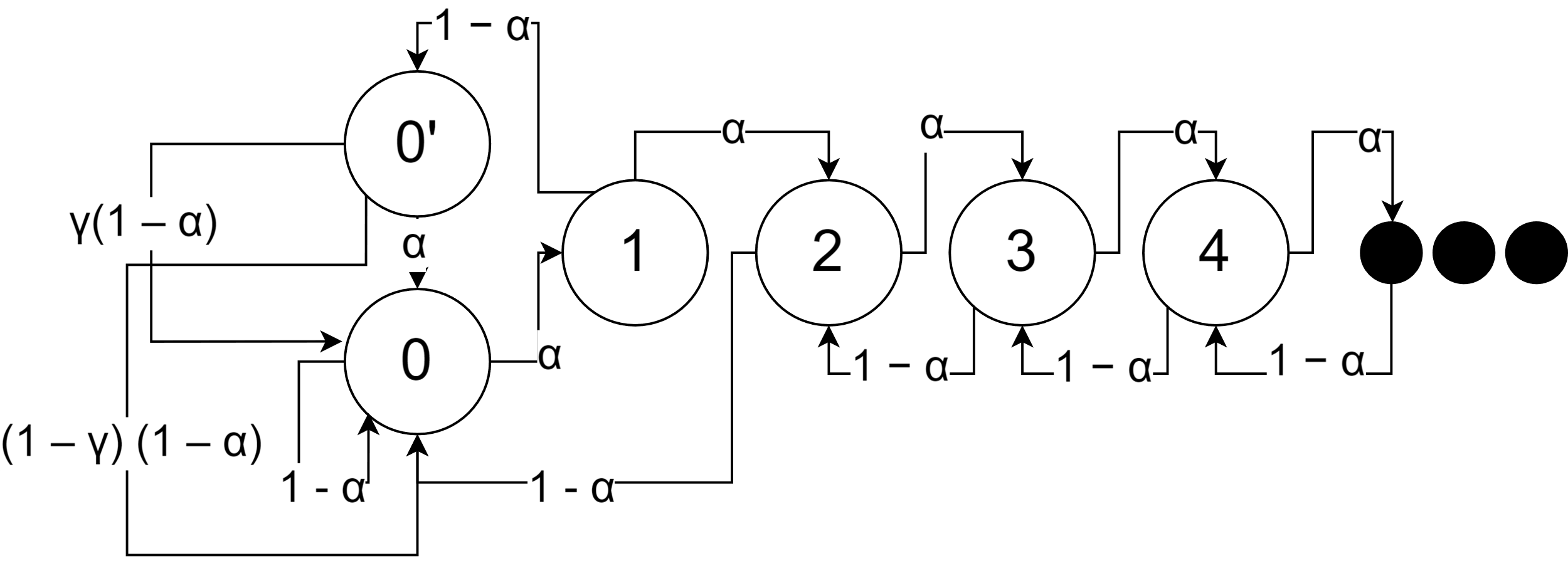
When the selfish mining group discovers a block, they gain an advantageous position by having a one-block lead over the public branch where honest miners are working. Instead of immediately sharing this private block and informing other miners about the discovery, selfish miners opt to keep it within their pool. At this juncture, two scenarios can unfold:

1. The honest miners find a new block on the public branch, thereby canceling out the pool's lead.
2. The pool manages to mine a second block, further widening its lead over the honest miners.

In scenario I, the pool chooses to publish its block to match the honest network. The selfish miners unanimously decide to adopt and extend the previously private branch, whereas honest miners make their choice based on the dissemination of notifications, opting to mine on either branch. If the selfish pool succeeds in mining a subsequent block before the honest miners who haven't adopted the pool's recently revealed block, it promptly publishes it to benefit from the earnings of both the first and second blocks of its branch. Conversely, if the honest miners mine a block following the pool's revealed block, the pool reaps the rewards of its block while the others gain revenue from their own block. Finally, if the honest miners mine a block subsequent to their own, they reap the benefits of their two blocks while the pool receives nothing.

In scenario II, should the selfish pool succeed in securing a second block, it establishes a two-block lead. Once reaching this stage, the pool persists in mining at the forefront of its private branch, releasing one block from its private branch for each block discovered by others. As the selfish pool represents a minority, its lead will likely diminish to just one block over time. At this juncture, the pool unveils its private branch. Given that the private branch surpasses the public branch by a block, all miners unanimously adopt it as the primary branch, allowing the pool to profit from all its blocks.

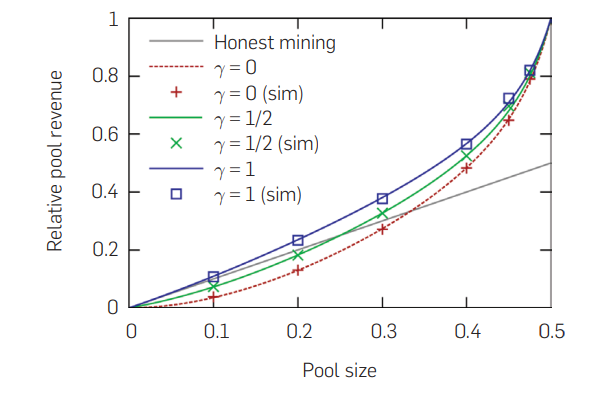
To investigate the above operation further, [9] presents a state machine capturing the transition frequencies involving and .



Each state in the above state machine represents the lead: The difference between the length of the pool’s private branch and the length of the public branch. Note the distinction between state 0 and 0’. State 0 is the state where there is only a single, global, public longest chain and the state 0’ is the state where there are two public branches of length one: the main branch, and the branch that was private to the selfish miners and published to match the main branch. Transition between each state is triggered by a block mining operation either by the pool with a probability of or honest network with a probability of .

Authors of [9] formulated the revenue rate of each agent: *revenue rate ratio* in the pool as given below.

However as highlighted in [9], [11] the drawback of selfish mining is that it is not always more profitable than honest mining. It means there are certain scenarios where honest mining may shine. This limitation happens when nodes do not employ a large enough share of the computing resources and also due to propagation limitations in the network.



The above figure illustrates the pool’s revenue for different values with varying from 0 to 0.5.

When , selfish mining is much better than honest mining as the pool can quickly propagate its one-block to all the honest miners, so all honest miners will mine on the pool’s block. However, when honest mining looks more profitable for smaller pool sizes.

However as highlighted in [6], [11] the drawback of selfish mining is that it is not always more profitable than honest mining. It means there are certain scenarios where honest mining may shine. This limitation happens when nodes do not employ a large enough share of the computing resources and also due to propagation limitations in the network. In their study [6], they showed that an attacker can earn a higher profit if the pool has a mining power which exceeds 25% profit-threshold i.e. The minimal computational power an attacker needs in order to gain more than its fair share.

Selfish mining within the Bitcoin ecosystem has undergone thorough examination, with a range of mining strategies being put forth for consideration. Nevertheless, the prevailing selfish mining models cannot be readily transferred to Ethereum due to disparities in the mining framework compared to Bitcoin. This distinction arises from Ethereum's provision of two block rewards: uncle block reward, and nephew block reward in addition to the standard block reward in the Bitcoin system. Building upon the findings of [6], the authors of [10] have investigated the selfish mining strategy within the context of Ethereum. They introduced a two-dimensional Markov Decision Process (MDP) and tracked the block rewards in a probabilistic way to effectively model the behavior associated with selfish mining strategies capturing the impact of uncle and nephew rewards specific to Ethereum. Utilizing this mathematical framework, they have derived that the threshold of computational power required for an attacker to gain from selfish mining stands at 16.3%. This figure is notably lower than the profit threshold of 25% established in [6]. Indeed, the inclusion of uncle and nephew rewards in Ethereum's mining scheme amplifies the threat posed by selfish mining, making it a more significant concern for the Ethereum network compared to Bitcoin. This indicates that Ethereum exhibits a higher susceptibility to selfish mining compared to Bitcoin.

[8] presented a family of mining strategies called "stubborn mining" namely Lead Stubborn, Equal Stubborn and Trail Stubborn that go beyond and surpass the selfish mining approach allowing the miners to earn even higher rewards than selfish mining. The intuition behind stubborn mining strategies is the attacker can often gain more profit by mining on his/her private chain more frequently even if the attacker’s private chain falls behind the honest chain. This stands in contrast to the selfish mining strategy, where the selfish miner withholds mined blocks and publishes them solely when they find themselves trailing the honest chain. Depending on the environmental parameters, their analysis [8] shows that stubborn mining strategies can surpass selfish mining by as much as 25%, even without employing any network-level attacks. Their analysis subsequently reveals that employing the trail-stubborn mining strategy can yield a 13% increase in gains compared to a non-trail-stubborn counterpart. Authors of [8] further show that an attacker can increase his/her reward by following non-trivial combinations of stubborn mining and network-level eclipse attacks [19] by exploiting the network layer subsequently inspecting and controlling incoming and outgoing connections of the victim to further increase his/her revenue. Depending on the parameters, their work shows that these strategies can at times yield gains of up to 30% when compared with the naive utilization of eclipsed nodes.

As the selfish mining strategy proposed in [6] was not optimal, [9] extended the MDP mining models of [6], [8] to a more generalized form. In [9] authors showed that there are selfish mining strategies that allow miners to earn a higher reward and also profitable for small miners compared to [6]. To do that, first, [9] formulated the mining problem as a single-player decision problem of the form where is the state space, is the action space, is the stochastic transition matrix, and is the reward matrix. However, M cannot be considered as an MDP as the objective function is nonlinear: The player aims to maximize its share of the accepted blocks, rather than the absolute number of its own accepted ones.

Let be the number of blocks that have been built by the attacker after the latest fork (length of attacker chain) and be the number of those built by honest nodes (length of the honest chain).

**Action:** Action space consists of four possible actions as given below.

* *Adopt*:The attacker accepts the honest chain and mines on the last block of the honest chain
* *Override:*  The attacker publishes blocks to override the conflicting blocks of the honest chain, and is feasible whenever a > h.
* *Match*: The attacker publishes the same number of blocks as the honest chain to the whole network. This action creates a fork deliberately and initiates an open mining competition between the two branches of the adversary and the honest network.
* *Wait*: The attacker does not publish blocks and keeps mining on its own chain.

**State**: Each state in the state space is denoted by a tuple of size 3 of the form . The fork entry can take three values as given below.

* *Relevant*: if the latest block is mined by the honest network. For instance, if the previous state was and the honest network was able to mine a block then state changes to
* *Irrelevant*: if the latest block is mined by the attacker. For instance, if the previous state was and the pool was able to mine a block then state changes to
* Active: if the pool has executed the action match from the previous state, and the blockchain is now split into two branches. It simply means a fork has been known to the public and there is active competition between the two branches.

Note that the initial state of the network is with probability or with .

**Transition and Reward Matrices**: Execution of an action, every state transition corresponds to the creation of a new block either by the honest network or attacker. A description of the corresponding transition and reward matrices is shown in Table 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **State x Action** | **State** | **Probability** | **Reward** |
|  |  |  |  |
|  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |

**Objective Function**: As derived in [9], we define the relative revenue of the attacker as given below.

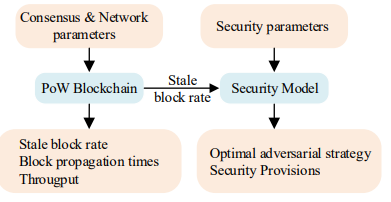
where ,is the immediate reward issued in the block interval under the action defined by policy , and is the size of the observing window.

Based on the provided formula, the adversary's goal is to maximize the ratio of its total rewards to the total rewards achieved by the entire network but not to maximize its absolute total reward. In other words, the adversary aims to maximize its portion of accepted blocks rather than simply the absolute count of its own accepted blocks. This is because blockchain adjusts its mining difficulty regularly to maintain a consistent block production rate, for instance, approximately one block every 10 minutes in Bitcoin. This adjustment is crucial for the network's stability and security. As more miners join or leave the network, the total computing power dedicated to mining fluctuates. By adjusting the difficulty, Bitcoin ensures that blocks are neither generated too quickly nor too slowly, maintaining the intended block time and, consequently, the overall functionality and reliability of the network. To derive an optimal policy, since the objective function is nonlinear, [9] first transformed the model into a family of MDPs with linear objectives and then used a standard MDP solver combined with a numerical search over the family of MDPs to find the optimal mining strategy. Utilizing these MDP families [9] obtained upper and lower bounds on the attacker’s profit. Based on their simulations [9], it was demonstrated that an attacker can achieve a greater portion of the rewards by employing a lower profit threshold of 23.21%, in contrast to the 25% profit threshold illustrated in the previous study [6].

However, the limitation of this approach is, this is a model-based approach. It means that this approach requires the knowledge of parameters of the network such as the attacker’s computational power () and communication capability (). In real blockchain networks, these values are difficult to determine as they may change over time.

Studies [6]-[9] have employed models to analyze the selfish mining strategy, operating under the premise of a singular selfish miner, while considering the potential presence of multiple colluding pools nearing the profitability threshold. Diverging from the aforementioned assumption, the authors of [11] introduced a novel MDP to depict the state transitions between public and private chains, taking into account the presence of multiple selfish miners within the Bitcoin network and setting out a limit on the maximum length of the private chain. The selfish mining scenario was modeled with the inclusion of an honest pool representing all legitimate miners within the network, alongside two independent selfish miners unaware of each other's noncompliant behavior. In their paper [11], the authors addressed the question of profitability by taking into account both the hash rate of the attackers and the adjustments made to the mining difficulty. They demonstrated that for two selfish miners to achieve profitability, the minimum required threshold of profitability (Hashrate) is symmetrically set at 21.48% and becomes more challenging when one of the selfish miners increases their hash rate. They also demonstrated that if the hash rates of selfish miners are both set at 22%, then the attackers can reap the benefits of selfish mining after 51 rounds of mining difficulty adjustments (equivalent to 714 days in Bitcoin), and this period reduces to 5 rounds (equivalent to 70 days) when the hash rates are increased to 33%.

In the study [12], authors introduced an innovative quantitative framework designed to assess the security and performance associated with different consensus and network parameters within Proof of Work (PoW) blockchains. Their framework comprised two essential components, namely a blockchain instance and a blockchain security model. In their model, they instantiated the PoW blockchain with various consensus and network parameters, such as network delays, block propagation times, block sizes, and information propagation mechanisms. This approach aimed to more realistically capture real-world blockchain instances. The output of their security model was the stale block rate, which served as input to their security framework which was built upon the work laid out by [9], encompassing more real-world features such as stale block rates, network delays, eclipse attacks [19], block propagation times, block size, block generation intervals, and others to quantify the optimal adversarial strategies for double-spending and selfish mining.



Based on the developed MDP model, their analysis indicates that an adversary possessing 30% of the mining power can acquire an average of 209 block rewards by following their MDP mining strategy when 1000 blocks are mined by the entire network, whereas according to [10], the average yield is 205.8 block rewards. Moreover, they examined the influence of propagation parameters, stale block rates, and eclipse attacks on adversarial strategies across various cryptocurrency networks such as Bitcoin, Ethereum, Litecoin, and Dogecoin. Their analysis reveals that in order to achieve a security level equivalent to Bitcoin's, which utilizes 6 block confirmations, when facing an adversary possessing 30% of the total mining power Ethereum would require a minimum of 37 transaction confirmations, 28 for Litecoin, and 47 transaction confirmations Dogecoin.

(Include a table on the impact of various parameters on adversarial strategy)

Mitigating issue of requiring the knowledge of various blockchain parameters such as adversary computational power and communication capability of [9], authors of [12] proposed a model-free, reinforcement learning (RL) based approach which allows a RL agent to dynamically learn a mining strategy with performance approaching that of the optimal mining strategy. For their work they adopted the MDP mining model proposed in [9] with Q-Learning algorithm to derive the optimal mining strategy. In their research, they employed a refined version of the Q-Learning algorithm, termed the Multidimensional Q-Learning algorithm, owing to the nonlinearity inherent in the objective function (Equation 1). Leveraging this multidimensional Q-Learning algorithm, they successfully optimized the nonlinear objective function to attain the optimal mining strategy. However, this approach faced limitations in its applicability to a real blockchain environment due to two primary reasons. Firstly, it relied on the model outlined in [9], which lacked consideration for real-world blockchain parameters like stale block rates, eclipsed attacks, propagation parameters, among others. Even if they were to adopt the mining model proposed in [12], it would present significant challenges for the RL agent to learn an optimal mining strategy, primarily due to the vast state-action space. In their study, they employed a tabular Q-Learning algorithm, which would prove highly inefficient for handling such large state-action spaces. Indeed, this issue directly impacts the convergence of the algorithm. If the algorithm requires a substantial amount of time to discover the optimal policy, it becomes economically unviable for miners. This is because prolonged computation time translates to increased expenses for hardware and computing power, diminishing the economic feasibility of mining operations.

As delineated in Section I, within the Bitcoin ecosystem, every transaction is subject to validation by the nodes comprising the Bitcoin network before including in a publicly accessible distributed ledger, commonly referred to as the blockchain. Bitcoin blockchain is a decentralized, distributed ledger that records transactions across a network of computers in a secure and tamper-resistant manner. Each Bitcoin transaction is verified by a consensus mechanism among the network participants, and once validated, it is added to a chronological chain of blocks. As delineated in the study [18], malicious entities possess the capability to disrupt the synchronization of the ledger across multiple nodes, facilitating a form of attack known as the double-spending attack. This tactic involves redirecting previously validated transactions, thereby enabling the attackers to utilize the same coins twice. In this scheme, the attacker might initiate a transaction with a merchant and subsequently illicitly construct a longer chain of blocks excluding this transaction. By publishing their longer chain, they can prompt the ledger to undergo replacement, effectively nullifying the original transaction or redirecting the payment to another destination. In other words, in a double spending attack, a malicious actor spends the same cryptocurrency tokens twice by creating conflicting transactions in different parts of the network. This can be profitable if the attacker can control a significant portion of the network's hash rate. Furthermore, in the seminal paper by Satoshi Nakamoto [1], it is shown that if an attacker controls less than 50% of the computational power within the network, the likelihood of successful double-spend attacks diminishes exponentially over time. Typically, merchants and exchanges require multiple confirmations (blocks added after the transaction) to consider a transaction final and irreversible. The analysis presented in [18] indicates that the success of this attack is contingent not solely on the number of confirmations but also on the duration until the transaction is authorized. Given that blocks are typically generated approximately once every 10 minutes in Bitcoin, there exists a significant delay before a specific transaction is incorporated into the blockchain.   
While this form of attack is not inherently a mining attack, it can be enabled through manipulation of the mining process.

The 51% attack [17], [20], a form of mining attack wherein the attacker commands 51% of the computational power, or hashing power, of the entire network. This enables the attacker to mine blocks at a faster rate than other miners. In this scenario, the attacker initiates the creation of blocks privately, without disseminating them across the network, thereby maintaining its own version of the blockchain. Subsequently, the attacker reveals their private chain to the network at a later stage. Given that the attacker possesses more than half of the hashing power of the entire network, they can establish the longest chain by influencing network nodes to adopt their published chain [1]. This capability grants the attacker the opportunity to engage in double-spending. Specifically, if the attacker executes multiple transactions before revealing their private chain, and neglects to include those transactions in their own private chain, even though those transactions are confirmed and added to the public chain, the network will accept the attacker's private chain once it is published, superseding the public chain. Consequently, the attacker can exploit this situation to double-spend their coins. For instance, as outlined in [17], the GHash.IO pool temporarily commanded 54% of the total hashrate, surpassing the critical threshold of 51% widely recognized as the theoretical point for potential network vulnerability. Promptly, the community took remedial actions by redistributing mining resources to alternative pools. However, the underlying incentives driving the formation of sizable pools persist, thereby posing an ongoing risk of potential network disruption, should such concentrations of mining power arise again.

Another form of mining attack, known as Pool Hopping attack [23], involves miners strategically timing their participation in mining pools, alternately directing their computational resources towards the pool, and diverting them elsewhere. By doing so, these miners aim to gain rewards that exceed what would be considered equitable based on their proportional contribution to the pool's total computational power making other continuous miners to earn a lesser reward. The most recognized manifestation of pool-hopping occurs within pools employing the proportional method (Section 1), renowned for its simplicity, widespread adoption, and susceptibility to hopping tactics. As delineated in Section 1, the reward allocated for each share is denoted as B/N and the distribution of a block's reward among miners is proportional to the quantity of shares each miner has submitted since the preceding block (nB/N). Consequently, the value of a share submitted at any given moment is influenced by the total number of shares submitted since the last block was found. In other words, as the duration of a mining round extends, the value of each individual share becomes less worth. Consequently, it becomes advantageous for miners to strategically submit shares to the pool during shorter rounds and redirect their efforts elsewhere during longer rounds. This strategy is driven by the understanding that participating in a pool with a substantial accumulation of shares and no corresponding discovery of a block diminishes the expected reward, as it will be distributed among all contributing miners. Thus, there arises a point where it becomes financially advantageous to cease mining within such a pool and allocate resources elsewhere.

The authors of [24] present yet another mining attack called Difficulty Raising Attack wherein the attack involves adjusting the difficulty of their own chain. Here also the attacker operates a competitive blockchain characterized by blocks that lack correlation with the honest chain. Concurrently, the attacker manipulates the automatic difficulty adjustment mechanism within their clandestine chain to augment the likelihood of surpassing the public honest chain. Upon achieving this objective, the attacker discloses their secret chain.

**References**

[1] S. . Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," , . [Online]. Available: http://bitcoin.org/bitcoin.pdf. [Accessed 13 4 2024].

[2] A. M. Antonopoulos, Mastering Bitcoin: Unlocking Digital Crypto-Currencies, ed., vol. , , : O'Reilly Media, , p.

[3] Hattab, siham, & Taha Alyaseen, I. F. . (2019). Consensus Algorithms Blockchain: A comparative study. *International Journal on Perceptive and Cognitive Computing*, *5*(2), 66–71. <https://doi.org/10.31436/ijpcc.v5i2.103>

[4] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun, "A review on consensus algorithm of blockchain," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017, pp. 2567-2572, doi: 10.1109/SMC.2017.8123011.

[5] N. Chaudhry and M. M. Yousaf, "Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities," 2018 12th International Conference on Open Source Systems and Technologies (ICOSST), Lahore, Pakistan, 2018, pp. 54-63, doi: 10.1109/ICOSST.2018.8632190.

[6] 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.

[7] D. Kraft, “Difficulty control for blockchain-based consensus systems,” Peer-to-Peer Networking and Applications, vol. 9, no. 2, pp. 397–413, 2016.

[8] 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). Saarbrucken, Germany: IEEE, March 2016, pp. 305–320.

[9] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, “Optimal selfish mining strategies in bitcoin,” in International Conference on Financial Cryptography and Data Security. Springer, 2016, pp. 515–532

[10] Chen Feng, Jianyu Niu: Selfish Mining in Ethereum. ICDCS 2019: 1306-1316.

[11] Qianlan Bai, Xinyan Zhou, Xing Wang, Yuedong Xu, Xin Wang, Qingsheng Kong: A Deep Dive Into Blockchain Selfish Mining. ICC 2019: 1-6.

[12] A. Gervais, G. O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, ¨ and S. Capkun, “On the security and performance of proof of work blockchains,” in Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, 2016, pp. 3–16.

[13] Taotao Wang, Soung Chang Liew, Shengli Zhang: When Blockchain Meets AI: Optimal Mining Strategy Achieved By Machine Learning. CoRR abs/1911.12942 (2019).

[14] Martijn Bastiaan: Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin[C]//Availab le at http://referaat. cs. utwente. nl/conference/22/paper/7473/preventingthe-51-attack-a-stochasticanalysis-oftwo-phase-proof-of-work-in-bitcoin. pdf. 2015.

[15] Yonatan Sompolinsky, Aviv Zohar: Secure High-Rate Transaction Processing in Bitcoin. Financial Cryptography 2015: 507-527.

[16] Seb Neumayer, Mayank Varia, Ittay Eyal: An Analysis of Acceptance Policies For Blockchain Transactions. IACR Cryptology ePrint Archive 2018: 40 (2018).

[17] Martijn Bastiaan: Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin[C]//Availab le at http://referaat. cs. utwente. nl/conference/22/paper/7473/preventingthe-51-attack-astochasticanalysis-oftwo-phase-proof-of-work-in-bitcoin. pdf. 2015.

[18] Yonatan Sompolinsky, Aviv Zohar: Secure High-Rate Transaction Processing in Bitcoin. Financial Cryptography 2015: 507-527.

[19] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin’s peer-to-peer network. 2015.

[20] Sayeed, Sarwar, and Hector Marco-Gisbert. "Assessing blockchain consensus and security mechanisms against the 51% attack." Applied sciences 9.9 (2019): 1788.

[21] Seb Neumayer, Mayank Varia, Ittay Eyal: An Analysis of Acceptance Policies For Blockchain Transactions. IACR Cryptology ePrint Archive 2018: 40 (2018).

[22] Marianna Belotti, Sofiane Kirati, Stefano Secci. Bitcoin Pool-Hopping Detection. 2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI), Sep 2018, Palerme, Italy. ff10.1109/RTSI.2018.8548376ff. ffhal-02481221f

[23] Meni Rosenfeld: Analysis of Bitcoin Pooled Mining Reward Systems, Preprit Dec 2011, <http://arxiv.org/abs/1112.4980>

[24] Lear Bahack: Theoretical Bitcoin Attacks with less than Half of the Computational Power (draft), <http://eprint.iacr.org/2013/868>.

**Notes**

Last ref id: 16

1. **Grinding Attack**: Grinding attacks exploit vulnerabilities in mining algorithms to manipulate the probability of finding a valid block hash. By exploiting computational power strategically, attackers can increase their chances of mining blocks and earning rewards.
2. Finney Attack: Named after Bitcoin pioneer Hal Finney, this attack involves a miner pre-mining a high-value transaction and then quickly mining a block containing that transaction themselves. This allows them to spend their own coins immediately after the block is mined, potentially before the network detects the double spend.
3. Siphoning Attack: This attack involves a miner manipulating network traffic to siphon off a portion of block rewards intended for other miners. By diverting rewards to themselves, attackers can increase their profitability at the expense of other miners.

1. [Cryptocurrency Prices, Charts, and Crypto Market Cap | CoinGecko](https://www.coingecko.com/)

   [Cryptocurrency Ownership Data – Triple-A](https://triple-a.io/cryptocurrency-ownership-data/) [↑](#footnote-ref-1)
2. [Information and Communication Technology Agency of Sri Lanka (icta.lk)](https://www.icta.lk/media/blog/cryptocurrency) [↑](#footnote-ref-2)