Insightful Mining Equilibria

Mengqian Zhang[1], Yuhao Li[2], Jichen Li[3], Chaozhe Kong[3], and Xiaotie Deng[3(✉)]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
mengqian@sjtu.edu.cn

[2] Columbia University, New York, NY 10027, USA
yuhaoli@cs.columbia.edu

[3] Center on Frontiers of Computing Studies, School of Computer Science, Peking University, Beijing 100871, China
{limo923,kcz,xiaotie}@pku.edu.cn

Abstract. The selfish mining attack, arguably the most famous game-theoretic attack in blockchain, indicates that the Bitcoin protocol is not incentive-compatible. Most subsequent works mainly focus on strengthening the selfish mining strategy, thus enabling a single strategic agent more likely to deviate. In sharp contrast, little attention has been paid to the resistant behavior against the selfish mining attack, let alone further equilibrium analysis for miners and mining pools in the blockchain as a multi-agent system. In this paper, first, we propose a novel strategy called insightful mining to counteract the selfish mining attack. By infiltrating an undercover miner into the selfish pool, the insightful pool could acquire the number of its hidden blocks. We prove that, with this extra insight, the utility of the insightful pool is strictly greater than the selfish pool's when they have the same mining power. Then we investigate the mining game where all pools can choose to be honest or take the insightful mining strategy. We characterize the Nash equilibrium of such a game and derive three corollaries: (a) each mining game has a *pure* Nash equilibrium; (b) there are at most two insightful pools under some equilibrium no matter how the mining power is distributed; (c) honest mining is a Nash equilibrium if the largest mining pool has a fraction of mining power no more than 1/3. Our work explores, for the first time, the idea of spying in the selfish mining attack, which might shed new light on researchers in the field.

Keywords: Blockchain · Selfish mining · Markov process · Insightful mining · Mining game

This work was supported by Science and Technology Innovation 2030 - "New Generation Artificial Intelligence" Major Project No. 2018AAA0100901. This work has been performed with support from the Algorand Foundation Grants Program.
Y. Li—Supported by NSF grants CCF-1563155, CCF-1703925, IIS-1838154, CCF-2106429 and CCF-2107187.

K. A. Hansen et al. (Eds.): WINE 2022, LNCS 13778, pp. 21–37, 2022.
https://doi.org/10.1007/978-3-031-22832-2_2

1 Introduction

Bitcoin [16], as the pioneering blockchain ecosystem, proposes an electronic payment system without any trusted party. It creatively uses *Proof-of-Work* (PoW) to incentivize all *miners* to solve a cryptopuzzle (also known as *mining*). The winner will gain the record-keeping rights to generate a *block* and be awarded the newly minted tokens. As more and more computational power is invested into mining, it may take sole miner months or even years to find a block [24]. In order to reduce the uncertainty, a group of miners forms a *mining pool* to share their computational resources. Under the leadership of the pool manager, all miners in a pool solve the same puzzle in parallel and share the block rewards. In the Bitcoin system, so long as all participants behave honestly, one's expected revenue will be proportional to its hashing power.

However, in practice, miners are rational and may act strategically. Thus, game theory naturally stands out as a tool for analyzing the robustness of the Bitcoin protocol. The conventional wisdom would expect a proof of the incentive compatibility of the Bitcoin protocol and subsequently the strategyproofness against manipulative miners.

Such a hope was broken by the seminal work [6], which proposed the selfish mining strategy, arguably the most well-known game-theoretic attack in blockchain. It indicates that the Bitcoin mining protocol is not incentive-compatible. The key idea behind the attack is to induce honest miners to waste their mining power. As a result, the selfish pool could obtain more revenue than its fair share.

Pushing this approach to the extreme, [23] expanded the action space of selfish mining, modeled it as a Markov Decision Process (MDP), and pioneered a novel technique to resolve the non-linear objective function of the MDP to get a more powerful selfish mining strategy, for a revenue arbitrarily close to the optimum. A series of works have since been initiated to study the mining strategies of a rational pool under the same assumption that other pools behave honestly [7,11,14,17,18,21].

In sharp contrast, little attention has been paid to the incentive of other pools, which plays an important role in studying the strategic interactions among participants and understanding the stable state of blockchain systems. In this paper, we propose and study the following vital questions.

1. *Can a pool strategically defend against the selfish mining attack?*
2. *Moreover, what equilibrium will the ecosystem of different types of agents eventually reach?*

1.1 Our Contributions

In this work, we propose a strategy called *insightful mining* (Fig. 1). Once detecting a selfish pool, an *insightful pool* that adopts the insightful mining strategy can infiltrate an undercover miner into it to monitor the number of hidden blocks.[1]

[1] We discuss this action in more detail in Sect. 3.1.

With this key information, the insightful pool clearly knows the real-time state of the mining competition and thus responds strategically. From a high-level view, when observing that the selfish pool is taking the lead, the insightful pool would behave honestly to end its leading advantage as quickly as possible. On the other hand, when the insightful pool is taking the lead, it will take action similar to selfish mining, regarding the selfish pool and the honest pool as "others". Note that by infiltrating spies, a strategic player can gain more information (*e.g.*, the hash values of hidden blocks) than the length of the private branch. With this information, there are lots of things that a player could do. This paper, however, focuses on the insightful mining strategy, which only utilizes the number of hidden blocks.

Although using very little information, the strategy firmly answers our first research question: A pool can strategically defend against the selfish mining attack with the insightful mining strategy. Specifically, the system consists of three types of players: the honest pool, the selfish pool, and the insightful pool. With different mining strategies, the three players may hold different branches and have asymmetric information during the mining competition. The honest pool, following the protocol, has the public information (*i.e.*, the length of its public branch). The selfish pool keeps a selfish branch and is aware of the length of the public branch and its selfish branch. Owing to the infiltrated spy, the insightful pool learns all information (in particular, the length of the honest branch, the selfish branch, and its insightful branch). We model their interactions as a two-dimensional Markov reward process with an infinite number of states (Table 1 and Fig. 2). We prove that when there is a selfish pool and an insightful pool with the same mining power, the insightful pool will get a strictly greater expected revenue than the selfish pool (Theorem 1). This demonstrates that the extra insight significantly reverses the selfish pool's advantage.

Then we investigate the scene where all n mining pools are strategic. Besides counteracting the selfish mining attack, insightful mining can be adopted directly as a mining strategy. Specifically, insightful mining resembles selfish mining if there is no pool mining selfishly. We study the mining game where each pool plants spies into all other pools and chooses either to follow the Bitcoin protocol or to take the insightful mining strategy. Such a mining game can be formulated as an n-player normal-form game. Note that although there are 2^n pure strategy profiles, the payoff function of each player is explicitly represented (Proposition 1). Our main result is a characterization theorem of the Nash equilibrium in mining games (Theorem 2). Concretely, Theorem 2 derives three corollaries: (a) each mining game has a *pure* Nash equilibrium; (b) there are at most two insightful pools under some equilibrium no matter how the mining power is distributed; (c) honest mining is a Nash equilibrium if the largest mining pool has a fraction of total hashing power no more than 1/3. These corollaries are surprising. Taking (a) as an example, there is no guarantee of the existence of pure Nash equilibria in general.

Beyond our theoretical results, we also conduct several simulations to understand insightful mining (Sect. 5). First, we visualize the relative revenue of the

selfish pool and the insightful pool when they have the same mining power. An interesting observation is that when their hashing power is larger than 1/3, the insightful pool can gain most of the revenue. Besides, we explore the performance of the insightful mining strategy when they have different mining power. Simulation results provide compelling evidence that the insightful pool could still gain more revenue even if it holds less mining power than the selfish pool.

In the end, we discuss the role of the undercover miner in the context of selfish mining and blockchain, which sheds new light on future research directions (Sect. 6).

1.2 Related Work

The classic selfish mining attack was first proposed and mathematically modeled as a Markov reward process in the seminal paper [6]. Observing that the classic selfish mining strategy could be suboptimal for a large parameter space, several works [17,23] further generalized the system as a Markov Decision Process (MDP) to find the optimal selfish mining strategy. Aiming to solve the average-MDP with a non-linear objective function, [23] proposed a binary search procedure by converting the problem into a series of standard MDPs. A recent work [28] developed a more efficient method called Probabilistic Termination Optimization, converting the average-MDP into only one standard MDP.

Studying other agents' incentives against one selfish miner was more challenging due to the tremendous state spaces and complicated Markov reward processes. The work of [15] presented some simulation results on systems involving multiple selfish miners [6] or involving multiple stubborn miners [17]. On the learning side, a recent work [12] proposed a novel framework called SquirRL, which is based on deep reinforcement learning (deep-RL) techniques. Their experiments suggest that adopting selfish mining might not be the optimal choice when facing selfish mining. We *prove* such a result by providing the insightful mining strategy and the dominating theorem (Theorem 1). The strength of SquirRL is a more general strategy space generated by deep-RL. However, we highlight that it cannot cover our insightful mining strategy since our greatest strength comes from our undercover miner's insights (information), which have not been discussed in the broad selfish mining context.

To our best knowledge, the most related work that theoretically studied the equilibria with multiple selfish mining pools is [4]. Due to the analytical challenges of infinite states in the classic selfish mining strategy, they proposed a simplified version called semi-selfish mining, where the strategic mining pool will only keep a private chain of the length of at most two. Such a restriction makes the Markov reward process have finite states (as long as there is a finite number of semi-selfish miners) and simplifies the equilibrium analysis. However, our insightful mining strategy works against the classic selfish mining strategy and may also keep an arbitrary long private chain. While this leads to a 2-dimensional Markov reward process with an infinite number of states, the techniques in the mathematical analysis are sufficient for us to prove the desired dominating theorem (Theorem 1) and equilibrium characterization (Theorem 2).

2 Preliminaries

2.1 Proof of Work

In the context of blockchain, Proof of Work was first introduced in Bitcoin [16]. As mentioned, the security of Bitcoin heavily relies on the Proof-of-Work scheme, which has also been widely adopted by other blockchain systems like Ethereum [2]. The past decade has seen a great amount of research around PoW, with respect to its block rewards design [3], strategic deviation [13], the difficulty adjustment algorithm [10,19], energy costs [8], and so on.

Taking Bitcoin as an example, PoW requires a miner to randomly engage in the hashing function calls to solve a cryptopuzzle. Typically, miners should search for a *nonce* value satisfying that

$$H(previous\ hash;\ address;\ Merkle\ root;\ nonce) \leq D \tag{1}$$

where $H(\cdot)$ is a commonly known cryptographic hash function (*e.g.*, SHA-256 in Bitcoin); *previous hash* is the hash value of the previous block; *address* is the miner's address to receive potential rewards; *Merkle root* is an integrated hash value of all transactions in the block; and D is the target of the problem and reflects the difficulty of this puzzle.[2] Started from the genesis block, all miners compete to find a feasible solution, thus generating a new block appended to the previous one. In return, they will be awarded the newly minted bitcoins for their efforts in maintaining the blockchain system. The standard Bitcoin protocol treats the longest chain as the main chain. Once encountering two blocks at the same block height, miners randomly choose one to follow according to the uniform tie-breaking rule. Thus, in order to be accepted by more miners, it is suggested to publish the newly generated block immediately. In this paper, the miners who stick to the Bitcoin protocol are referred to be *honest*.

2.2 Mining Pool

With more and more hashing power invested into mining, the chances of finding a block as a sole miner are quite slim. Nowadays, miners tend to participate in organizations called mining pools.

Generally, a mining pool comprises a pool manager and several peer miners. All participants shall cooperate to solve the same puzzle. Specifically, each miner will receive a task like (1) above from the pool manager and a work unit a work unit containing a particular range of nonce. Instead of trying all possible nonce values, the miner only needs to search for the answer from the received work unit. In this way, all miners in the pool work in parallel. Once any miner finds a valid solution, this pool succeeds in this mining competition. Then a new task will be organized and further released to all miners in the pool. Also, participants will share the mining rewards according to the reward allocation protocol like Pay

[2] For security, the difficulty of puzzles will be adjusted automatically to ensure that the mean interval of block generation is 10 min.

Per Share (PPS), proportional (PROP), Pay Per Last N Shares (PPLNS) [27], and so on. In expectation, the miners' rewards are proportional to their hashing power. As a result, miners who join the mining pool can significantly reduce the variance of mining rewards. Currently, most of the blocks in Bitcoin are generated by mining pools such as AntPool [1], Poolin [20], F2Pool [9].

2.3 Selfish Mining

It has long been believed that the Bitcoin protocol is incentive-compatible. However, Eyal and Sirer [6] indicate that it is not the case. It describes a well-known attack called selfish mining. A pool could receive higher rewards than its fair share via the selfish mining strategy. This attack ingeniously exploits the conflict-resolution rule of the Bitcoin protocol, in which when encountering a fork, only one chain of blocks will be considered valid. With the selfish mining strategy, the attacker deliberately creates a fork and forces honest miners to waste efforts on a stale branch. Specifically, the selfish pool strategically keeps its newly found block secret rather than publishing it immediately. Afterward, it continues to mine on the head of this private branch. When the honest miners generate a new block, the selfish pool will correspondingly publish one private block at the same height and thus create a fork. Once the selfish pool's leads reduce to two, an honest block will prompt the selfish pool to reveal all its private blocks. As a well-known conclusion, assuming that the honest miners apply the uniform tie-breaking rule, if the fraction of the selfish pool's mining power is greater than 25%, it will always get more benefit than behaving honestly.

3 Insightful Mining Strategy

3.1 Model and Strategy

This paper considers a system of n miners. Each miner i has m_i fraction of total hashing power, such that $\sum_{i=1}^{n} m_i = 1$. Let $\mathcal{H}$, $\mathcal{S}$, $\mathcal{I}$ denote the set of honest miners, selfish miners, and insightful miners, respectively. As the honest miners strictly follow the Bitcoin protocol and do not hide any block information from each other, they are regarded as a whole, referred to as the *honest pool* in the paper. Similarly, all selfish miners who adopt the selfish mining strategy combine together to behave as a single agent, which is called the *selfish pool*. The remaining miners form the *insightful pool* and adopt the insightful strategy stated later. Let α and β denote the fraction of mining power controlled by the selfish pool and the insightful pool, respectively. We have $\alpha = \sum_{i \in \mathcal{S}} m_i$ and $\beta = \sum_{i \in \mathcal{I}} m_i$. Then the total power of the honest pool can be represented as $1 - \alpha - \beta$. Following the previous work [6,23], in this paper, we also assume that the time to broadcast a block is negligible and the transaction fee is negligible. In other words, the pools' revenue mainly comes from block rewards. In addition, the block generation is treated as a randomized model, where a new block is generated in each time slot.

Now we describe the insightful mining strategy. Before getting into the details, we state that the insightful pool could learn how many blocks the selfish pool has been hiding by doing the following. The manager of the insightful pool shall pretend to join the selfish pool as a spy. As a pool member, it will receive a mining task from the manager of the selfish pool. The hash value of the previous block can be parsed from the task. Normally, this hash value corresponds to the last block of the main chain. Once the selfish pool mines a block,[3] its manager will keep the block private and publish a new task based on it. From the spy's perspective, there is no newly published block in the system, but the selfish manager releases a new task based on an unknown block. Then it is reasonable to believe that the selfish manager is hiding blocks. Furthermore, the number of hidden blocks is exactly the number of recently received tasks with unmatched *previous hash*.

By working as a spy,[4] the insightful pool has a clear understanding of the system's situation, *i.e.*, the mining progress of each player. Although all pools are mining after the main chain, the three players may hold different sub-chain (also referred to as *branch*) during the mining competition. Let l_h, l_s, l_i denote the length of honest branch, selfish branch, and insightful branch respectively. In the process of mining, the honest pool only knows the public information l_h. The selfish pool is aware of both l_h and l_s, while the insightful pool can observe all three lengths. Then the three types of players compete to generate blocks based on their own information. Their competition works in rounds. Each round begins with a global consensus on the current longest chain. When the selfish pool and insightful pool reveal all their private blocks, or they have no hidden blocks while the honest pool finds a block (see *Case 1* below), the round ends, leading to a new global consensus. For the first block in a round, there are three possible cases.

Case 1: the honest pool generates the first block. With probability $1 - \alpha - \beta$, the honest pool mines a block and broadcasts it immediately. In this case, the insightful pool accepts this newly generated block and mines after it. According to the selfish mining strategy, the selfish pool will do the same. Consequently, all players reach a consensus in this case and compete for the next block.

Case 2: the selfish pool generates the first block. With probability α, the selfish pool mines a block. Based on the selfish mining strategy, the selfish pool will keep it private, aiming to further extend its lead. After observing this situation through the spy in the selfish pool, the insightful pool behaves honestly until the selfish pool reveals all its hidden blocks. Recall that when facing two branches of the same height, the honest pool chooses one of them uniformly. The insightful pool, however, will deterministically mine on the opposite of the selfish branch.

[3] A member of the selfish pool finds an acceptable nonce to the cryptopuzzle and submits it to the manager.

[4] We assume that the mining power of this spy is negligible, as well as its revenue from the selfish pool.

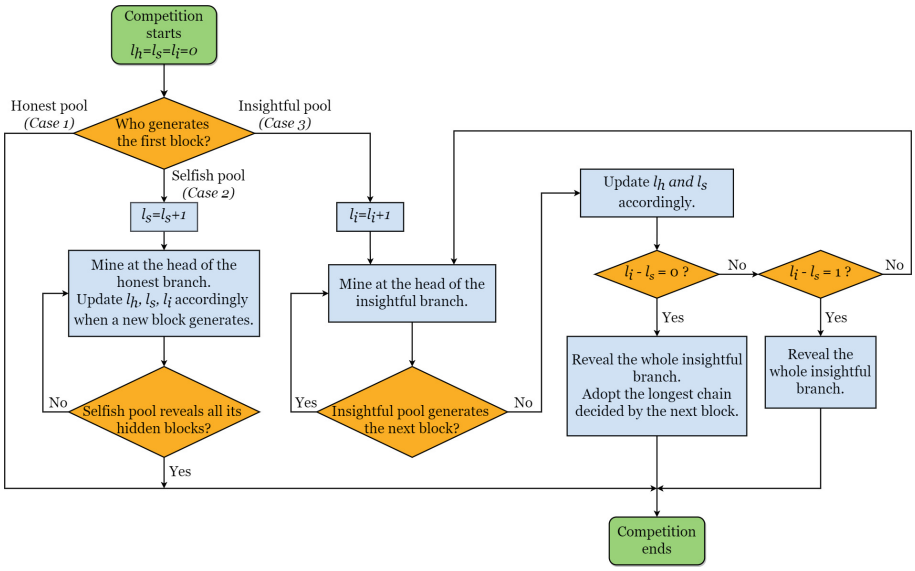

Fig. 1. Flow chart of the insightful mining strategy. l_h, l_s and l_i are the length of the honest branch, selfish branch, and insightful branch, respectively.

The key insight behind this strategy is to prompt[5] the selfish pool to reveal all its hidden blocks and end its leading advantage as quickly as possible.

Case 3: the insightful pool generates the first block. With probability β, the insightful pool mines a block. It hides this block and takes the following actions, which are similar to selfish mining. The insightful pool keeps a watchful eye on how many blocks the selfish pool and the honest pool have mined respectively. In the following competition, when its lead is larger than one (*i.e.*, $l_i - \max\{l_h, l_s\} > 1$), the insightful pool always hides all its mined blocks. Otherwise, it reveals the private branch all at once. Here, the way of releasing blocks is different from selfish mining, which reveals blocks one by one in response to honest behavior.

The above three cases complete the description of our insightful mining strategy. We also show the flow chart of the strategy in Fig. 1. We emphasize that even if there is no selfish pool, the insightful mining could also work as an independent strategy, where *Case 2* never appears.

Remark 1. Note that with different strategies, players in the system have asymmetric information. Each of them can be characterized by the depth of their strategic thought, which forms a hierarchy of levels of iterated rationality.

[5] The meaning of "prompt" is that by generating blocks on the opposing branch, the selfish pool will be encouraged to reveal its hidden blocks one by one. Note that in this process, the selfish pool does not know the insightful pool exists, which is critical to the strategy design in game theory.

- Level zero. The honest pool, as the naive level-0 player, truthfully follows the protocol and has the public information (*i.e.*, l_h).
- Level one. The selfish pool, as the level-1 player, acts on the belief that other players are level-0 players. It adopts the selfish mining strategy, keeps a selfish branch, and is aware of l_h and l_s.
- Level two. Due to the infiltrated spy, the insightful pool works as a more sophisticated level-2 player. It observes that the population consists of both level-0 and level-1 players, learns all information (*i.e.*, l_h, l_s, and l_i), and adopts the insightful mining strategy.

Next, we will discuss the revenue of the three types of players with different levels of cognition. The scenario where all players are at the same cognitive level will be explored in Sect. 4.

3.2 Markov Reward Process

To analyze the relative revenue of different players under the insightful mining strategy, we use a two-dimensional state $s = (x, y)$ to reflect the system status and further model the mining events as a Markov Reward Process. The state x denotes the selfish pool's lead over the honest pool, *i.e.*, the number of blocks that the selfish pool has not revealed. Similarly, y is the insightful pool's lead over the selfish pool. Thus, we have $x, y \in \mathbb{N} \cup \{0'\}$ ($0'$ will be explained soon). Here, zero means the selfish pool (corresponding to x) or the insightful pool (corresponding to y) has no hidden blocks. Specifically, it contains two different states, which we use 0 and $0'$ to distinguish. Take x as an example. The state $x = 0$ indicates that the honest pool and the selfish pool are in agreement about a public chain. In other words, their branches are exactly the same. The state $x = 0'$ means that the selfish pool and others (the honest pool or the insightful pool) hold a separate branch of the same length, and the selfish pool has revealed all blocks on its branch. In the state of $0'$, the next block will break the tie and decides the longest chain. For y, the meanings of state 0 and $0'$ are similar to the above, with the insightful pool and others (the selfish pool and the honest pool) as two players.

Let $Pr[s, \tilde{s}]$ denote the probability of changing from state s to state $\tilde{s}$. The vector $r[s, \tilde{s}]$ represents the expected reward obtained from this state transition. It contains three components corresponding to the revenue of the honest pool, the selfish pool, and the insightful pool, respectively. With the help of these notations, Table 1 lists the detailed state transitions and corresponding revenues in the system. Specifically, the item (1) formalizes Case 1 in Sect. 3.1. Items (2)-(9) correspond to Case 2, and Case 3 contains items (10)–(24). The detailed analysis of each transition can be found in [26]. Figure 2 illustrates the overall state transitions in a more intuitive way. We denote the Markov Reward Process of Fig. 2 by $Markov(\alpha, \beta)$.

Recall that a branch will win at the end of one round. It is easy to verify that in our design, each block of the final winning branch will be awarded to some player once and only once.

Table 1. The state transitions and corresponding revenues.

No.	State s	State $\tilde{s}$	$Pr\,[s, \tilde{s}]$	$r\,[s, \tilde{s}]$	Conditions
1	(0,0)	(0,0)	$1-\alpha-\beta$	$(1,0,0)$	
2	(0,0)	(1,0)	α	$(0,0,0)$	
3	(1,0)	$(0',0)$	$1-\alpha-\beta$	$(0,0,0)$	
4	$(0',0)$	(0,0)	1	$(\frac{3-3\alpha-\beta}{2}, \frac{1+3\alpha-\beta}{2}, \beta)$	
5	(1,0)	$(1,0')$	β	$(0,0,0)$	
6	$(1,0')$	(0,0)	1	$(1-\alpha-\beta, \frac{1+3\alpha-\beta}{2}, \frac{1-\alpha+3\beta}{2})$	
7	$(x,0)$	$(x+1,0)$	α	(0,0,0)	$\forall x \geq 1$
8	(2,0)	(0,0)	$1-\alpha$	(0,2,0)	
9	$(x,0)$	$(x-1,0)$	$1-\alpha$	(0,1,0)	$\forall x \geq 3$
10	(0,0)	(0,1)	β	(0,0,0)	
11	(0,1)	$(1,0')$	α	(0,0,0)	
12	(0,1)	$(0,0')$	$1-\alpha-\beta$	(0,0,0)	
13	$(0,0')$	(0,0)	1	$(\frac{3-2\alpha-3\beta}{2}, \alpha, \frac{1+3\beta}{2})$	
14	(0,1)	(0,2)	β	(0,0,0)	
15	(0,2)	(0,0)	$1-\beta$	(0,0,2)	
16	(x,y)	$(x,y+1)$	β	(0,0,0)	$\forall x \in \{0'\} \bigcup \mathbb{N},\ y \geq 2$
17	$(0,y)$	$(0,y-1)$	$1-\alpha-\beta$	(0,0,1)	$\forall y \geq 3$
18	(x,y)	$(x+1,y-1)$	α	(0,0,1)	$\forall x \geq 0,\ y \geq 3$
19	$(1,y)$	$(0',y)$	$1-\alpha-\beta$	(0,0,0)	$\forall y \geq 2$
20	$(2,y)$	$(0,y)$	$1-\alpha-\beta$	(0,0,0)	$\forall y \geq 2$
21	(x,y)	$(x-1,y)$	$1-\alpha-\beta$	(0,0,0)	$\forall 3 \geq 2,\ y \geq 2$
22	$(x,2)$	(0,0)	α	(0,0,2)	$\forall x \geq 1$
23	$(0',2)$	(0,0)	$1-\beta$	(0,0,2)	
24	$(0',y)$	$(0,y-1)$	$1-\beta$	(0,0,1)	$\forall y \geq 3$

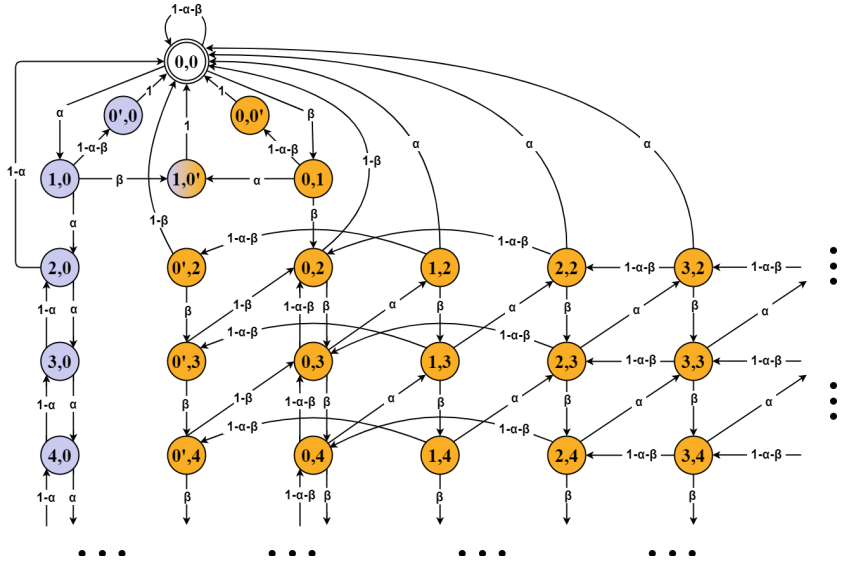

Fig. 2. The Markov Process of the system under the insightful mining strategy.

3.3 The Dominating Theorem

Let $M := \{H, IM, SM\}$. The utility of each $i \in M$ is the relative revenue (denoted by $RREV_i$) defined as follows:

$$\mathbb{E}\left[\liminf_{T\to\infty} \frac{\sum_{t=1}^{T} r_i[s_t, s_{t+1}] \mid s_0 = (0,0), s_{t+1} \sim Pr[s_t, s_{t+1}]}{\sum_{t=1}^{T}\sum_{j\in M} r_j[s_t, s_{t+1}] \mid s_0 = (0,0), s_{t+1} \sim Pr[s_t, s_{t+1}]}\right].$$

Note that the transition probability $Pr[s_t, s_{t+1}]$, $r_i[s_t, s_{t+1}]$, and $RREV_i$ ($\forall i \in M$) should depend on the mining power α and β. Here we simplify the notation without ambiguity.

Like previous work [12], here we focus on the scenario where the selfish pool and the insightful pool have the same mining power. The following theorem asserts that, in this case, the expected revenue of the insightful pool is strictly greater than the expected revenue of the selfish pool. The scenario with different pool sizes (*i.e.*, $\alpha \neq \beta$) will be explored in Sect. 5.

Theorem 1. *Let α and β be the fraction of mining power that the selfish pool and the insightful pool control, respectively. When $0 < \alpha = \beta < \frac{1}{2}$, $RREV_{SM}(\alpha, \beta) < RREV_{IM}(\alpha, \beta)$ holds.*

Here, we give the intuition why Theorem 1 holds, and the formal proof can be found in [26]. First, when the selfish pool takes the lead (*Case 2*), the insightful pool cooperates with the honest pool as a whole. However, when the insightful pool is taking the lead (*Case 3*), the selfish pool still competes with the honest pool (*i.e.*, inducing it to waste the mining power on a stale branch), which causes their internal friction. The second intuition is that, when facing two branches with the same length (one is the honest branch and the other is the selfish branch), the insightful pool can clearly know the selfish pool's branch and play against it. Conversely, when confronted with an honest branch and an insightful branch of the same length, the selfish pool will uniformly choose one of them. These two reasons enable the insightful pool to get more revenue than the selfish pool when both have the same mining power.

4 The Mining Game and Equilibria

In this section, we consider the scenario where all n mining pools are strategic and study its Nash equilibrium. Specifically, during the competitive interaction, the honest and selfish pool may realize the existence of insightful mining and learn to do the same, where all players are at the same level of recognition. It is worth noting that insightful mining is a well-defined strategy and can be adopted directly. If there is no selfish pool in the system, insightful mining will look the same as selfish mining. Then we consider the scenario where each pool can choose to follow the Bitcoin protocol truthfully or take the insightful mining strategy. We formally define its strategy space in Sect. 4.1, analyze the utility functions in Sect. 4.2, and characterize the Nash equilibrium in Sect. 4.3.

4.1 Strategy Space

There are n mining pools, and we denote by $[n] \coloneqq \{1, \cdots, n\}$. The fraction of their hashing power is denoted by $\{m_1, \cdots, m_n\}$ and we have $\sum_{i=1}^{n} m_i = 1$. Each pool i will infiltrate undercover miners into all other pools to monitor their real-time state, namely, whether a certain pool is mining selfishly and, if any, how many blocks are hidden. As a result, each pool i could adopt the insightful mining strategy.

In the mining game, each pool has two strategies: *refined honest mining* and *insightful mining*, denoted by *RHonest* and *Insightful* respectively. The insightful mining strategy is exactly the same as we proposed before, while the refined honest mining is a slightly modified version of the standard mining strategy. Specifically, refined honest mining requires the pool to mine after the longest public chain and to publish its newly-generated block immediately. If someone hides the block, each pool could detect it through the spy therein. Then when facing two branches of the same length, the pool adopting *RHonest* shall clearly follow the honest branch instead of choosing one of them uniformly.

It is important to note that, in this mining game, at most one player is hiding blocks at any time. This is because once an insightful pool mines the first block and hides it, each other pool adopting no matter *RHonest* or *Insightful* will play against it until this mining competition ends. This makes the following analysis of the expected reward function fairly clean and enables us to complete the equilibrium analysis.

4.2 Expected Reward Functions

This section gives the formula of the expected reward function $ER_i(x_1, \cdots, x_n)$ of each pool i under the pure strategy profile $(x_1, \cdots, x_n) \in \{RHonest, Insightful\}^n$.

Proposition 1. *For an n-player mining game $(m_1, \cdots, m_n)$, let $(x_1, \cdots, x_n)$ be a (pure) strategy profile. Let c be a value depending on $(m_1, \cdots, m_n)$ and $(x_1, \cdots, x_n)$.*[6] *Let $Q \subseteq [n]$ be the set of pools that adopt $Insightful$ strategy. Then we have*

$$ER_i(x_1, \cdots, x_n) = \begin{cases} c \cdot \left(f(m_i) + m_i \cdot \sum_{j \in Q} 2m_j(1 - m_j) \right), & i \in Q; \\ c \cdot \left(m_i + m_i \cdot \sum_{j \in Q} 2m_j(1 - m_j) \right), & i \notin Q, \end{cases} \tag{2}$$

where $f(y) \coloneqq y^2 \cdot (2 - 3y)/(1 - 2y)$.

The proof of Proposition 1 can be found in [26].

[6] We note that c will not affect the calculation of a pool's relative revenue in the subsequent section.

4.3 Equilibria Characterization

The following theorem characterizes the pure Nash equilibria of the mining game. We refer readers to [26] for the proofs.

Theorem 2. *For an n-player mining game* $(m_1, \cdots, m_n)$ *with* $m_1 \geq \cdots \geq m_n$, *there are three types of pure Nash equilibrium* $(x_1, \cdots, x_n)$, *where*

(1) $(x_1 = \cdots x_n = RHonest)$ *is a Nash equilibrium if and only if* $m_1 \leq 1/3$;
(2) $(x_1 = Insightful, x_2 = \cdots x_n = RHonest)$ *is a Nash equilibrium if and only if* $m_1 \geq 1/3$ *and* $m_2 \leq g(m_1)$;
(3) $(x_1 = x_2 = Insightful, x_3 = \cdots x_n = RHonest)$ *is a Nash equilibrium if and only if* $m_1 \geq 1/3$ *and* $m_2 \geq g(m_1)$,

where $g(y) := \frac{-y^3 + 2y^2 + y - 1}{2y^2 + 4y - 3}$.

Remark 2 (Interpretation of two thresholds in Theorem 2). The analysis of Theorem 2 (1) is to consider the case where one player (say player 1) is deciding to choose *RHonest* or *Insightful* while all other players are adopting *RHonest*. Note that when such a player is adopting *Insightful*, it is the only one that may hide some blocks, and whenever it hides blocks, all other pools will play against it. This case corresponds to the $\gamma = 0$ case of the seminal work [6],[7] where they also got a 1/3 threshold (see Observation 1 in [6]). However, the cases of Theorem 2 (2) and (3) are much more interesting since there exists more than one strategic player with complicated (but explicit) utility functions. For the $g(\cdot)$ function, we note that the threshold $g(m_1) < 1/3$ whenever $m_1 > 1/3$. The interpretation is from the following observation: When player 1 behaves honestly, player 2's relative revenue is exactly proportional to its hashing power (say m_2). But when player 1 adopts *Insightful* ($m_1 \geq 1/3$ by Theorem 2 (1)), the relative revenue of player 2 is lower than m_2 (see proof in [26] for the specific revenue function). Hence, player 2 is more likely to deviate from *RHonest* if someone else (*i.e.*, player 1 here) has been behaving strategically. As a result, the threshold for player 2 to adopt *Insightful* is also lower than (the original) 1/3, and the exact bound is $g(m_1)$.

Theorem 2 has the following three corollaries.

Corollary 1. *Every n-player mining game* $(m_1, \cdots, m_n)$ *has a pure Nash equilibrium.*

Corollary 2. *For an n-player mining game* $(m_1, \cdots, m_n)$, *(RHonest,* $\cdots$, *RHonest) is a Nash equilibrium if* $m_1 \leq 1/3$.

Corollary 3. *For every n-player mining game* $(m_1, \cdots, m_n)$, *there is an equilibrium with at most two insightful pools.*

[7] In [6], γ denotes the ratio of honest miners that choose to mine on the private block when facing two branches with the same length.

5 Simulation

This section conducts several simulations to evaluate the effectiveness of the insightful mining strategy. Three agents are considered: the honest pool, the selfish pool, and the insightful pool. Their interactions are simulated as a discrete-time random walk process. In each step, one of the pools generates a block with a probability proportional to its hashing power, and others respond according to their strategies. The simulation ends after 2e9 steps. Then we calculate each pool's relative revenue during the process, which is defined as the proportion of blocks it generates on the main chain to the total number of blocks therein.

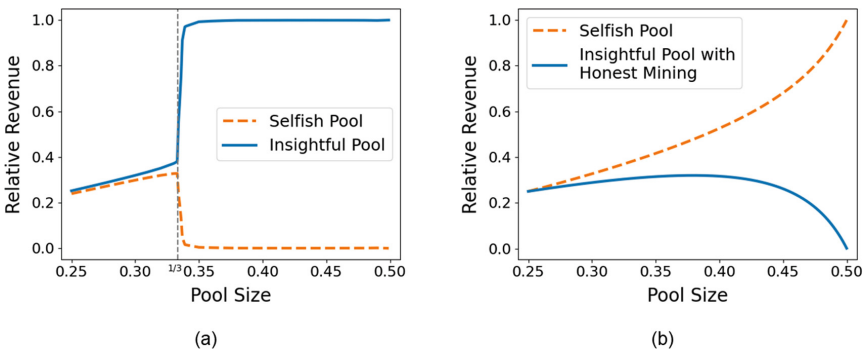

Fig. 3. Relative revenue of the selfish pool and the insightful pool with the same mining power. The selfish pool adopts the selfish mining strategy. (a) The insightful pool adopts the insightful mining strategy. (b) The insightful pool mines honestly.

Recall that α and β are the fractions of hashing power of the selfish pool and the insightful pool, respectively. First, we focus on the scenario where the insightful pool and the selfish pool have the same hashing power, *i.e.*, $\alpha = \beta$. Figure 3(a) visualizes the relative revenue of the insightful pool and the selfish pool when their hashing power belongs to $(0.25, 0.5)$. As can be seen, the insightful pool can always gain more revenue than the selfish pool. It is exactly consistent with our theoretical result in Theorem 1. Surprisingly, if their hashing power is larger than $1/3$ (*i.e.*, $\alpha = \beta > 1/3$), the insightful pool can gain most of the revenue. For a clear comparison, we also show their relative revenue under the circumstance that the insightful pool mines honestly in Fig. 3(b). As mentioned in the Introduction, the insightful pool suffers heavy losses in this case, which grow rapidly with the pool size increasing. Comparing Fig. 3(a) and 3(b) shows that the insightful mining strategy dramatically helps the pool turn things around when facing selfish mining.

Then we explore the scenario where $\alpha > \beta$, to consider whether less hashing power can also enable the insightful pool to earn more. Here, two definitions of "more revenue" are studied. One is the aforementioned relative revenue, which corresponds to the dashed line in Fig. 4. It demonstrates the threshold above

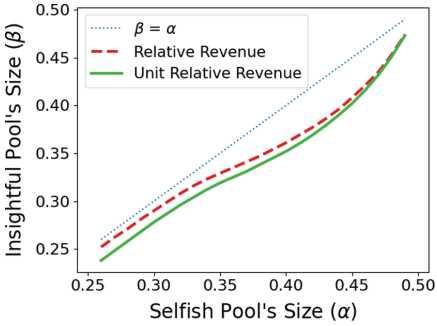

Fig. 4. Threshold of the insightful pool's size, above which it could obtain more relative revenue or unit relative revenue than the selfish pool.

which $RREV_{IM}(\alpha, \beta) > RREV_{SM}(\alpha, \beta)$. The other is the unit relative revenue. The solid line in Fig. 4 represents the corresponding threshold, above which we have $\frac{RREV_{IM}(\alpha,\beta)}{\beta} > \frac{RREV_{SM}(\alpha,\beta)}{\alpha}$. This curve is below the former. Both curves have similar trends, and they are all below the line of $\beta = \alpha$. It provides compelling evidence that with the insightful mining strategy, less computing power can also yield more revenue.

6 Discussion

In blockchain, the action of planting a spy in the pool has been deeply discussed in the context of Block Withholding Attack [5,22]. In such an attack, the attacker infiltrates miners into opponent pools to reduce their revenue. The undercover miner sends only partial solutions (*i.e.*, proofs of work) to the pool manager to share rewards. If it luckily finds a full solution which means a valid block, the undercover miner will discard the full proof of work directly, causing a loss to the victim pool. Our work explores, for the first time, the idea of spying in the selfish mining attack. It will shed new light on the researchers in the field.

Infiltrating spies dramatically expands the action spaces that a pool can take to counteract the selfish mining attack. Besides insightful mining, other strategies are worth exploring. Here, we roughly describe a potential idea. Recalling that the spy can actually extract the hash value of the latest hidden block from the new task issued by the pool manager. With this information, other pools can mine directly behind the latest block, although its full contents are not yet known.[8] By this strategy, all pools could follow the longest chain, which makes selfish mining ineffective. In other words, keeping the block secret for the

[8] Such an idea was discussed in [25]. In that context, the strategic miner mines on a newly generated block directly even before it is validated. To avoid potential conflict, the miner can choose to embed no transaction in the block being mined and just try to win the potential block rewards. Our discussion mainly focuses on the role of spies against the selfish mining attack.

selfish pool is equivalent to revealing it honestly, which extremely benefits the blockchain system. Nevertheless, such a strategy might not be the best choice for strategic mining pools. Further research should be undertaken to investigate the optimal mining strategy. It is also worthwhile to extend the action of planting spies to other blockchain scenarios.

Back to our work, insightful mining tells us that insight brings more revenue to a pool. It would be interesting to study the interactions between the insightful mining strategy and other strategies or protocols.

Acknowledgements. We would like to thank anonymous reviewers, Hongyin Chen, Yurong Chen, Zhaohua Chen, Zhijian Duan, Wenhan Huang, Hanyu Li, Kai Li, Qian Wang, and Xiang Yan for helpful comments on an earlier draft.

References

1. Antpool: World leading BTC mining pool (2014). https://antpool.com/. Accessed 20 Oct 2022
2. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. White Paper. **3**(37), 1–2 (2014)
3. Chen, X., Papadimitriou, C., Roughgarden, T.: An axiomatic approach to block rewards. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, pp. 124–131 (2019)
4. Cossío, F.J.M., Brigham, E., Sela, B., Katz, J.: Competing (semi-)selfish miners in bitcoin. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, AFT 2019, Zurich, Switzerland, 21–23 October 2019, pp. 89–109. ACM (2019)
5. Eyal, I.: The miner's dilemma. In: 2015 IEEE Symposium on Security and Privacy, pp. 89–103. IEEE (2015)
6. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: Christin, N., Safavi-Naini, R., (eds.) International Conference on Financial Cryptography and Data Security. LNAI, vol. 7151, pp. 436–454. Springer, Cham (2014). https://www.springerprofessional.de/en/majority-is-not-enough-bitcoin-mining-is-vulnerable/4391572
7. Feng, C., Niu, J.: Selfish mining in Ethereum. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 1306–1316. IEEE (2019)
8. Fiat, A., Karlin, A., Koutsoupias, E., Papadimitriou, C.: Energy equilibria in proof-of-work mining. In: Proceedings of the 2019 ACM Conference on Economics and Computation, pp. 489–502 (2019)
9. Gencer, A.E., Basu, S., Eyal, I., van Renesse, R., Sirer, E.G.: Decentralization in bitcoin and Ethereum networks. In: Meiklejohn, S., Sako, K. (eds.) FC 2018. LNCS, vol. 10957, pp. 439–457. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-58387-6_24
10. Goren, G., Spiegelman, A.: Mind the mining. In: Proceedings of the 2019 ACM Conference on Economics and Computation, pp. 475–487 (2019)
11. Grunspan, C., Perez-Marco, R.: Selfish mining in Ethereum. In: Pardalos, P., Kotsireas, I., Guo, Y., Knottenbelt, W. (eds.) Mathematical Research for Blockchain Economy. SPBE, pp. 65–90. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53356-4_5

12. Hou, C., et al.: SquirRL: automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning. In: 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, 21–25 February 2021. The Internet Society (2021)
13. Kiayias, A., Koutsoupias, E., Kyropoulou, M., Tselekounis, Y.: Blockchain mining games. In: Proceedings of the 2016 ACM Conference on Economics and Computation, EC 2016, Maastricht, The Netherlands, 24–28 July 2016, pp. 365–382. ACM (2016)
14. Li, Q., Chang, Y., Wu, X., Zhang, G.: A new theoretical framework of pyramid Markov processes for blockchain selfish mining. J. Syst. Sci. Syst. Eng. **30**(6), 667–711 (2021)
15. Liu, H., Ruan, N., Du, R., Jia, W.: On the strategy and behavior of bitcoin mining with n-attackers. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security, pp. 357–368 (2018)
16. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Decentralized Bus. Rev. 21260 (2008)
17. Nayak, K., Kumar, S., Miller, A., Shi, E.: Stubborn mining: generalizing selfish mining and combining with an eclipse attack. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 305–320. IEEE (2016)
18. Negy, K.A., Rizun, P.R., Sirer, E.G.: Selfish mining re-examined. In: Bonneau, J., Heninger, N. (eds.) FC 2020. LNCS, vol. 12059, pp. 61–78. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51280-4_5
19. Noda, S., Okumura, K., Hashimoto, Y.: An economic analysis of difficulty adjustment algorithms in proof-of-work blockchain systems. In: Proceedings of the 21st ACM Conference on Economics and Computation, pp. 611–611 (2020)
20. Poolin: A great bitcoin and multi-cryptocurrency mining pool (2017). https://www.poolin.com/. Accessed 20 Oct 2022
21. Ritz, F., Zugenmaier, A.: The impact of uncle rewards on selfish mining in Ethereum. In: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 50–57. IEEE (2018)
22. Rosenfeld, M.: Analysis of bitcoin pooled mining reward systems. arXiv preprint arXiv:1112.4980 (2011)
23. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in bitcoin. In: Grossklags, J., Preneel, B. (eds.) FC 2016. LNCS, vol. 9603, pp. 515–532. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54970-4_30
24. Schrijvers, O., Bonneau, J., Boneh, D., Roughgarden, T.: Incentive compatibility of bitcoin mining pool reward functions. In: Grossklags, J., Preneel, B. (eds.) FC 2016. LNCS, vol. 9603, pp. 477–498. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54970-4_28
25. Sompolinsky, Y., Zohar, A.: Bitcoin's underlying incentives. Commun. ACM **61**(3), 46–53 (2018)
26. Zhang, M., Li, Y., Li, J., Kong, C., Deng, X.: Insightful mining equilibria. arXiv preprint arXiv:2202.08466 (2022)
27. Zolotavkin, Y., García, J., Rudolph, C.: Incentive compatibility of pay per last n shares in bitcoin mining pools. In: Rass, S., An, B., Kiekintveld, C., Fang, F., Schauer, S. (eds) Decision and Game Theory for Security. GameSec 2017. LNCS, vol. 10575, pp. 21–39. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68711-7_2
28. Zur, R.B., Eyal, I., Tamar, A.: Efficient MDP analysis for selfish-mining in blockchains. In: Proceedings of the 2nd ACM Conference on Advances in Financial Technologies, pp. 113–131 (2020)