# Game Theory in Incentive Mechanism of Blockchain Security

Team Number: 14
Author: Junhao Dai June 10, 2024

## Abstract

We delves into the incentive mechanisms within blockchain security, with a particular focus on the application of game theory in Bitcoin mining. Initially, the report introduces fundamental concepts related to Bitcoin mining, including Proof of Work (PoW), forks, mining pools, and block withholding attacks. It proceeds to examine the theoretical framework and empirical outcomes of selfish mining attacks, revealing the conditions under which selfish miners can reap profits exceeding their proportion of mining power. We further constructs a model for mining pool games, analyzing scenarios involving a single attacker and mutual attacks between two pools, as well as the Prisoner's Dilemma among multiple pools. The study finds that selfish mining attacks can be profitable under certain conditions but also carries risks. Moreover, the dynamics between mining pools involve complex strategic interactions that can be analyzed through Nash equilibrium to identify stable strategies.

**Keywords** **Blockchain Security** **Selfish Mining** **Withholding Attack**

## Contents

# 1 Introduction

In 2008, Satoshi Nakamoto introduced the concept of Bitcoin in his paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System."[8] A year later, the Bitcoin network, based on its open-source code, was officially launched. The Satoshi Protocol is a public, immutable distributed ledger system, more commonly known as blockchain. Within the blockchain, users can create transactions to modify their currency amounts. Each block is connected to the previous one through cryptographic hashes, forming a chain-like structure, which is the origin of the term "blockchain."

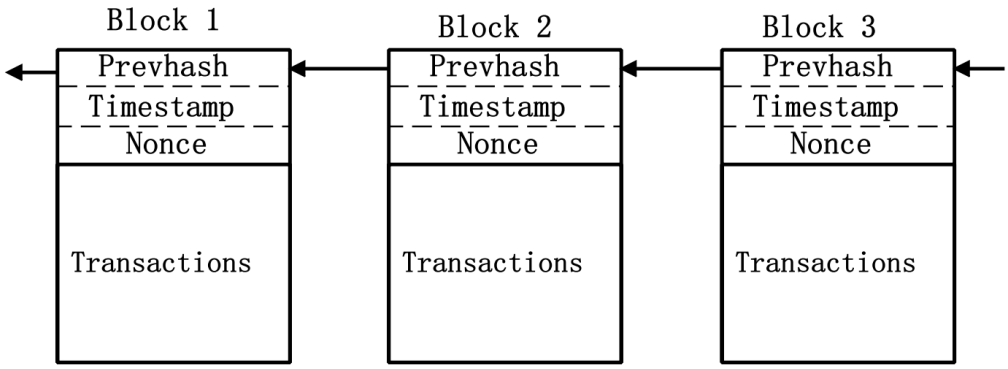


Figure 1: An illustration of the blockchain data structure.

Figure 1 is a schematic diagram of a simple blockchain structure. [11] In addition to transaction information and the cryptographic hash of the previous block, each block also contains a timestamp, a nonce, and other metadata. The nonce is obtained through extensive calculations, which is commonly referred to as Proof of Work (PoW), and this aspect will be introduced in subsequent chapters. The process of searching for the nonce value is called mining, and the participants are referred to as miners. Miners adhere to the Longest Chain Rule (LCR), meaning that miners always extend the longest chain they receive. This will be detailed in the discussion of selfish mining.

When participating in the maintenance and mining of a blockchain, miners need to cover the costs of their computational hardware (such as CPUs, GPUs, etc.), electricity, and other expenses. Therefore, the real economic costs make it unlikely for miners to voluntarily participate in the blockchain. As a result, public blockchains typically require a well-designed incentive mechanism. Miners can earn block rewards corresponding to the new blocks they publish, as well as transaction fees for all transactions included in the block[8]. This makes mining profitable.

However, this mechanism assumes that all miners are rational and honest[8, 7]. In reality, if deviating from the protocol can yield higher profits, selfish miners will inevitably emerge to attack other miners in pursuit of greater gains than those obtained through honest mining. Therefore, to ensure the rationality of honesty, the blockchain system should ensure incentive compatibility, meaning that miners will suffer economic losses if they deviate from the protocol.

Under the current mechanisms, attacks between miners and among mining pools are inevitable as each party seeks to maximize their own interests. A balanced strategy will naturally emerge, and we can use game theory to analyze selfish mining attacks, mining

pool attacks, and other scenarios to explore their Nash equilibrium strategies. By feeding these results back into the incentive mechanism, we can optimize it.

# 2 Preliminaries of Bitcoin Mining

Before formally introducing the incentive mechanisms for Bitcoin security, we first need to introduce some basic concepts related to the Bitcoin system, including Proof of Work, the Longest Chain Rule, mining pools, block withholding attacks, and other concepts.

## 2.1 Revenue for Proof of Work(PoW)

The blockchain is a distributed ledger system that records each transaction in the form of blocks. As mentioned in the introduction, a valid block will contain the cryptographic hash of the previous block, the hash of the transactions in the current block, and a Bitcoin address that receives a reward for generating this block. This reward is commonly referred to as the block reward, which is typically the cryptocurrency of the Bitcoin system—Bitcoin.

The task required of miners in mining involves repeatedly calculating a hash function, specifically the SHA-256 function located in the block header, which is shown in Figure 2. To demonstrate that they have completed this computational work, miners provide a probabilistic proof. Within the block generated by the miner, there is a random number field called "nonce" which can contain any value. The miner places different values in this field and calculates the hash for each value. If the hash result is less than the target value set by the system, then this nonce is considered a solution and the block is deemed valid.

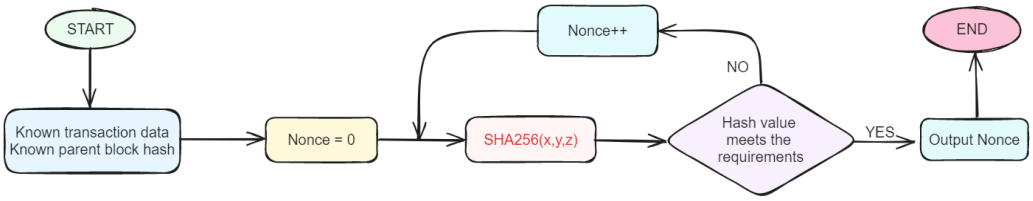


Figure 2: An illustration mining Process.

Therefore, as indicated by the above text, the number of attempts miners make to find a single hash can be considered a random variable that follows a geometric distribution, because each attempt is a Bernoulli trial with a probability determined by the target value. Given the enormous hash rate and extremely small target value set by the current system, the time taken to find a single hash can be approximated as following an exponential distribution. **Consequently, the average time for miners to find a solution is directly proportional to their hash rate or, equivalently, their mining power.**

To maintain a constant rate of Bitcoin generation and as part of the defense against denial-of-service and other attacks, the system normalizes the rate of block generation. The protocol defines the target value for each block based on the time recently required to generate blocks, ensuring that the average time to find each block is 10 minutes. Additionally, since the exponential distribution is memoryless, when all miners are mining block $n$ and one miner finds this block at time $t$, all miners will immediately switch their work to mining block $n+1$ without altering the probability distribution of their chances to find a new block after time $t$.

## 2.2 Forks

After a new block is generated, it is immediately broadcasted to the entire network. Ideally, before the next block is produced, all participants would accept this block. However, within the entire blockchain network, due to the propagation of blocks taking several seconds, two distant miners may potentially generate competing blocks that both regard the same block as their predecessor, leading to a fork [2]. To resolve such a fork, honest miners always accept the longest chain as the valid chain and continue mining after its last block.

If multiple chains have the same length, miners will choose to continue mining on the chain that they received first. Forks do not persist indefinitely because the longest branch will eventually win the competition and be accepted by all miners. The common prefix of such a longest chain is referred to as the main chain. **It is important to note that this mechanism results in the discarded blocks from the competition, which are not reattached to the longest chain afterward.** This aspect will be revisited when discussing the selfish mining attack.

## 2.3 Pools

As the value of Bitcoin rises, Bitcoin mining has become a rapidly growing industry, with mining using hardware other than the strongest mining equipment being unprofitable, as the energy costs would exceed the expected revenue. Although the expected revenue is always proportional to the mining power, individual miners using small-scale equipment may not mine a block for an extremely long time [10]. Therefore, miners tend to form a "mining pool."

The mechanism of a mining pool does not increase the expected income for each individual; rather, it is a group of miners who share the block reward when one of them successfully mines a block. For each mined block, the reward is distributed proportionally according to the mining power of each miner, so the expected income of pool members is the same as that of solo mining. However, due to the pool's powerful mining capacity, it mines blocks at a higher rate, thus enabling stable long-term income.

## 2.4 Block Withholding

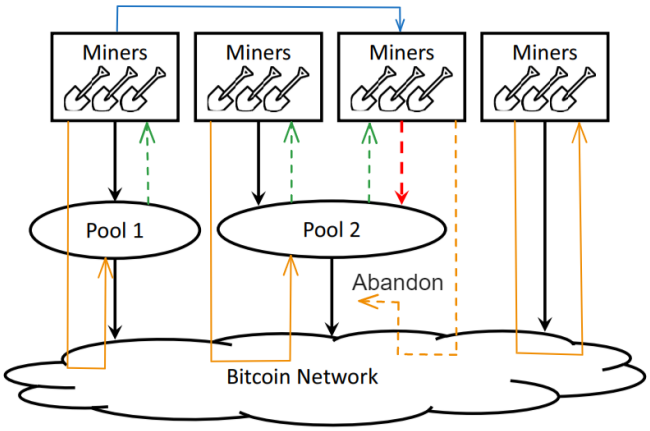


Figure 3: An illustration mining Process.

The classic block withholding attack is an attack launched by one mining pool against another [9]. The attacking miner registers with the target pool and appears to begin mining honestly, periodically sending partial proof of work to the pool. However, when the attacking miner finds a nonce that constitutes a complete proof of work, it discards the nonce, aiming to reduce the total revenue of the attacked pool. This type of attack is depicted in Figure 3.

From Figure 3, we can observe that the attacking pool diverts a portion of its miners to infiltrate another pool. These infiltrating miners do not directly alter the mining capability of the target pool, but due to the revenue-sharing agreement, they siphon off a portion of the total earnings from the attacked pool.

# 3 Selfish Mining Attack

The idea of selfish mining was initially proposed on the Bitcoin forums, and later Eyal and Sirer formally described and analyzed the selfish mining attack[4]. Selfish mining allows a sufficiently large mining pool to obtain revenue that exceeds its proportion of mining power. For simplicity, and without loss of generality, we assume that miners are divided into two groups: one is a colluding minority pool that follows the selfish mining strategy, and the other is a majority that follows the honest mining strategy (others). It does not matter whether the honest miners operate as a single group, a collection of groups, or individually.

## 3.1 Selfish Mining in Bitcoin

The key to the selfish mining strategy is to force honest miners to waste their computational power on stale public branches, meaning that selfish mining forces honest miners to spend their computational efforts on blocks that are destined not to be part of the blockchain. Selfish miners achieve this by selectively revealing the blocks they mine to invalidate the blocks of honest miners.

During the attack, selfish miners first privatize their newly discovered blocks, creating a hidden branch. When some honest miners mine a new block, the selfish miners immediately publish some of their private blocks, making their branch the longest chain. Due to the longest chain rule, the blocks of honest miners are abandoned, and the selfish miners can obtain more block rewards.

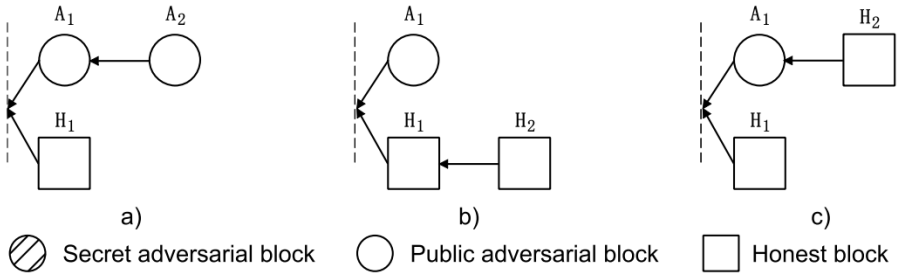


Figure 4: Example of three cases of the selfish mining attack in Bitcoin.

Figure 4 illustrates three scenarios of the selfish mining attack. In the figure, $A$ represents the blocks mined by the selfish miners, and $H$ represents the blocks mined by the honest

miners. If the selfish miners and the honest miners form two branches of equal length, miners cannot determine which is the longest and can only continue mining until a longer branch emerges as the winner. Since selfish miners aim to maximize their own interests, they will mine on the $A$ chain. Due to the propagation time and the longest chain principle, other miners decide which chain to continue mining on based on the order in which they receive the chains. Therefore, the next block can be further divided into three scenarios:

1. The selfish miners generate and publish the next block $A_2$ after their own branch. Following the Longest Chain Rule (LCR), all blocks of the selfish miners are accepted by the honest miners, and the honest block $H_1$ is discarded.

2. Honest miners who have chosen the honest block as their prefix generate and publish the next block $H_2$. The selfish miners accept the honest blocks $H_1$ and $H_2$ as their prefix, and their selfish block $A_1$ is discarded.

3. Honest miners who have chosen the selfish block as their prefix generate the next block $H_2$ after the $A_1$ block. The block $H_1$ is discarded due to the LCR.

### 3.2 Theoretical Analysis and Result

From above analysis, selfish mining attacks are not always profitable, and there is still a risk of loss for selfish mining. The success of the attack depends on two factors: the computational power owned by the selfish miners (represented by $\alpha$) and the computational power of honest miners who choose to work on the selfish blockchain (represented by $\gamma$).

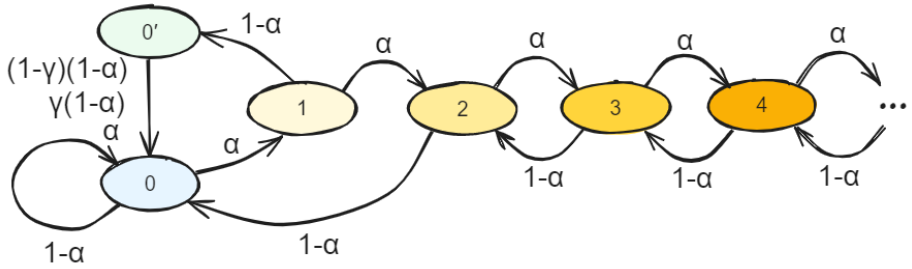


Figure 5: State machine with transition frequencies.

Figure 5 illustrates the process of state transitions in the form of a state machine [5]. The system's state represents the lead of the selfish mining pool, which is the difference between the number of unpublished blocks in the private branch and the length of the public branch. Zero lead is divided into state 0 and state $0'$, with the former being the state without branches and the latter being the state with two public branches of length 1. The transitions in the figure correspond to mining events, which occur at average frequencies of $\alpha$ and $(1-\alpha)$ respectively on exponential intervals.

For state $s = 0, 1, 2, \ldots$, the selfish miners mine a block with probability $\alpha$, increasing the lead to $s+1$; for state $s = 3, 4, \ldots$, the honest miners mine a block with probability $(1-\alpha)$, decreasing the lead to $s-1$. If the other miners mine a block when the lead is 2, the selfish miners publish their private branch, and the system drops to a lead of 0. If the honest miners mine a block when the lead is 1, the system reaches state $0'$. From state $0'$, there are three possible transitions, all leading back to state 0 with a total probability of 1:

1. The selfish miners mine a block on their previously private branch (frequency $\alpha$)
2. The honest miners mine a block on the previously private branch (frequency $\gamma(1-\alpha)$)
3. The honest miners mine a block on the public branch (frequency $(1-\gamma)(1-\alpha)$

We analyze the state machine and calculate the probabilities of the states as $p_{0'}, p_0, p_1, \ldots$, and so on. By calculating the probability distribution, we determine the final revenue obtained by the selfish miners and the honest miners as follows:

$$\begin{aligned} r_{\text{honest}} &= p_{0'} \cdot \gamma\,(1-\alpha) \cdot 1 + p_{0'} \cdot (1-\gamma)\,(1-\alpha) \cdot 2 + p_0 \cdot (1-\alpha) \cdot 1 \\ r_{\text{selfish}} &= p_{0'} \cdot \alpha \cdot 2 + p_{0'} \cdot \gamma\,(1-\alpha) \cdot 1 + p_2 \cdot (1-\alpha) \cdot 2 + p\,[i>2]\,(1-\alpha) \cdot 1 \end{aligned} \tag{1}$$

Selfish mining results in some blocks being discarded, causing a decrease in the total block generation rate, which is $r_{\text{selfish}} + r_{\text{honest}} < 1$. Therefore, the actual revenue rate should be the ratio of their respective revenue rates, representing the proportion of blocks in the main chain. Since the Bitcoin community stipulates that no mining pool's computing power should exceed 50% of the system's total computing power [5], we use Equation 2 to represent the revenue of the selfish mining pool when $0 < \alpha < \frac{1}{2}$.

$$R_{\text{selfish}} = \frac{r_{\text{selfish}}}{r_{\text{selfish}} + r_{\text{honest}}} = \frac{\alpha\,(1-\alpha)^2\,(4\alpha + \gamma\,(1-2\alpha)) - \alpha^3}{1 - \alpha\,(1 + (2-\alpha)\,\alpha)} \tag{2}$$

When the relative revenue of the selfish miners is greater than $\alpha$, which is the proportion of the total system computing power they possess, miners will choose to initiate an attack. By substituting this condition into Equation 2, we can obtain that for a given $\gamma$, selfish miners with $\alpha$ fraction of computing power can achieve greater revenue than honest mining within the following range:

$$\frac{1-\gamma}{3-2\gamma} < \alpha < \frac{1}{2} \tag{3}$$

# 4 Pool Game

## 4.1 Model Construction of Pool Game

To analyze the attack game between mining pools, we first assume that there are enough miners such that the mining power can be arbitrarily divided without constraint. Let $p$ denote the number of mining pools, $m$ represent the total mining power of the system, and $m_i$ denote the number of miners participating in pool $i$. We adopt a quasi-static analysis, where the participation of miners in mining pools does not change over time.

**Pool** A mining pool acts as a coordinator node, where multiple miners can register to work. At each step, the mining pool generates a task for each registered miner. Each miner receives the task and works on it within the time frame of that step. At the end of the step, miners send back the complete and partial proofs of work they have found to the mining pool. The mining pool collects all proofs of work, records the partial proofs, and publishes the complete proofs. The overall revenue obtained is distributed to each miner according to the proportion of partial proofs of work they have sent, which is equivalent to distributing it according to their mining power.

**Withholding Miner** Miners registered to a mining pool can execute a classic block withholding attack, **where the attacking miners appear to work for the pool but only send partial proofs of work at the end of each step, discarding any complete proofs they find**. As a result, the attacking miners do not contribute to the overall mining power of their pool, yet they still share the pool's revenue proportionally based on the partial proofs they send.

To evaluate the efficiency of a mining pool, we define a new criterion—income density. The income density of a mining pool is the ratio of the average income of its members to the average income of miners operating independently. The income density of independent miners, as well as miners cooperating with an unattacked mining pool, is 1. If a mining pool is subjected to a block withholding attack, its income density will decrease.

## 4.2 General Analysis

Suppose that mining pool $i$ uses a fraction of its mining power to infiltrate mining pool $j$ and execute a block withholding attack, with $x_{i,j}(t)$ representing the amount of this infiltrating mining power at step $t$. Miners working for mining pool $i$, whether they are mining honestly or infiltrating mining pool $j$, are loyal to mining pool $i$. At the end of each round, mining pool $i$ aggregates the mining revenue for the current round and the infiltration revenue from the previous round, and distributes it proportionally to all loyal miners based on their partial proofs of work.

### 4.2.1 Revenue Convergence

If the infiltration rate of the mining pool is constant, then the income of the mining pool will converge. We use $r_i\,(t)$ to denote the income density of mining pool $i$ at the end of step $t$, and define the income density vector:

$$\mathbf{r(t)} \triangleq (r_1(t), \ldots, r_p(t))^T \tag{4}$$

In each round, mining pool $i$ uses its mining power $m_1 - \sum_j^p x_{1,j}$ for direct mining and distributes it among its $m_1 + \sum_j^p x_{j,1}$ members, which includes malicious infiltrators. We represent the direct mining income density of each mining pool with a vector:

$$\mathbf{m} \triangleq \left( \frac{m_1 - \sum_j x_{1,j}}{m_1 + \sum_j x_{j,1}}, \ldots, \frac{m_p - x_{p,j}}{m_p + \sum_j x_{j,p}} \right)^T \tag{5}$$

The income that mining pool $i$ obtains through infiltration from the income of mining pool $j$ at step $t-1$ is $x_{i,j} r_j\,(t-1)$. Mining pool $i$ distributes its income among its $m_i + \sum_k x_{k,i}$ members—loyalists and infiltrators. We define the $p \times p$ infiltration matrix through its $i$, $j$ elements:

$$\mathbf{G} \triangleq \left[ \frac{x_{i,j}}{m_i + \sum_k x_{k,i}} \right]_{ij} \tag{6}$$

Therefore, the income vector at step $t$ is:

$$\mathbf{r}(t) = \mathbf{m} + \mathbf{G}\mathbf{r}(t-1) \tag{7}$$

Since the sum of each row in the infiltration matrix is less than 1, according to the Perron-Frobenius theorem[5], its largest eigenvalue is less than 1. Therefore, the income of all mining pools converges as follows:

$$\mathbf{r}(t) = \left(\sum_{t'=0}^{t-1} G^{t'}\right)\mathbf{m} + G^t\mathbf{r}(0) \xrightarrow{t\to\infty} (1-\mathbf{G})^{-1}\mathbf{m} \tag{8}$$

#### 4.2.2 Revenue Density

**In the mining pool game, the mining pools attempt to optimize their penetration rate into other pools in order to maximize their revenue.** [3] We assume that the total number of miners and the number of miners loyal to each pool remain constant throughout the game. The mining rate of pool $i$ is the number of its loyal miners minus the number of miners used for penetration. This effective mining rate, divided by the total mining rate of those in the system who do not participate in penetration and focus solely on mining, yields the direct mining rate of pool $i$ at step $t$:

$$R_i \triangleq \frac{m_i - \sum_{j=1}^{p} x_{i,j}}{m - \sum_{j=1}^{p}\sum_{k=1}^{p} x_{j,k}} \tag{9}$$

The revenue density of mining pool $i$ at the end of step $t$ is the sum of its direct mining revenue plus the revenue obtained from the penetrated pools, divided by the sum of the number of miners loyal to pool mi and the number of miners that have penetrated into pool $m_i$.

$$r_i(t) = \frac{R_i(t) + \sum_{j=1}^{p} x_{i,j}(t) r_j(t)}{m_i + \sum_{j=1}^{p} x_{j,i}(t)} \tag{10}$$

Subsequently, we turn to static analysis and omit t in the expression.

### 4.3 One Attacker

We begin with a simple game between two mining pools, assuming that pool 1 conducts a penetration attack on pool 2, while pool 2 does not penetrate pool 1, as shown in Figure 6. The dashed red arrow represents that pool 1 has sent $x_{1,2}$ miners to attack pool 2. Additionally, there are $m-m_1-m_2$ miners who mine independently. According to Equation 9, the direct revenue for the two pools is as follows:

$$\begin{aligned} R_1 &= \frac{m_1 - x_{1,2}}{m - x_{1,2}} \\ R_2 &= \frac{m_2}{m - x_{1,2}} \end{aligned} \tag{11}$$

Pool 2 distributes its revenue to its loyal miners and the miners that have penetrated into pool 2, so its revenue density is:

$$r_2 = \frac{R_2}{m_2 + x_{1,2}} \tag{12}$$

Pool 1 distributes its revenue to its registered miners, which includes the direct mining revenue and the revenue obtained by the penetrators from other pools, which is $r_2 \cdot x_{1,2}$. Therefore, the revenue density for each miner loyal to pool 1 is:

$$r_1 = \frac{R_1 + x_{1,2} \cdot r_2}{m_1} \tag{13}$$

We substitute $r_2$ from Equation 12 and $R_1$, $R_2$ from Equation 11 into the equation, resulting in the expression for $r_1$ in Equation 13.

$$r_1 = \frac{m_1 (m_2 + x_{1,2}) - x_{1,2}^2}{m_1 (m - x_{1,2}) (m_2 + x_{1,2})} \tag{14}$$

$r_1$ has a maximum value point within the feasible range $0 \leq x_{1,2} \leq m_1$. Since pool 2 cannot respond to the attack from pool 1, this point represents a stable state of the system. We denote the value of $x_{1,2}$ at this point as $\bar{x}_{1,2} \triangleq \arg\max_{x_{1,2}} r_1$. To simplify the expression, we standardize by setting $m = 1$.

$$\bar{x}_{1,2} = \frac{m_2 - m_1 m_2 - \sqrt{-m_2^2 (-1 + m_1 + m_1 m_2)}}{-1 + m_1 + m_2} \tag{15}$$

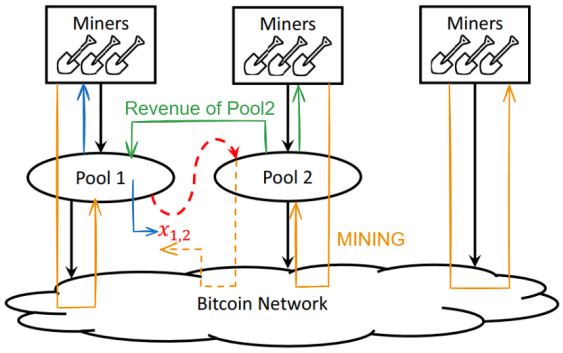


Figure 6: The one-attacker scenario. Pool 1 infiltrates pool 2.

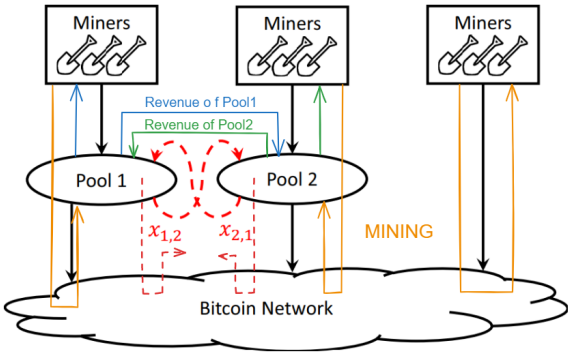


Figure 7: Two pools infiltrating each other.

From the above analysis, we can also conclude that in a system with p mining pools, the point $\forall j, k \ : x_j^k = 0$ is not an equilibrium point. If we assume that no mining pool attacking is an equilibrium point, and it is known that mining pool 1 can increase its revenue by attacking mining pool 2, let $\bar{x}_{1,2}$ represent the penetration rate of mining pool 1. When this is incorporated back into the system with p mining pools:

$$x_{1,2} = \tilde{x}_{1,2} \forall (j,k) \neq (1,2) \ : x_{1,2} \ = \ 0 \tag{16}$$

At this time, mining pool 1 has better revenue, so mining pool 1 can increase its revenue density by attacking other mining pools. A situation where no one is attacking is not an equilibrium point.

### 4.4 Two Pools Attack Each Other

We continue to analyze the scenario where two mining pools attack each other, as shown in Figure 7. Suppose there are mining pools 1 and 2 with sizes $m_1$ and $m_2$, respectively. The penetration rate of mining pool 1 into mining pool 2 is $x_{1,2}$, and the penetration rate of mining pool 2 into mining pool 1 is $x_{2,1}$. The direct mining rates $R_1$ and $R_2$ for the mining pools are given by the following formulas:

$$\begin{aligned} R_1 &= \frac{m_1 - x_{1,2}}{m - x_{1,2} - x_{2,1}} \\ R_2 &= \frac{m_2 - x_{2,1}}{m - x_{1,2} - x_{2,1}} \end{aligned} \tag{17}$$

The total revenue of each mining pool consists of its direct mining revenue and the revenue from penetration in the previous round. The total revenue is distributed among its loyal miners and the miners that have penetrated it, proportionally to their share of the proof of work. In a stable state, the revenue density of the two mining pools is as follows:

$$\begin{aligned} r_1 &= \frac{R_1 + x_{1,2} r_2}{m_1 + x_{2,1}} \\ r_2 &= \frac{R_2 + x_{2,1} r_1}{m_1 + x_{1,2}} \end{aligned} \tag{18}$$

To solve for r1 and r2, we obtain a closed-form expression for the revenue of each mining pool. We express the revenue as a function of $x_{1,2}$ and $x_{2,1}$.

$$\begin{aligned} r_1\left(x_{1,2}, x_{2,1}\right) &= \frac{m_2 R_1 + x_{1,2}\left(R_1 + R_2\right)}{m_1 m_2 + m_1 x_{1,2} + m_2 x_{2,1}} \\ r_2\left(x_{2,1}, x_{1,2}\right) &= \frac{m_1 R_2 + x_{2,1}\left(R_1 + R_2\right)}{m_1 m_2 + m_1 x_{1,2} + m_2 x_{2,1}} \end{aligned} \tag{19}$$

There exists an equilibrium point where neither mining pool 1 nor mining pool 2 can increase their revenue density by altering their penetration rates. In other words, any pair of values $x_1', x_2'$ satisfies the following conditions:

$$\begin{aligned} &\begin{cases} \arg\max_{x_{1,2}} r_1\left(x_{1,2}, x_{2,1}^{'}\right) = x_{1,2}^{'} \\ \arg\max_{x_{2,1}} r_2\left(x_{1,2}^{'}, x_{2,1}\right) = x_{2,1}^{'} \end{cases} \\ s.t. \quad & 0 < x_1^{'} < m_1 \qquad m_1 > 0 \\ & 0 < x_2^{'} < m_2 \qquad m_2 > 0 \\ & m_1 + m_2 \leq m \end{aligned} \tag{20}$$

For all feasible variable values, the revenue function $r_i$ is concave with respect to $x_i$ $\left(\partial^2 r_i / \partial x_i^2 < 0\right)$. Therefore, the solution to the equation is unique and either lies on the boundary of the feasible region or satisfies $\partial r_i / \partial x_{i,j} = 0$. From the previous section, we know that no attack is not an equilibrium point because each pool can increase its revenue

by choosing strictly positive penetration rates, i.e., $x_{1,2} = x_{2,1} = 0$ is not a solution to Equations 20. Therefore, a Nash equilibrium exists, where the values of $x_{1,2}, x_{2,1}$ satisfy:

$$\begin{cases} \frac{\partial r_1(x_{1,2},x_{2,1})}{\partial x_{1,2}} = 0 \\ \frac{\partial r_2(x_{2,1},x_{1,2})}{\partial x_{2,1}} = 0 \end{cases} \tag{16}$$

### 4.5 Prisoner's Dilemma and Multiple Pools Game

In a healthy Bitcoin environment, where no mining pool controls more than half of the system's mining power, the revenue of two mining pools in an equilibrium state will be lower than the revenue when neither pool attacks. Suppose there are mining pools 1 and 2. If pool 2 does not attack, pool 1 can increase its revenue above 1 by attacking. If pool 2 attacks but pool 1 does not, we use $\tilde{r}_1$ to represent the revenue of pool 1. The exact value of $\tilde{r}_1$ depends on the values of $m_1$ and $m_2$, but it is always less than 1. As mentioned above, if pool 1 does choose to attack, its revenue will increase, but it will not exceed 1. The results of this game are summarized in Table 1.

| Pool 2 \ Pool 1 | no attack | attack |
|---|---|---|
| no attack | $(r_1 = 1, r_2 = 1)$ | $(r_1 > 1, r_2 = \tilde{r}_2 < 1)$ |
| attack | $(r_1 = \tilde{r}_1 < 1, r_2 > 1)$ | $(r_1 < \tilde{r}_1 < 1, r_2 < \tilde{r}_2 < 1)$ |

Table 1: Prisoner's Dilemma for two pools.

When played only once, this is the classic Prisoner's Dilemma, where attacking is the dominant strategy, as the revenue density for pool 1 is always greater when attacking, regardless of whether pool 2 chooses to attack or not. The same applies to pool 2. Therefore, mutual attacking is a Nash equilibrium state.

However, mining is a long-term process, and mining pools can change strategies and even communicate information. A pool can detect whether it is being attacked and infer that the other pool is violating the agreement. **In such a supergame, although the single Nash equilibrium in each round is to attack, the cooperative state where neither pool attacks is a possible stable state**[6, 1].

$$\begin{aligned} R_1 &= \frac{m_i - (q-1)x_{1,\neg 1}}{m - (q-1)(q-1)x_{\neg 1,*} - (q-1)x_{1,\neg 1}} \\ R_{\neg 1} &= \frac{m_i - (q-1)x_{\neg 1,*}}{m - (q-1)(q-1)x_{\neg 1,*} - (q-1)x_{1,\neg 1}} \\ r_1 &= \frac{R_1 + (q-1)x_{1,\neg 1}r_{\neg 1}}{m_i + (q-1)x_{\neg 1,1}} \\ r_{\neg 1} &= \frac{R_{\neg 1} + (q-2)x_{\neg 1,*}r_{\neg 1} + x_{\neg 1,*}r_1}{m_i + (q-2)x_{\neg 1,*} + x_{1,\neg 1}} \end{aligned} \tag{21}$$

If we consider a scenario with $q$ mining pools of equal size attacking each other, there exists a symmetric equilibrium in this case. Due to the symmetry, the attack rates among the mining pools are the same. Let $x_{1,\neg 1}$ represent the attack rate of mining pool 1 on any other mining pool, and let $x_{\neg 1,*}$ represent the attack rate of any non-1 mining pool on any

other mining pool (including mining pool 1). Let $R_1$ and $R_{\neg 1}$ denote the direct revenue of mining pool 1 and other mining pools, respectively. Let $r_1$ and $r_{\neg 1}$ represent the revenue density of mining pool 1 and other mining pools, respectively. We can obtain Equation 21.

In the symmetric case, we have $r_1 = r_{\neg 1}$. Given any values of $q$ and $m_i$ (where $qm_i < 1$), the allowed range of penetration rates is $0 \leq x_{i,j} \leq m_i/q$. Within this range, $r_i$ is continuous, differentiable, and concave with respect to $x_{1,\neg 1}$. Therefore, the optimal point for mining pool 1 is where $\partial r_1/\partial x_{1,\neg 1} = 0$. To find the symmetric equilibrium, we set $x_{1,\neg 1} = x_{\neg 1,1} = x_{\neg 1,*}$ and obtain a single feasible solution. The equilibrium penetration rate and corresponding revenue are shown in equation 22. We can see that, as in the previous two cases, the revenue at the symmetric equilibrium is lower than the non-equilibrium strategy of no one attacking.

$$\begin{aligned} \bar{x}_{1,\neg 1} = \bar{x}_{\neg 1,1} = \bar{x}_{\neg 1,*} &= \frac{q - m_i - \sqrt{(m_i - q)^2 - 4\,(m_j)^2\,(q-1)^2\,q}}{2\,(q-1)^2\,q} \\ \bar{r}_1 = \bar{r}_{\neg 1} &= \frac{2q}{q - m_i + 2m_i q + \sqrt{(m_i - q)^2 - 4\,(m_i)^2\,(q-1)^2\,q}} \end{aligned} \tag{22}$$

# 5 Conclusion

Through the lens of game theory, we have provided an in-depth analysis of strategic behaviors in Bitcoin mining, specifically focusing on selfish mining attacks and the game dynamics between mining pools. The following conclusions are drawn:

1. Feasibility of Selfish Mining: Selfish mining can yield excess profits for selfish miners under specific conditions, but it is not always advantageous as it relies on the computational power controlled by the miners and market conditions.

2. Complexity of Pool Games: The strategic interactions between mining pools are intricate, involving both attacks and cooperation. While attacks may enhance short-term gains, cooperation may represent a superior stable strategy in the long run.

3. Importance of Incentive Mechanisms: To ensure the long-term security and stability of blockchain, it is essential to design incentive mechanisms that reward miners for adhering to the protocol and economically penalize deviations.

4. Policy Recommendations: Based on the analysis, this paper suggests that the Bitcoin community and developers consider introducing additional incentives to encourage miners to follow the protocol and prevent potential attacks.

5. Future Research Directions: This study lays the groundwork for understanding strategic behaviors in Bitcoin mining. Future research can further explore the economic impacts of different mining strategies and how technological innovation can enhance the blockchain's resilience to attacks.

# A Answers to the Last Homework

1. **Please state the Hierarchy of Equilibrium Concepts.**
   In game theory, the hierarchy of equilibrium concepts evolves from the basic Nash equilibrium to more refined notions like Subgame Perfect Nash Equilibrium and Perfect Bayesian Nash Equilibrium, progressively eliminating strategies based on irrational beliefs to ensure optimality at every decision point.

2. **What is the Tragedy of the commons? What is the Price of Stability, and the difference to the PoA?**
   The Tragedy of the Commons refers to the depletion of a shared resource due to individual self-interest. PoS measures the ratio of the optimal social cost in a stable state to the actual social cost, ensuring no player can improve their outcome by deviating. PoA, on the other hand, compares the social cost in a Nash equilibrium to the optimal social cost, reflecting the inefficiency caused by selfish behavior without considering stability.

3. **Please briefly state the history of Online Ad Auction. What is the GSP, and its equilibrium results?**
   Online ad auctions began with simple bidding based on ad performance evaluation and later evolved into strategic bidding based on metrics like click-through rates and conversion rates. With the innovation of auction mechanisms, the introduction of the GSP auction transformed it into a dynamic game where advertisers continuously adjust their strategies and engage in signaling in an environment of information asymmetry.
   GSP is an online advertising auction mechanism where the highest bidder wins the ad spot but pays the price of the second-highest bid. The equilibrium outcome of GSP is a Nash equilibrium, meaning that given the bids of other bidders remain unchanged, no bidder can improve their outcome by altering their own bid, resulting in a stable sequence of bids.

4. **What is the fictitious play dynamic?**
   Fictitious Play Dynamics is a learning process where participants update their strategies based on their opponents' past behaviors, assuming that their opponents will repeat their previous strategies. It simulates a fictitious game in which each player tries to predict and adapt to the strategic patterns of their opponents.

5. **What is the cooperative game, Shapely value and core?**
   In cooperative game theory, players can form coalitions to maximize collective payoff. The Shapley value allocates payoffs fairly among players based on their marginal contributions to all possible coalitions, while the core represents a set of stable payoff allocations where no coalition has an incentive to deviate.

## References


[1] Robert J Aumann and Lloyd S Shapley. Long-term competition—a game-theoretic analysis. In *Essays in Game Theory: In Honor of Michael Maschler*, pages 1–15. Springer, 1994.

[2] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1–10. IEEE, 2013.

[3] Ittay Eyal. The miner's dilemma. In *2015 IEEE symposium on security and privacy*, pages 89–103. IEEE, 2015.

[4] Ittay Eyal and Emin Gün Sirer. Majority is not enough: bitcoin mining is vulnerable. *Commun. ACM*, 61(7):95–102, jun 2018.

[5] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7):95–102, 2018.

[6] James W Friedman. A non-cooperative equilibrium for supergames. *The Review of Economic Studies*, 38(1):1–12, 1971.

[7] Arthur Gervais, Ghassan O Karame, Vedran Capkun, and Srdjan Capkun. Is bitcoin a decentralized currency? *IEEE security & privacy*, 12(3):54–60, 2014.

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

[9] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980*, 2011.

[10] Eric Swanson. Bitcoin mining calculator, 2013.

[11] Duc A Tran, My T Thai, and Bhaskar Krishnamachari. *Handbook on Blockchain*, volume 194. Springer Nature, 2022.