A Game Theoretic Analysis on Block Withholding Attacks Using the Zero-Determinant Strategy

Qin Hu
Beijing Normal University
Beijing, China
Indiana University-Purdue University Indianapolis
Indianapolis, Indiana, USA
qinhu@iupui.edu

Shengling Wang*
College of Information Science and Technology
Beijing Normal University
Beijing, China
wangshengling@bnu.edu.cn

Xiuzhen Cheng
Department of Computer Science
The George Washington University
Washington DC, USA
cheng@gwu.edu

ABSTRACT

In Bitcoin's incentive system that supports open mining pools, block withholding attacks incur huge security threats. In this paper, we investigate the mutual attacks among pools as this determines the macroscopic utility of the whole distributed system. Existing studies on pools' interactive attacks usually employ the conventional game theory, where the strategies of the players are considered pure and equal, neglecting the existence of powerful strategies and the corresponding favorable game results. In this study, we take advantage of the Zero-Determinant (ZD) strategy to analyze the block withholding attack between any two pools, where the ZD adopter has the unilateral control on the expected payoffs of its opponent and itself. In this case, we are faced with the following questions: *who can adopt the ZD strategy? individually or simultaneously? what can the ZD player achieve?* In order to answer these questions, we derive the conditions under which two pools can individually or simultaneously employ the ZD strategy and demonstrate the effectiveness. To the best of our knowledge, we are the first to use the ZD strategy to analyze the block withholding attack among pools.

CCS CONCEPTS

• **Security and privacy** → *Network security*; • **Networks** → *Network reliability*.

KEYWORDS

Bitcoin, blockchain, block withholding attack, game theory

ACM Reference Format:
Qin Hu, Shengling Wang, and Xiuzhen Cheng. 2019. A Game Theoretic Analysis on Block Withholding Attacks Using the Zero-Determinant Strategy. In *IEEE/ACM International Symposium on Quality of Service (IWQoS '19), June 24–25, 2019, Phoenix, AZ, USA.* 10 pages. https://doi.org/10.1145/3326285.3329076

*Corresponding author.

IWQoS '19, June 24–25, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6778-3/19/06…$15.00
https://doi.org/10.1145/3326285.3329076

1 INTRODUCTION

As the most prevailing and typical cryptocurrency, Bitcoin [1] owns a market capitalization of 186 billion US dollars in current time, occupying an increasingly high market share of 56% in the whole cryptocurrency market[1]. The success of Bitcoin is based on its robust incentive mechanism where the miners are rewarded with valuable bitcoins for their submitted *proofs of work* (PoWs) through solving cryptographic puzzles. As the global ledger of a distributed system, *Blockchain* records all the historical transactions in a serialization form, with each block being generated by a full PoW costing a larger amount of computational power of the miners, so as to guarantee the inalterability and integrity of the transactions, and further maintain the stability and scalability of the whole system.

Being aware of the difficulty of mining blocks individually to acquire revenues, miners are inclined to form mining pools so as to obtain stable incomes, where all participants in a pool mine blocks cooperatively and share the benefit once a valid block is generated. In order to conduct fair distribution of the reward among all participating miners in a pool, the pool manager evaluates their efforts according to the submitted partial PoWs from the miners. In this case, a malicious miner can launch a *block withholding attack* to an open pool by submitting only partial PoWs to unfairly share the achievements of other honest miners, which yields huge security threats to the distributed system as only a full PoW can produce a valid block representing an effective contribution. This attack was carried out in practice at 2014, resulting in a loss of about 300 BTC at the victim pool[2].

On account of the tremendous threats of block withholding attacks, numerous studies have been conducted, which can be classified into individual attacker based [2–5] and competitive pool based [6, 7]. Analysis from the perspective of an individual attacker is committed to working out an optimal attacking strategy for the rogue miner to acquire filthy lucre as much as possible, such as how to wisely split the computational power for attacking and honestly mining; while research on block withholding attack among pools concentrates on the impact of attacking each other on the short-term utilities of the pools and long-term status of the system. In this paper, we are also engaged in the pools' mutual attack as it depicts the attacking behavior on a macroscopic level, boosting our further understanding about the dynamics of the block withholding attack and its impact on the whole system.

[1] https://www.newsbtc.com/2018/09/11/cryptocurrency-market-update-bitcoin-dominance-reaches-new-2018-high/
[2] https://bitcointalk.org/?topic=441465.msg7282674

Existing studies on block withholding attacks among pools always make use of game theory to model the mutual attacks as it can well describe the competitive relationship among pools and capture their antagonistic interactions of the attacking behaviors [6, 7]. However, the pools' strategies of attacking or not are considered pure and equally important in the above conventional game theory based studies, which prevents the discovery of interesting and advantageous results.

In this paper, we conduct an analysis on the block withholding attack among two pools from a new game theoretic perspective where the strategy of one player can unilaterally and significantly affect the game result. This unique property stems from the power of the Zero-Determinant (ZD) strategy [8], which thoroughly refreshes our understanding of the two-player game. Particularly, by utilizing ZD, one game player can unilaterally control the weighted sum of the two players' expected payoffs, as well as the specific payoff of the opponent or itself, regardless of the strategy of the opponent. With such a powerful strategy, we are faced with the following questions in our pool game: *who can adopt the ZD strategy? individually or simultaneously? what can the ZD player achieve?*

In order to answer the above questions, we model the block withholding attack among any two pools in the Bitcoin system[3] as a two-player simultaneous game, based on which we conduct a ZD strategy oriented analysis. More specifically, we first investigate the condition under which any pool can employ the ZD strategy individually; besides, we examine the possibility of both pools employing ZD concurrently and demonstrate the effectiveness of the ZD strategy as well. To the best of our knowledge, we are the first to utilize ZD for analyzing the block withholding attack among mining pools. Our conclusions can be summarized as follows:

- Each pool can individually utilize ZD to set the expected payoff of itself or that of the opponent.
- Both pools can simultaneously employ ZD to set the expected payoffs of each other under certain condition, but they are not eligible to set their own expected payoffs concurrently.
- When two pools can simultaneously use ZD, the game becomes a Prisoner's Dilemma where the Nash equilibrium leads to the lowest social welfare. In this case, even though ZD is not constantly dominant to the classical strategies, it provides a relatively good result. More importantly, the ZD strategy enables the ZD player to solely enforce a fixed social welfare which is larger than that in the equilibrium state, no matter what strategy the opponent employs.
- In the case where only one pool is capable of using ZD, when the ZD player sets the highest expected payoff for itself, the non-ZD player suffers a lower payoff, and its best response action is to not attack.

The rest of the paper is organized as follows. We investigate the most related work in Section 2 and formulate the problem in Section 3. In Section 4, we analyze the pool strategies and study the case of using the ZD strategy individually. Then we explore the possibility of simultaneously employing ZD for two pools in Section 5, which is further evaluated in Section 6. We conclude the paper in Section 7.

[3]Our results are applicable to all similar cryptocurrency systems.

2 RELATED WORK

Typically, there exist two types of threats in Bitcoin by means of withholding blocks, i.e., *block withholding delay* and *block withholding attack*, which differ in that whether the withheld blocks are finally published or not.

Block withholding delay refers to the temporary and intentional delay of publishing a mined valid block, so as to increase the revenue of the attackers. In this case, two specific attacks can be achieved, i.e., *selfish mining* and *double spending*. A selfish-mining attacker utilizes the withheld block to prevent the successful block publication of the other pools, which can help increase its own relative revenue in the whole mining system [9–11]. While double spending can be realized through withholding one block containing a specific transaction but publishing another conflicting transaction, and then posting the former one to invalidate the published one that has been admitted by the recipient, so as to improve the absolute revenue of the attacker [12].

In the block withholding attack, the withheld blocks will never be published; instead, they are discarded at once, degrading the mining utility of the victim pool, as well as undermining the overall performance of the whole Bitcoin network. According to [2], Rosenfeld was the first to put forward this attack. After that, Courtois *et al.* [3] refined the concrete concept of block withholding attack and presented the detailed implementation steps for rogue miners to get maximized benefit in the long term. While in [4], Bag *et al.* proposed a variant block withholding attack where the attacker was sponsored by one pool to attack another pool and derived the optimal strategy for the attacker to split its mining power wisely so as to obtain the highest revenue. To suppress such a malicious attack, countermeasures [13, 14] were designed to eliminate the behavior of withholding valid blocks by providing extra incentives of submitting blocks to attackers.

Beyond these conventional studies on block withholding attacks, many researchers turn to employ game models to investigate the problem as game theory can capture complicated interactions among competitive parties. In [5], a computational power splitting game was formulated to analyze the block withholding attack in Bitcoin, which demonstrates the long-run incentive of the attackers. Note that the mutual attack of the competitive pools was not considered in [5]. Eyal [6] modeled the block withholding attack among identical mining pools as a pool game, drawing the conclusion that no attack was not the Nash equilibrium and a dilemma was thus formed. Kwon *et al.* [7] proposed an upgraded version of the block withholding attack named *Fork After Withholding Attack*, which could benefit the attacker 56% more than the traditional one, and conducted a game theoretic analysis on two pools' mutual attack. The study in [7] also reported that under certain specific conditions, the miner's dilemma mentioned above disappears. Note that the pool strategies in the above two studies were considered pure and identical, missing other promising and favorable game results.

It is obvious that most existing game theory based studies on block withholding attacks either involve no interactive behaviors among pools or analyze the strategies of pools in an equivalent manner. In contrast, our work consider the mutual attack of pools with mixed game strategies to explore the unexpected existing of unilateral control in the game.

3 PROBLEM FORMULATION

In most of the contemporary digital currency systems, e.g., Bitcoin, miners often form mining pools to get stable incomes. As mentioned in [6], a pool may use a certain amount of infiltrating mining power to perform block withholding attacks to sabotage another pool for more interest. In this paper, we consider a scenario with two mining pools in a distributed system, where both pools can perform block withholding attacks to against each other during the mining process of an effective block. Once a legitimate block is generated by a miner in a pool, the pool manager would publish it and get the corresponding revenue. In order to fairly distribute the revenue to all participating miners, the pool manager collects partial PoWs to evaluate the miners' respective efforts in mining the block. As the infiltrating mining power also contributes to partial PoWs, the victim pool is hard to detect the attacking behavior in a timely manner while it may figure out the malicious attack in a long run as its real income would be less than that brought by the estimated mining power from partial PoWs. Thus, the victim pool may fight back in the subsequent mining process. This sort of interactions between two mining pools can exactly be considered as a two-player simultaneous game. In the case of repeated interactions between them, the game turns to be iterated.

Generally speaking, the action of a pool mining without attacking another pool is defined as *cooperation* (c), and mining with attacking is denoted as *defection* (d). For differentiation, we name the two pools as pool 1 and pool 2, and their actions are denoted as a_1 and a_2, respectively. Thus we have four possible game results $a_1a_2 = (cc, cd, dc, dd)$.

Denote by m_1 and m_2 the registered mining power of pool 1 and pool 2, respectively. As the probability of finding a new block by a pool equals the ratio of the pool's effective mining power to the total mining power in the whole system, which is assumed to be m, the revenue of each pool is proportional to the share of its mining power, i.e., $\frac{m_1}{m}$ for pool 1 and $\frac{m_2}{m}$ for pool 2. For simplicity, we regard this mining power share as the payoff in the game. Thus, the above values are the payoffs of the two pools when both cooperate. When pool 2 sabotages pool 1 with an amount of x_2 malicious mining power but pool 1 remains cooperative, the victim's effective mining power becomes $m_1 - x_2$. Because the malicious attacking mining power is only used to solve partial PoWs rather than any full PoW, the total mining power of the system is also reduced by x_2. Thus, pool 1's payoff decreases to $\frac{m_1-x_2}{m-x_2}$, but pool 2 obtains a higher payoff as $\frac{m_2}{m-x_2}$. Similarly, when pool 1 chooses defection by attacking pool 2 through infiltrating x_1 amount of mining power to carry out the block withholding attack but pool 2 behaves cooperatively, the effective mining power of pool 2 for mining a full block decreases to $m_2 - x_1$ and its share of mining power in the whole system becomes $\frac{m_2-x_1}{m-x_1}$, while that of the attacker, i.e., pool 1, increases to $\frac{m_1}{m-x_1}$. If both pools are defective, i.e., attacking each other, their payoffs turn to be $\frac{m_1-x_2}{m-x_1-x_2}$ and $\frac{m_2-x_1}{m-x_1-x_2}$. We summarize the above four situations with different payoffs of the two pools in Table 1 and denote the payoff vectors of pool 1 and pool 2 respectively as,

$$\begin{aligned}\mathbf{S}_1 &= (S_1^1, S_1^2, S_1^3, S_1^4)\\ &= (\frac{m_1}{m}, \frac{m_1 - x_2}{m - x_2}, \frac{m_1}{m - x_1}, \frac{m_1 - x_2}{m - x_1 - x_2}),\end{aligned}$$

$$\begin{aligned}\mathbf{S}_2 &= (S_2^1, S_2^2, S_2^3, S_2^4)\\ &= (\frac{m_2}{m}, \frac{m_2}{m - x_2}, \frac{m_2 - x_1}{m - x_1}, \frac{m_2 - x_1}{m - x_1 - x_2}),\end{aligned}$$

following the order of the game results $a_1a_2 = (cc, cd, dc, dd)$.

Table 1: Payoff Matrix.

Pool 1 \ Pool 2	Cooperation	Defection
Cooperation	$\frac{m_1}{m}, \frac{m_2}{m}$	$\frac{m_1-x_2}{m-x_2}, \frac{m_2}{m-x_2}$
Defection	$\frac{m_1}{m-x_1}, \frac{m_2-x_1}{m-x_1}$	$\frac{m_1-x_2}{m-x_1-x_2}, \frac{m_2-x_1}{m-x_1-x_2}$

Note that the infiltrating mining power from the attacker should be less than that of the victim pool; thus we have constraints $0 < x_1 < m_2$ and $0 < x_2 < m_1$. On the other hand, according to the stability requirement of the cryptocurrency system, we assume that no pool can control the majority of the mining power, i.e., $m_1, m_2 < \frac{m}{2}$.

Given the above two-player simultaneous game, it is clear that the payoff of each side is jointly determined by the actions of both players. And when the game is repeated round-by-round, a player's individual choice of action is also affected by that of its opponent. However, the ZD strategy proposed in [8] provides us new inspirations to study the interactions between two players, where one can dominantly formulate a linear relationship between the expected payoffs of both sides, and further unilaterally set the expected payoff of itself or its opponent, no matter what action of the opponent is. Such nice features of ZD motivate us to design ZD-based strategies to drive both players cooperate. To proceed, we need to answer the following questions: who can adopt the ZD strategy? individually or simultaneously? what can the ZD player achieve? In the following we study the ZD strategy under the block withholding attack scenario in Blockchain and analyze the attacking results between two mining pools.

4 INDIVIDUAL ZD STRATEGY ANALYSIS

In this section, we investigate the problem of identifying which player in the above pool game can individually adopt the ZD strategy. As mentioned before, the game result in the previous round can influence the actions of both sides. Thus we define the strategies of the two pools as the conditional probabilities of cooperation under different outcomes of the last round.

Definition 4.1. The strategy of pool 1 is $\mathbf{p} = (p_1, p_2, p_3, p_4)$, where each element is the probability of pool 1 choosing cooperation when the game result of the previous round is $a_1a_2 = (cc, cd, dc, dd)$.

Definition 4.2. The strategy of pool 2 is $\mathbf{q} = (q_1, q_2, q_3, q_4)$, where each element is the probability of pool 2 choosing cooperation when the game result of the previous round is $a_2a_1 = (cc, cd, dc, dd)$.

The probabilities of the two pools being defective under different previous game results are $1-p_i$ and $1-q_i, i \in \{1,2,3,4\}$, respectively. In light of this, one can derive the Markov state transition matrix of the game as follows:

$$\mathbf{M} = \begin{bmatrix} p_1q_1 & p_1(1-q_1) & (1-p_1)q_1 & (1-p_1)(1-q_1) \\ p_2q_3 & p_2(1-q_3) & (1-p_2)q_3 & (1-p_2)(1-q_3) \\ p_3q_2 & p_3(1-q_2) & (1-p_3)q_2 & (1-p_3)(1-q_2) \\ p_4q_4 & p_4(1-q_4) & (1-p_4)q_4 & (1-p_4)(1-q_4) \end{bmatrix},$$

where each element $M_{ij}, i,j \in \{1,2,3,4\}$, denotes the probability of the game outcome in the current round being $a_1a_2 \in \{cc, cd, dc, dd\}$ given the previous outcome of $a_1'a_2' \in \{cc, cd, dc, dd\}$. For example, $M_{12} = p_1(1-q_1)$ is the probability of the game state in the current round $a_1a_2 = cd$ when that in the last round is $a_1'a_2' = cc$.

Suppose that the stable vector of the above Markov matrix is $\mathbf{v}$; then one can calculate the expected payoffs of the two pools as:

$$E_1 = \frac{\mathbf{v}\cdot \mathbf{S}_1}{\mathbf{v}\cdot \mathbf{1}},\ E_2 = \frac{\mathbf{v}\cdot \mathbf{S}_2}{\mathbf{v}\cdot \mathbf{1}}, \tag{1}$$

where $\mathbf{1}$ is the vector with four elements of 1.

At the stable state, there exists $\mathbf{v}^{\mathrm{T}}\mathbf{M} = \mathbf{v}^{\mathrm{T}}$. Denote by $\mathbf{M'} = \mathbf{M} - \mathbf{I}$; then we have $\mathbf{v}^{\mathrm{T}}\mathbf{M'} = \mathbf{0}$.On the other hand, by applying Cramer's rule on $\mathbf{M'}$ we get $\mathrm{Adj}(\mathbf{M'})\mathbf{M'} = \det(\mathbf{M'})\mathbf{I} = 0$. According to the above two equations, we can conclude that each row of $\mathrm{Adj}(\mathbf{M'})$ is proportional to the stable vector $\mathbf{v}$. Thus, for any vector $\mathbf{y} = (y_1, y_2, y_3, y_4)$, its dot product with $\mathbf{v}$ can be written into the following equation after the elementary column transformation on matrix $\mathbf{M'}$ by adding the first column to the second and third columns,

$$\mathbf{v}\cdot\mathbf{y} = \det\begin{bmatrix} p_1q_1 - 1 & p_1 - 1 & q_1 - 1 & y_1 \\ p_2q_3 & p_2 - 1 & q_3 & y_2 \\ p_3q_2 & p_3 & q_2 - 1 & y_3 \\ p_4q_4 & p_4 & q_4 & y_4 \end{bmatrix}. \tag{2}$$

It is worthy of noting that the second column of the above matrix is only dependent on the strategy of pool 1, which is denoted as $\tilde{\mathbf{p}} = (p_1-1, p_2-1, p_3, p_4)^{\mathrm{T}}$; similarly, the third column is only related to the strategy of pool 2, denoted as $\tilde{\mathbf{q}} = (q_1-1, q_3, q_2-1, q_4)^{\mathrm{T}}$.

Thus, given constant parameters α, β, and γ, according to (1), one can calculate a linear combination of the expected payoffs of the two pools as

$$\alpha E_1 + \beta E_2 + \gamma = \frac{\mathbf{v}\cdot(\alpha\mathbf{S}_1 + \beta\mathbf{S}_2 + \gamma\mathbf{1})}{\mathbf{v}\cdot\mathbf{1}}.$$

As indicated by (2), we can find that when $\tilde{\mathbf{p}}$ or $\tilde{\mathbf{q}}$ is proportional to the last column, i.e., $\alpha\mathbf{S}_1 + \beta\mathbf{S}_2 + \gamma\mathbf{1}$, the above equation turns to be zero as the determinant in the numerator is vanished. In other words, when the strategy of pool 1 $\mathbf{p}$ satisfies $\tilde{\mathbf{p}} = \chi(\alpha\mathbf{S}_1 + \beta\mathbf{S}_2 + \gamma\mathbf{1}),\ (\chi \neq 0)$, or that of pool 2 $\mathbf{q}$ complies with $\tilde{\mathbf{q}} = \chi(\alpha\mathbf{S}_1 + \beta\mathbf{S}_2 + \gamma\mathbf{1}),\ (\chi \neq 0)$, we have

$$\alpha E_1 + \beta E_2 + \gamma = 0. \tag{3}$$

Then the corresponding strategy is therefore named *zero-determinant* (ZD) strategy. Particularly, the ZD player (the ZD-strategy adopter), either pool 1 or pool 2, can specifically set $\alpha = 0$ or $\beta = 0$ to enforce $E_2 = -\frac{\gamma}{\beta}$ or $E_1 = -\frac{\gamma}{\alpha}$, as long as their strategies are meaningful, i.e., $p_i, q_i \in [0,1],\ i \in \{1,2,3,4\}$. In other words, each player may be able to unilaterally set the expected payoff of its opponent or itself with the help of the powerful ZD strategy.

In the following, we inspect the possibility of any pool being a ZD player individually and reveal the corresponding conditions.

4.1 Pool 1 Using the ZD Strategy

We first examine the potential of pool 1 adopting the ZD strategy to set the expected payoff of the opponent and its own.

4.1.1 *Pool 1 Sets Pool 2's Expected Payoff.*

Theorem 4.3. *When the infiltrating mining powers of the two pools satisfy* $\frac{x_1}{x_2} > \frac{m_2}{m-m_2}$, *pool 1 can utilize the ZD strategy to independently set the expected payoff of pool 2 as* $E_2 = \frac{(1-p_1)S_2^4 + p_4S_2^1}{1-p_1+p_4}$.

Proof. When pool 1 uses the ZD strategy to control pool 2's expected payoff as $E_2 = -\frac{\gamma}{\beta}$ with $\alpha = 0$, the specific ZD strategy of pool 1, i.e., $\mathbf{p}$, should satisfy $\tilde{\mathbf{p}} = \chi(\beta\mathbf{S}_2 + \gamma\mathbf{1}),\ (\chi \neq 0)$. Specifically,

$$\begin{cases} p_1 - 1 = \chi(\beta S_2^1 + \gamma), \\ p_2 - 1 = \chi(\beta S_2^2 + \gamma), \\ p_3 = \chi(\beta S_2^3 + \gamma), \\ p_4 = \chi(\beta S_2^4 + \gamma). \end{cases}$$

Using p_1 and p_4 to express β and γ, we have

$$\begin{cases} \beta = \dfrac{p_1 - p_4 - 1}{S_2^1 - S_2^4}, \\ \gamma = \dfrac{(1-p_1)S_2^4 + p_4S_2^1}{S_2^1 - S_2^4}. \end{cases}$$

Furthermore, we can solve p_2 and p_3,

$$\begin{cases} p_2 = \dfrac{p_1(S_2^2 - S_2^4) - (1+p_4)(S_2^2 - S_2^1)}{S_2^1 - S_2^4}, \\ p_3 = \dfrac{(1-p_1)(S_2^4 - S_2^3) + p_4(S_2^1 - S_2^3)}{S_2^1 - S_2^4}. \end{cases}$$

By examining all the payoff-related components in the above equations, we find those components in the numerators satisfy

$$\begin{aligned} S_2^2 - S_2^4 &= \frac{(m-m_2-x_2)x_1}{(m-x_2)(m-x_1-x_2)} > 0, \\ S_2^2 - S_2^1 &= \frac{m_2x_2}{m(m-x_2)} > 0, \\ S_2^4 - S_2^3 &= \frac{(m_2-x_1)x_2}{(m-x_1)(m-x_1-x_2)} > 0, \\ S_2^1 - S_2^3 &= \frac{(m-m_2)x_1}{m(m-x_1)} > 0, \end{aligned}$$

since $x_1 \in (0, m_2), x_2 \in (0, m_1)$ and $m_1 + m_2 < m$. While the denominators of p_2 and p_3, i.e.,

$$S_2^1 - S_2^4 = \frac{(m-m_2)x_1 - m_2x_2}{m(m-x_1-x_2)},$$

has no certain sign relationship.

Considering the constraint $p_i \in [0,1],\ i \in \{1,2,3,4\}$, we know that only when $S_2^1 - S_2^4 > 0$ can p_2 and p_3 have feasible solutions; otherwise, we have $p_2 \geq 1$ and $p_3 \leq 0$, which renders the strategy of pool 1 to have only one fixed solution $\mathbf{p} = (1,1,0,0)$, i.e., always cooperating or defecting. This strategy is obviously impractical to control the expected payoff of pool 2.

Thus, we have $\frac{(m-m_2)x_1-m_2x_2}{m(m-x_1-x_2)} > 0$, which leads to $\frac{x_1}{x_2} > \frac{m_2}{m-m_2}$. On the other hand, $E_2 = -\frac{\gamma}{\beta} = \frac{(1-p_1)S_2^4+p_4S_2^1}{1-p_1+p_4}$, which has a range of $[S_2^4, S_2^1]$. □

4.1.2 Pool 1 Sets Its Own Expected Payoff.

Theorem 4.4. *When the infiltrating mining powers of the two pools satisfy $\frac{x_1}{x_2} > \frac{m-m_1}{m_1}$, pool 1 can unilaterally set its own expected payff as $E_1 = \frac{(1-p_1)S_1^4+p_4S_1^1}{1-p_1+p_4}$.*

Proof. As mentioned before, pool 1 can set $E_1 = -\frac{\gamma}{\alpha}$ regardless of the strategy of pool 2 through executing the ZD strategy with $\beta = 0$, and its ZD strategy $\mathbf{p}$ needs to meet $\tilde{\mathbf{p}} = \chi(\alpha \mathbf{S}_1 + \gamma \mathbf{1})$, $(\chi \neq 0)$, i.e.,

$$\begin{cases} p_1 - 1 = \chi(\alpha S_1^1 + \gamma), \\ p_2 - 1 = \chi(\alpha S_1^2 + \gamma), \\ p_3 = \chi(\alpha S_1^3 + \gamma), \\ p_4 = \chi(\alpha S_1^4 + \gamma). \end{cases}$$

We can solve it for p_2 and p_3 as

$$\begin{cases} p_2 = \dfrac{(1+p_4)(S_1^1 - S_1^2) - p_1(S_1^4 - S_1^2)}{S_1^1 - S_1^4}, \\ p_3 = \dfrac{-(1-p_1)(S_1^3 - S_1^4) - p_4(S_1^3 - S_1^1)}{S_1^1 - S_1^4}. \end{cases}$$

As $x_1 \in (0, m_2), x_2 \in (0, m_1)$ and $m_1 + m_2 < m$, the components related to pool 1's payoff vector in the above expressions satisfy

$$\begin{aligned} S_1^1 - S_1^2 &= \frac{(m-m_1)x_2}{m(m-x_2)} > 0, \\ S_1^4 - S_1^2 &= \frac{(m_1-x_2)x_1}{(m-x_2)(m-x_1-x_2)} > 0, \\ S_1^3 - S_1^4 &= \frac{(m-m_1-x_1)x_2}{(m-x_1)(m-x_1-x_2)} > 0, \\ S_1^3 - S_1^1 &= \frac{m_1x_1}{m(m-x_1)} > 0, \end{aligned}$$

but the denominator $S_1^1 - S_1^4$ needs further discussion.

When $S_1^1 - S_1^4 > 0$, it is easy to figure out that $p_2 \geq 1, p_3 \leq 0$, leading to the single feasible point of pool 1's ZD strategy, which is obviously meaningless for our scenario. While $S_1^1 - S_1^4 < 0$ results in $p_2 \leq 1, p_3 \geq 0$, and thus $\mathbf{p}$ has more solutions so as to control its own expected payoff in a range.

In other words, the condition for pool 1 being the ZD player to set its own expected payoff is $\frac{(m-m_1)x_2-m_1x_1}{m(m-x_1-x_2)} < 0$, i.e., $\frac{x_1}{x_2} > \frac{m-m_1}{m_1}$. And its expected payoff becomes $E_1 = -\frac{\gamma}{\alpha} = \frac{(1-p_1)S_1^4+p_4S_1^1}{1-p_1+p_4} \in [S_1^1, S_1^4]$. □

4.2 Pool 2 Using the ZD Strategy

We adopt a similar analysis for pool 2 to derive the conditions on which it can employ the ZD strategy to set the expected payoff of its opponent (i.e., pool 1) and that of itself.

4.2.1 Pool 2 Sets Pool 1's Expected Payoff.

Theorem 4.5. *When $\frac{x_2}{x_1} > \frac{m_1}{m-m_1}$, pool 2 can use the ZD strategy to unilaterally set the expected payoff of pool 1 to be $E_1 = \frac{(1-q_1)S_1^4+q_4S_1^1}{1-q_1+q_4}$.*

Proof. By setting $\beta = 0$, pool 2 can enforce the expected payoff of pool 1 as $E_1 = -\frac{\gamma}{\alpha}$ when it sets the ZD strategy $\mathbf{q}$ as $\tilde{\mathbf{q}} = \chi(\alpha \mathbf{S}_1 + \gamma \mathbf{1})$. We can derive q_2 and q_3 in terms of q_1 and q_4,

$$\begin{cases} q_2 = \dfrac{-(1+q_4)(S_1^3 - S_1^1) + q_1(S_1^3 - S_1^4)}{S_1^1 - S_1^4}, \\ q_3 = \dfrac{q_4(S_1^1 - S_1^2) + (1-q_1)(S_1^4 - S_1^2)}{S_1^1 - S_1^4}. \end{cases}$$

As analyzed in Section 4.1.2, all the payoff-related components in the numerators of q_2 and q_3 , i.e., $(S_1^3 - S_1^1)$, $(S_1^3 - S_1^4)$, $(S_1^1 - S_1^2)$, and $(S_1^4 - S_1^2)$, are positive, but the sign of the denominator , i.e., $S_1^1 - S_1^4$, is not determinate. With a similar discussion, we can conclude that only when $S_1^1 - S_1^4 = \frac{(m-m_1)x_2-m_1x_1}{m(m-x_1-x_2)} > 0$, which is equivalent to $\frac{x_2}{x_1} > \frac{m_1}{m-m_1}$, can q_2 and q_3 have feasible solutions. Thus we have $E_1 = \frac{(1-q_1)S_1^4+q_4S_1^1}{1-q_1+q_4}$, ranging in $[S_1^4, S_1^1]$. □

4.2.2 Pool 2 Sets Its Own Expected Payoff.

Theorem 4.6. *When $\frac{x_2}{x_1} > \frac{m-m_2}{m_2}$, pool 2 can take advantage of the ZD strategy to unilaterally set its own expected payoff as $E_2 = \frac{(1-q_1)S_2^4+q_4S_2^1}{1-q_1+q_4}$.*

Proof. When pool 2 wants to set its own expected payoff $E_2 = -\frac{\gamma}{\beta}$ with $\alpha = 0$, the corresponding ZD strategy should be calculated by $\tilde{\mathbf{q}} = \chi(\beta \mathbf{S}_2 + \gamma \mathbf{1})$. Thus, we have

$$\begin{cases} q_2 = \dfrac{(1+q_4)(S_2^1 - S_2^3) - q_1(S_2^4 - S_2^3)}{S_2^1 - S_2^4}, \\ q_3 = \dfrac{-(1-q_1)(S_2^2 - S_2^4) - q_4(S_2^2 - S_2^1)}{S_2^1 - S_2^4}. \end{cases}$$

Since the payoff-based components are proved to be positive in Section 4.1.1, here we omit it for brevity. And it is easy to prove that if $S_2^1 - S_2^4 = \frac{(m-m_2)x_1-m_2x_2}{m(m-x_1-x_2)} < 0$, there exist feasible solutions for the above equations. That is, when the infiltrating mining powers of the two pools x_1 and x_2 satisfy $\frac{x_2}{x_1} > \frac{m-m_2}{m_2}$, pool 2 can take advantage of the ZD strategy to unilaterally adjust its own expected payoff as $E_2 = \frac{(1-q_1)S_2^4+q_4S_2^1}{1-q_1+q_4} \in [S_2^1, S_2^4]$. □

In conclusion, one can see that both players are capable of individually conducting the ZD strategy to unilaterally set the expected payoff of its opponent or itself, with each corresponding to a unique condition of the relationship between the attacking mining powers of two pools.

Note that since both pools can individually utilize the ZD strategy under specific circumstances, we are faced with the following new problem: whether it is possible for both pools taking the ZD strategies concurrently to set the expected payoff of each other or those of themselves. This will be studied in the next section.

5 SIMULTANEOUS ZD STRATEGY ANALYSIS

In this section, we analyze the potential for the two players in the pool game to simultaneously employ the ZD strategies with the objective of independently controlling the expected payoffs of their opponents or themselves.

5.1 Setting the Expected Payoffs of the Opponent

We first study the situation where both pools utilize the ZD strategies to set the expected payoff of each other, which can be settled by the following theorem.

Theorem 5.1. *When the infiltrating mining powers set by pool 1 and pool 2 for the block withholding attack satisfy*

$$\frac{m_2}{m-m_2} < \frac{x_1}{x_2} < \frac{m-m_1}{m_1},$$

the two pools can achieve the goal of using ZD strategies to set the expected payoff of the opponent at the same time.

Proof. As analyzed in Section 4.1.1, the necessary condition for pool 1 using the ZD strategy to set the expected payoff of pool 2 is $\frac{x_1}{x_2} > \frac{m_2}{m-m_2}$; and as indicated in Section 4.2.1, the corresponding condition for pool 2 to set the expected payoff of pool 1 is $\frac{x_2}{x_1} > \frac{m_1}{m-m_1}$. Combining the above two conditions, we have

$$\frac{m_2}{m-m_2} < \frac{x_1}{x_2} < \frac{m-m_1}{m_1}.$$

In order to guarantee the rationality of the above constraint, we have to ensure that the relationship between the upper bound and the lower bound satisfies $\frac{m_2}{m-m_2} < \frac{m-m_1}{m_1}$, which is equivalent to $m_1 m_2 < (m-m_1)(m-m_2)$. This relationship obviously holds since $m > 0$ and $m > m_1 + m_2$. □

As the above parameter relationships are originated from the relationships between the payoffs of the game results $a_1a_2 = cc$ and dd, i.e., $S_1^1 > S_1^4$ and $S_2^1 > S_2^4$, we have the following theorem.

Theorem 5.2. *In case of both pools simultaneously adopting ZD strategies, i.e., $\frac{m_2}{m-m_2} < \frac{x_1}{x_2} < \frac{m-m_1}{m_1}$, the game between the two pools becomes a Prisoner's Dilemma (PD).*

Proof. In order to prove that the pool game is a PD, we need to illustrate that the game satisfies the following two requirements: i) mutual defection ($a_1a_2 = dd$) is the Nash equilibrium of the game; and ii) mutual cooperation ($a_1a_2 = cc$) is the state with maximum social welfare.

The first condition implies that no matter what the opponent's action is, defection is the dominant action for both players as it brings a higher payoff than cooperation. For pool 1, this means that its payoff in state $a_1a_2 = dc$ is larger than that in $a_1a_2 = cc$, which is larger than that in $a_1a_2 = dd$, and greater than that in $a_1a_2 = cd$. As clarified in Section 4.1.2, we know that $S_1^3 - S_1^1 > 0$ and $S_1^4 - S_1^2 > 0$. Combining with the relationship $S_1^1 > S_1^4$ implied by the parameter constraint, we have

$$S_1^3 > S_1^1 > S_1^4 > S_1^2.$$

Similarly, for pool 2, it gets a higher payoff in state $a_1a_2 = cd$ than that in $a_1a_2 = cc$ which is larger than that in $a_1a_2 = dd$ and then its payoff comes to the smallest in $a_1a_2 = dc$. As demonstrated in Section 4.1.1, we have $S_2^2 - S_2^1 > 0$ and $S_2^4 - S_2^3 > 0$. In addition, the parameter relationship is resulted from $S_2^1 > S_2^4$; thus there exist

$$S_2^2 > S_2^1 > S_2^4 > S_2^3.$$

The second requirement indicates that the total payoff of the two pools in state $a_1a_2 = cc$ is the largest compared to the other three states,

$$S_1^1 + S_2^1 > S_1^2 + S_2^2,$$
$$S_1^1 + S_2^1 > S_1^3 + S_2^3,$$
$$S_1^1 + S_2^1 > S_1^4 + S_2^4.$$

Referring to Table 1, one can see clearly that the above relationships hold as

$$\frac{m_1+m_2}{m} > \frac{m_1+m_2-x_2}{m-x_2},$$
$$\frac{m_1+m_2}{m} > \frac{m_1+m_2-x_1}{m-x_1},$$
$$\frac{m_1+m_2}{m} > \frac{m_1+m_2-x_1-x_2}{m-x_1-x_2},$$

given $x_1 \in (0, m_2)$, $x_2 \in (0, m_1)$, and $m > m_1 + m_2$. □

In this case, during the iterated game process, both pools witness that the Nash equilibrium of the game consists of their dominant actions, i.e., $a_1a_2 = dd$, which is unfavorable when compared to the mutual cooperation $a_1a_2 = cc$, from either the perspective of individual interest as $S_1^1 > S_1^4$, $S_2^1 > S_2^4$, or that of the group revenue because of $S_1^1 + S_2^1 > S_1^4 + S_2^4$. Therefore, both players have the incentive to let the game result be close to mutual cooperation as much as possible for the optimum social welfare. However, as the players choose the actions at the same time at each round with no accurate information of the opponent's intention, it is risky for any player to rashly select cooperation because it may suffer from the lowest payoff once its opponent performs defection.

On the other hand, as we summarized in Theorem 5.1, both pool 1 and pool 2 are capable of employing the powerful ZD strategy under the parameter conditions mentioned above to achieve the effective control of the opponent's expected payoff, no matter what action of the opponent is. Thus, it inspires us to think about whether it is possible for any player using the ZD strategy to control the game result as a whole and elicit the best result where the social welfare is maximized, without worrying about any trick from the opponent. In the following, we take pool 1 as an example to illustrate how to solve this problem.

As mentioned in Section 4, when pool 1 sets its strategy $\mathbf{p}$ satisfying $\tilde{\mathbf{p}} = \chi(\alpha \mathbf{S}_1 + \beta \mathbf{S}_2 + \gamma \mathbf{1})$, $(\chi \neq 0)$, there exists $\alpha E_1 + \beta E_2 + \gamma = 0$. In light of this, pool 1 can set $\alpha = \beta = 1$ to get the social welfare as

$$E_{all} = E_1 + E_2 = -\gamma. \tag{4}$$

In order to maximize the social welfare leveraging only its own power, pool 1 can make use of the ZD strategy, which can be formulated as

$$\max E_{all} = E_1(\mathbf{p}, \mathbf{q}) + E_2(\mathbf{p}, \mathbf{q}), \forall \mathbf{q},$$
$$\text{s.t.} \begin{cases} \mathbf{0} \leq \mathbf{p} \leq \mathbf{1}, \\ E_1 + E_2 + \gamma = 0. \end{cases}$$

As shown in (4), the above optimization problem is equivalent to

$$\min \gamma, \quad \text{s.t.} \begin{cases} \mathbf{0} \le \mathbf{p} \le \mathbf{1}, \\ \tilde{\mathbf{p}} = \chi(\mathbf{S}_1 + \mathbf{S}_2 + \gamma \mathbf{1}), \\ \chi \ne 0. \end{cases}$$

Because the sign of χ affects the calculation process, we consider the following two cases.

Case 1: $\chi > 0$. On one hand, as $\mathbf{p} \ge \mathbf{0}$, we can infer the lower bound of γ as

$$\gamma_{min} = \max(R_i),\ i \in \{1,2,3,4\},$$

where R_i is

$$R_i = \begin{cases} -S_1^i - S_2^i - \dfrac{1}{\chi}, & i = 1,2, \\ -S_1^i - S_2^i, & i = 3,4. \end{cases}$$

On the other hand, it is necessary to guarantee that $\mathbf{p} \le \mathbf{1}$, which leads to the upper bound of γ,

$$\gamma_{max} = \min(R_j),\ j \in \{5,6,7,8\}.$$

And R_j is calculated by

$$R_j = R_{i+4} = \begin{cases} -S_1^i - S_2^i, & i = 1,2, \\ -S_1^i - S_2^i + \dfrac{1}{\chi}, & i = 3,4. \end{cases}$$

As only when $\gamma_{min} < \gamma_{max}$ can γ have a feasible solution, we need to ensure that $\max(R_i) < \min(R_j)$, $i \in \{1,2,3,4\}$, $j \in \{5,6,7,8\}$, holds. Thus, once there exists any $\chi^* > 0$ that can meet this requirement, we can get the minimum value of γ as

$$\gamma_{min} = \max\{-S_1^1 - S_2^1 - \frac{1}{\chi^*}, -S_1^2 - S_2^2 - \frac{1}{\chi^*}, -S_1^3 - S_2^3, -S_1^4 - S_2^4\}.$$

Case 2: $\chi < 0$. Similarly, we first consider the constraint $\mathbf{p} \ge \mathbf{0}$; then the upper bound of γ can be derived as

$$\gamma_{max} = \min(R_i),\ i \in \{1,2,3,4\}.$$

While in light of $\mathbf{p} \le \mathbf{1}$, we have

$$\gamma_{min} = \max(R_j),\ j \in \{5,6,7,8\}.$$

And it is also true that only $\gamma_{min} < \gamma_{max}$ can bring feasible solutions to γ, which is equivalent to $\max(R_j) < \min(R_i)$, $i \in \{1,2,3,4\}$, $j \in \{5,6,7,8\}$. Again, if there exists a specific $\chi^* < 0$ satisfying this condition, the minimum γ is given by

$$\gamma_{min} = \max\{-S_1^1 - S_2^1, -S_1^2 - S_2^2, -S_1^3 - S_2^3 + \frac{1}{\chi^*}, -S_1^4 - S_2^4 + \frac{1}{\chi^*}\}.$$

No matter which case is met, as long as the minimum value of γ is derived as γ_{min} with a certain χ^*, the ZD strategy of pool 1 is

$$p_i = \begin{cases} \chi^*(S_1^i + S_2^i + \gamma_{min}) + 1, & i = 1,2, \\ \chi^*(S_1^i + S_2^i + \gamma_{min}), & i = 3,4. \end{cases}$$

Thus, the solution of pool 1 using the ZD strategy to maximize the social welfare exists, which can never be influenced by the action and strategy of its opponent in the pool game.

It is worth noting that as pool 2 is also able to utilize the ZD strategy under the same condition of $\frac{m_2}{m-m_2} < \frac{x_1}{x_2} < \frac{m-m_1}{m_1}$, one can use the same calculation method presented above to deduce the maximum social welfare as well as the corresponding ZD strategy. Thus we omit it here for brevity.

In summary, it is practical for both pools simultaneously using the ZD strategies to unilaterally set the expected payoff of each other under certain circumstance, where a PD game is formulated and each side is capable of taking advantage of the ZD strategy to enforce the best game result for both sides with no need to consider the strategy of the other side.

5.2 Setting the Expected Payoffs of Themselves

Now we turn to investigate the case of two pools relying on the ZD strategies to control their own expected payoffs concurrently.

Theorem 5.3. *It is impossible for two pools simultaneously using the ZD strategies to adjust their own expected payoffs.*

Proof. As shown in Section 4.1.2, if pool 1 aims to set its own expected payoff with the help of the ZD strategy, it is required that its attacking mining power must be higher than that of pool 2, i.e., $\frac{x_1}{x_2} > \frac{m-m_1}{m_1}$. On the other hand, as elaborated in Section 4.2.2, the condition for pool 2 utilizing the ZD strategy to control its own expected payoff is $\frac{x_2}{x_1} > \frac{m-m_2}{m_2}$. Therefore, if both pools implement the ZD strategies so as to determine their own expected payoffs at the same time, their infiltrating mining powers should satisfy the following constraint,

$$\frac{m-m_1}{m_1} < \frac{x_1}{x_2} < \frac{m_2}{m-m_2}.$$

However, when we scrutinize the relationship of the upper and lower bounds in the above formula, we find that $\frac{m-m_1}{m_1} < \frac{m_2}{m-m_2}$ can never be met as $m(m-m_1-m_2) < 0$ never holds with $m > 0$ and $m > m_1 + m_2$. That is, $\frac{x_1}{x_2}$ has no feasible solution under the above constraint. Thus, one can conclude that the two pools cannot simultaneously employ the ZD strategies to determine the expected payoffs for themselves in the game. □

According the theorem presented above, we know that if any pool i, $i \in \{1,2\}$ in the game tries to utilize the ZD strategy to set its own expected payoff, its attacking power x_i should be large enough to meet $x_i > \frac{m-m_i}{m_i} x_{-i}$, where x_{-i} denotes the attacking power of its opponent. And in this case, the opponent is absolutely not qualified to use the ZD strategy to control the expected payoff of either the ZD player or itself.

In light of this, it is evident that the ZD player can dominate the pool game over its opponent. Then one may concern about how bad the situation of the weak side (i.e., the non-ZD player) can be when it does not have the ability to execute the ZD strategy. Without loss of generality, we assume that it is pool 1 who can adopt the ZD strategy to set its own expected payoff.

As mentioned above, when the infiltrating mining power from pool 1 to attack pool 2 is larger than that in the opposite direction, i.e., $x_1 > \frac{m-m_1}{m_1} x_2$, the expected payoff of pool 1 can be set to $E_1 = \frac{(1-p_1)S_1^4 + p_4 S_1^1}{1-p_1+p_4}$, which is clearly a weighted sum of S_1^1 and S_1^4. As an intelligent and profit-driven player, pool 1 with the powerful ZD strategy inclines to set the highest expected payoff for itself, i.e., $E_1 = S_1^4$, which can be achieved with the strategy $p_1 \ne 1$ and

$p_4 = 0$. On the other hand, as shown in Section 4.1.2, we have

$$p_2 = \frac{(1+p_4)(S_1^1 - S_1^2) - p_1(S_1^4 - S_1^2)}{S_1^1 - S_1^4},$$

$$p_3 = \frac{-(1-p_1)(S_1^3 - S_1^4) - p_4(S_1^3 - S_1^1)}{S_1^1 - S_1^4}.$$

Obviously, p_2 and p_3 have feasible solutions only when $p_1 \to 1$ and $p_4 \to 0$ and the corresponding values are $p_2 \to 1$ and $p_3 \to 0$. Thus given $p_1 \neq 1$ and $p_4 = 0$, we have $p_2 = \frac{(S_1^1 - S_1^2) - p_1(S_1^4 - S_1^2)}{S_1^1 - S_1^4}$, $p_3 = \frac{-(1-p_1)(S_1^3 - S_1^4)}{S_1^1 - S_1^4}$. In order to make $p_2, p_3 \in [0,1]$, and further considering constraint $p_1 \neq 1$, one can derive the range of p_1 to be

$$\max\left(\frac{S_1^1 - S_1^2}{S_1^4 - S_1^2}, 1 - \frac{S_1^4 - S_1^2}{S_1^3 - S_1^4}\right) \leq p_1 < 1.$$

Thus, for any p_1^* satisfying the above constraint, the ZD strategy for pool 1 is fixed as $\mathbf{p}^* = (p_1^*, \frac{(S_1^1 - S_1^2) - p_1^*(S_1^4 - S_1^2)}{S_1^1 - S_1^4}, \frac{-(1-p_1^*)(S_1^3 - S_1^4)}{S_1^1 - S_1^4}, 0)$. In this case, pool 2's expected payoff E_2 is only dependent on its own strategy $\mathbf{q}$. For any rational player under this circumstance, the best response strategy is to maximize its own expected payoff to fight against the dominant ZD player. Thus, pool 2 can take advantage of any existing algorithm for multivariable function optimization (e.g., genetic algorithm), so as to derive its best strategy $\mathbf{q}^*$ and the corresponding maximum expected payoff E_2^*.

To explicitly study the outcomes of pool 2, we assume that the total mining power in the distributed network is $m = 1$, and the mining powers of the two pools are $m_1 = 0.1$ and $m_2 = 0.2$. Note that other values of m_1 and m_2 are also studied, which result in the same conclusions. Then the payoff vectors of the two players are $\mathbf{S}_1 = (0.1, \frac{0.1-x_2}{1-x_2}, \frac{0.1}{1-x_1}, \frac{0.1-x_2}{1-x_1-x_2})$ and $\mathbf{S}_2 = (0.2, \frac{0.2}{1-x_2}, \frac{0.2-x_1}{1-x_1}, \frac{0.2-x_1}{1-x_1-x_2})$. As the attacking mining power from pool 1 should be large enough to meet the condition $x_1 > 9x_2$, we assume that $x_1 = 0.19$, which is close to its upper bound $m_2 = 0.2$. Then the range of x_2 is $0 < x_2 < 0.0211$. Besides, according to the above analysis, p_1 should be in an appropriate range so that $p_2, p_3 \in [0,1]$. In this case, it turns out to be

$$\max(\theta(x_2), \delta(x_2)) \leq p_1 < 1,$$

where $\theta(x_2) = \frac{900x_2^2 - 729x_2}{190x_2 - 19}$ and $\delta(x_2) = \frac{143900x_2 - 1539}{71000x_2}$ are the limitations corresponding to $p_2, p_3 \in [0,1]$. As shown in Fig. 1, the lower bound of p_1 increases with $x_2 \in (0, 0.0211)$; thus we can set $p_1 = \max(\theta(0.0211), \delta(0.0211)) = 0.9995$. Then p_2 and p_3 can be calculated accordingly, which turn out to be functions of x_2.

E_2 can be calculated as a function of $\mathbf{q}$ and x_2. Here we omit the detailed expression as it is over-lengthy. By solving $\frac{\partial E_2}{\partial \mathbf{q}} = 0$, we get the stationary point $\mathbf{q}^* = (q_1^*, 0, 1, 0)$, where q_1^* relies on x_2 and its trend with $x_2 \in (0, 0.0211)$ is demonstrated in Fig. 2. It is obvious that the value of q_1^* is always larger than 1 in the domain of x_2, which is out of the range of a meaningful $\mathbf{q}$. Thus, we can infer that the maximum value of E_2 can only be obtained in the 16 endpoints, which turns out that both $\mathbf{q} = (1,1,0,1)$ and $\mathbf{q} = (1,1,1,1)$ can achieve the optimum $E_2 = -\frac{700x_2+1}{100x_2-81}$. Based on this, one can conclude that under the dominance of the ZD player setting the highest expected payoff for itself, the best response strategy for the weak side is to cooperate as much as possible. Further, it is easy to observe that E_2 increases with $x_2 \in (0, 0.0211)$ and always less than 0.2, which brings us another conclusion that even if the non-ZD player exerts all its infiltrating mining power to attack against the ZD player, it still suffers from a lower revenue than that in $a_1a_2 = cc$.

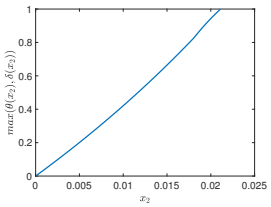


Figure 1: p_1's lower bound.

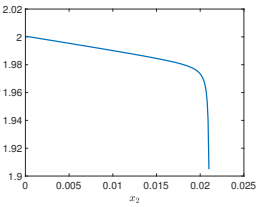


Figure 2: q_1^*.

6 EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the proposed social welfare maximization mechanism when both players are simultaneously available to adopt the ZD strategies for payoff control of each other as well as the performance of the non-ZD player in the situation where only one player can use the ZD strategy to set its own expected payoff.

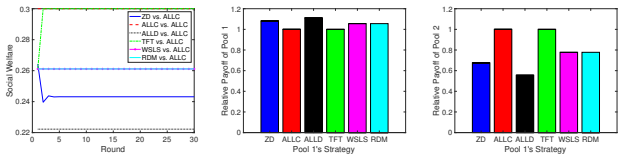
We first verify the effectiveness of the ZD based social welfare maximization scheme proposed in Section 5.1 with the parameter settings of $m = 1$, $m_1 = 0.1$, and $m_2 = 0.2$. As implied in Theorem 5.1, the values of x_1 and x_2 in this case must satisfy $\frac{1}{4} < \frac{x_1}{x_2} < 9$; thus we fix $x_1 = 0.1$ and $x_2 = 0.05$. Note that we also conduct experiments with other parameter settings and obtain similar results, which are omitted for brevity. Note that we simulate the game between the two pools for 100 rounds to reach the stable state.

Specifically, we compare the social welfare of the game when pool 1 adopts the ZD strategy as well as five other classical strategies: all-cooperation (ALLC, $\mathbf{p} = (1,1,1,1)$), all-defection (ALLD, $\mathbf{p} = (0,0,0,0)$), tit-for-tat (TFT, $\mathbf{p} = (1,0,1,0)$), win-stay-lose-shift (WSLS, $\mathbf{p} = (1,0,0,1)$), and random (RDM, $\mathbf{p} = (0.5,0.5,0.5,0.5)$). Pool 2 also adopts these six strategies; thus we have 36 strategy combinations. For each combination, we report the evolution of the social welfare as the game proceeds and the relative payoffs of the two pools in the corresponding stable state. The results are illustrated in Figs. 3 to 8, with each showing a case when the strategy of pool 2 is fixed while that of pool 1 varies, so as to demonstrate whether the proposed ZD strategy is the best decision for pool 1.

Fig. 3 reports the results when pool 2 adopts ALLC. It is obvious that a reasonable pool 1 can employ either ALLC or TFT to provide the highest social welfare and fairly high payoffs for both sides. This corresponds to the case in state $a_1a_2 = cc$; and ZD is only better than ALLD. When pool 2 adopts the ALLD strategy (Fig. 4), it seems that there exists no perfect strategy for pool 1. Reluctantly, WSLS and RDM perform slightly better than the other four strategies for pool 1 since ALLD and TFT result in the lowest social welfare; and although ALLC and ZD can bring a higher social welfare, they cause the lowest payoff for pool 1. This result is understandable as a stubbornly defective player can never be beaten by anyone. For a TFT pool 2 (Fig. 5), an ALLC pool 1 is the best as it can elicit c

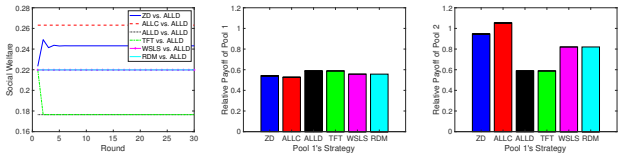
from the TFT player, and the ZD strategy performs sub-optimally. In Fig. 6 where pool 2 adopts WSLS, one can see that WSLS is the optimal strategy for pool 1 as the two players with WSLS can definitely lead to mutual cooperation, and ZD is better than the other four strategies with a relatively high social welfare and fair payoffs. When pool 2 adopts the RDM strategy (Fig. 7), it is clear that ZD outperforms others as it brings a relatively large social welfare and fairly good payoffs for both sides. Finally, when pool 2 employs the ZD strategy shown in Fig. 8, ZD and TFT are equally good from the perspectives of both social welfare and individual payoffs. From the above observations, we can derive the following conclusions:

(1) When pool 2 adopts two stubborn strategies, i.e., ALLC and ALLD, the proposed ZD strategy has no advantage over the other five classical strategies for pool 1.
(2) When pool 2 utilizes the two adaptive strategies, i.e., TFT and WSLS, ZD is a sub-optimal choice for pool 1 as ALLC and WSLS can evoke cooperation of pool 2 when it takes TFT and WSLS, respectively, and finally achieve mutual cooperation.
(3) When pool 2 utilizes the RDM or ZD strategy, ZD turns out to be attractive for pool 1 since it can bring the highest social welfare and fair payoffs.



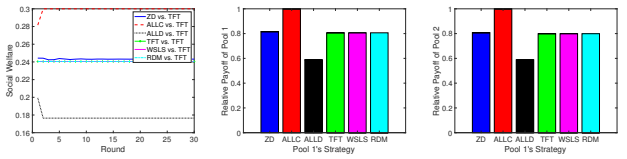
(a) Social welfare. (b) Pool 1's payoff. (c) Pool 2's payoff.

Figure 3: Case of pool 2 adopting the ALLC strategy.



(a) Social welfare. (b) Pool 1's payoff. (c) Pool 2's payoff.

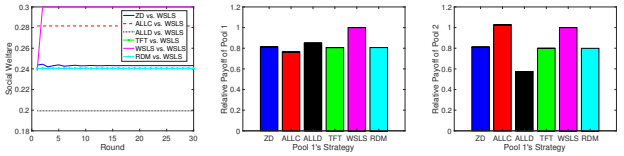
Figure 4: Case of pool 2 adopting the ALLD strategy.



(a) Social welfare. (b) Pool 1's payoff. (c) Pool 2's payoff.

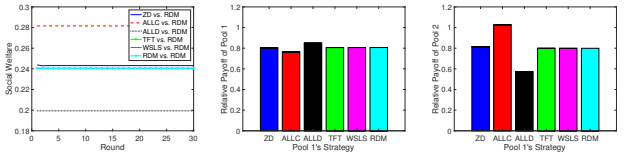
Figure 5: Case of pool 2 adopting the TFT strategy.

On the other hand, to explicitly explore the value of the ZD strategy, we plot the time-varying social welfare in Fig. 9 when



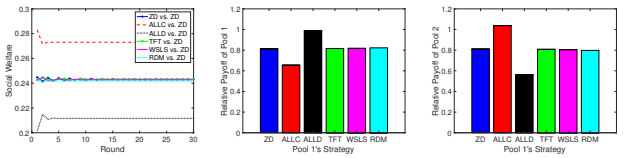
(a) Social welfare. (b) Pool 1's payoff. (c) Pool 2's payoff.

Figure 6: Case of pool 2 adopting the WSLS strategy.



(a) Social welfare. (b) Pool 1's payoff. (c) Pool 2's payoff.

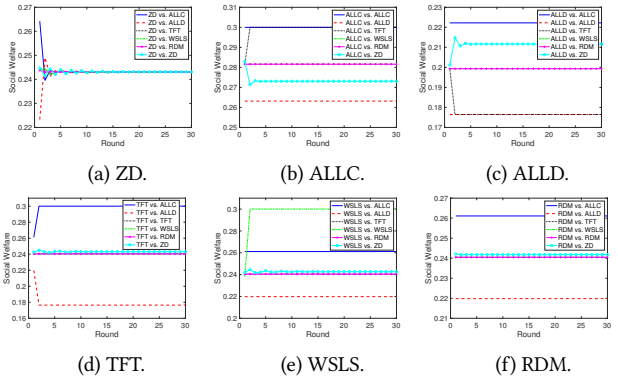
Figure 7: Case of pool 2 adopting the RDM strategy.



(a) Social welfare. (b) Pool 1's payoff. (c) Pool 2's payoff.

Figure 8: Case of pool 2 adopting the ZD strategy.

pool 1 fixes its strategy while pool 2 varies by taking six different strategies. It is easy to see that only when pool 1 uses ZD can the social welfare of the game stay at a fixed value of 0.243 no matter what strategy pool 2 employs, which is explicitly larger than that in state $a_1a_2 = dd$ of 0.176; the other five classical strategies have no such property, and the social welfare in these cases are jointly determined by the strategies of the two pools.



(a) ZD. (b) ALLC. (c) ALLD.

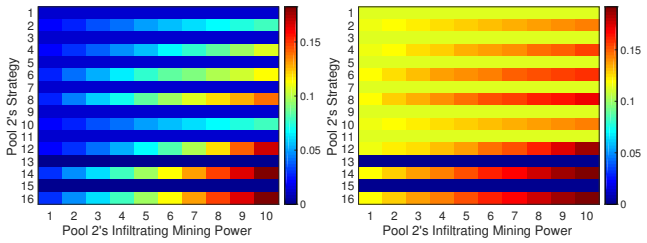
(d) TFT. (e) WSLS. (f) RDM.

Figure 9: Social welfare with a certain strategy of pool 1.

Next, we examine the results of the non-ZD player when only one player is capable of adopting the ZD strategy, so as to verify the conclusion we present before. Basically, we use the same values

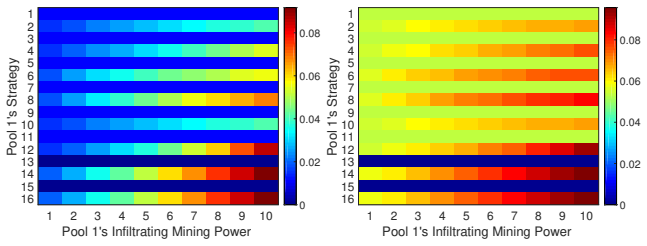
of m, m_1, and m_2 as in Section 5.2, and set the values of x_1, x_2 to meet the conditions of $\frac{x_1}{x_2} > 9$ for pool 1 and $\frac{x_2}{x_1} > 4$ for pool 2, such that each can adopt the ZD strategy to control its own payoff.

Specifically, as analyzed in Section 5.2, the optimum payoff of the non-ZD player is obtained at one of the 16 end-point strategies, which are denoted by the binary notation of 0 to 15, i.e., $(0, 0, 0, 0)$, $(0, 0, 0, 1)$, $(0, 0, 1, 0)$, $\cdots$, $(1, 1, 1, 1)$. And the range of the non-ZD player's infiltrating mining power (attacking parameter) is divided into 10 intervals. In Fig. 10, we plot the expected payoffs of pool 2 at the 16 endpoints when the value of its own attacking parameter x_2 changes while pool 1 adopts the ZD strategy to set its own payoff to be the highest. In the case of $x_1 = 0.19$, the range of x_2 is $(0, 0.0211)$; while when $x_1 = 0.10$, $x_2 \in (0, 0.0111)$. One can easily figure out that the more cooperative pool 2 is, the higher its expected payoff; and for a specific end-point strategy, the higher the attacking parameter of pool 2, the higher the payoff; but the payoff is always less than 0.2. We also study the case of pool 2 adopting the ZD strategy to set the highest payoff for itself, where x_2 is fixed to 0.09 or 0.05 to meet the condition of $x_2 < m_1$, and accordingly, the respective range of x_1 is (0,0.0225) and (0,0.0125). As shown in Fig. 11, the changes of the payoffs of pool 1 with x_1 and x_2 under different strategies demonstrate a similar conclusion.



(a) $x_1 = 0.19$. (b) $x_1 = 0.10$.

Figure 10: Pool 2's payoffs at the end points change with x_1 and x_2 when pool 1 adopts the ZD strategy.



(a) $x_2 = 0.09$. (b) $x_2 = 0.05$.

Figure 11: Pool 1's payoffs at the end points change with x_1 and x_2 when pool 2 adopts the ZD strategy.

7 CONCLUSIONS

In this paper, we focus on the block withholding attack among mining pools in Bitcoin. Different from the state-of-the-art research that studies the mutual attack among pools relying on the traditional game theory, we conduct an analysis from special perspective with the ZD strategy by which the ZD player can achieve the unilateral control of the expected payoff. Specifically, we model the block withholding attack between any two pools as a two-player simultaneous game. Based on this game model, we investigate the conditions under which any pool can individually adopt the ZD strategy and two pools can concurrently employ it. Through theoretical derivation and numerical analysis, we demonstrate the effectiveness of the ZD strategy in block withholding attacks. To the best of our knowledge, we are the first to use the ZD strategy to analyze the block withholding attack among mining pools.

8 ACKNOWLEDGMENTS

This work is partially supported by the US NSF under grants IIS-1741279 and CNS-1704397, and the National Natural Science Foundation of China under grants 61772080 and 61832012.

REFERENCES

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
[2] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980*, 2011.
[3] Nicolas T Courtois and Lear Bahack. On subversive miner strategies and block withholding attack in bitcoin digital currency. *arXiv preprint arXiv:1402.1718*, 2014.
[4] Samiran Bag, Sushmita Ruj, and Kouichi Sakurai. Bitcoin block withholding attack: Analysis and mitigation. *IEEE Transactions on Information Forensics and Security*, 12(8):1967–1978, 2017.
[5] Loi Luu, Ratul Saha, Inian Parameshwaran, Prateek Saxena, and Aquinas Hobor. On power splitting games in distributed computation: The case of bitcoin pooled mining. In *Computer Security Foundations Symposium (CSF), 2015 IEEE 28th*, pages 397–411. IEEE, 2015.
[6] Ittay Eyal. The miner's dilemma. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 89–103. IEEE, 2015.
[7] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 195–209. ACM, 2017.
[8] William H Press and Freeman J Dyson. Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 109(26):10409–10413, 2012.
[9] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7):95–102, 2018.
[10] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2016.
[11] Deepak K Tosh, Sachin Shetty, Xueping Liang, Charles A Kamhoua, Kevin A Kwiat, and Laurent Njilla. Security implications of blockchain cloud with analysis of block withholding attack. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 458–467. IEEE Press, 2017.
[12] Ghassan O Karame, Elli Androulaki, and Srdjan Capkun. Double-spending fast payments in bitcoin. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 906–917. ACM, 2012.
[13] Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden. Incentive compatibility of bitcoin mining pool reward functions. In *International Conference on Financial Cryptography and Data Security*, pages 477–498. Springer, 2016.
[14] Samiran Bag and Kouichi Sakurai. Yet another note on block withholding attack on bitcoin mining pools. In *International Conference on Information Security*, pages 167–180. Springer, 2016.