Fee-Free Pooled Mining for Countering Pool-Hopping Attack in Blockchain

Hongwei Shi, Shengling Wang, *Senior Member, IEEE*, Qin Hu, Xiuzhen Cheng, *Fellow, IEEE*, Junshan Zhang, *Fellow, IEEE*, and Jiguo Yu, *Senior Member, IEEE*

Abstract—The pool-hopping attack casts down the expected profits of both the mining pool and honest miners in Blockchain. The mainstream countermeasures, namely PPS (pay-per-share) and PPLNS (pay-per-last-N-share), can hedge pool hopping but need to charge miners some fees when they join in a pool. Obviously, the higher fee charged, the higher cost of joining the pool, the less motivation of a miner to mine in the pool. In this article, we apply the zero-determinant (ZD) theory to design a novel pooled mining which offers an incentive mechanism for motivating miners not to switch in pools strategically by economic means without fee charged. In short, the proposed pooled mining has three unique features: 1) *fee-free*. No fee is charged if the miner does not hop, 2) *wide applicability*. It can be employed in both prepaid and postpaid mechanisms, and 3) *fairness*. Even can dominate the game with any miner, a pool has to cooperate when a miner does not hop among pools, implying that the pool cannot squeeze the honest miners financially. The fairness of our scheme makes it have long-term sustainability. Both theoretical analyses and numerical simulations demonstrate the effectiveness of our scheme.

Index Terms—Pooled mining, pool-hopping attack, zero-determinant theory, incentive mechanism

1 Introduction

Blockchain is the underlying fabric of mainstream cryptocurrency systems such as Bitcoin [1] and Ethereum [2]. These cryptocurrencies have obtained a phenomenal success [3], [4], [5], recognized as the *wave of future* [6] with a total market capitalization around 179.6B dollars at present. To realize a distributed and trustable consensus, Blockchain is introduced as a public ledger, including a sequence of chained *blocks* with each recording a set of digital transactions. Since anyone can participate in creating and verifying blocks, Blockchain system is open, leading to its vulnerability.

To deter attacks incurred by the openness of Blockchain system which essentially originates from its decentralized nature, a Proof-of-Work (PoW) [1] mechanism is employed. PoW undoubtedly increases the cost of malicious behavior, making many security attacks such as Sybil attack financially unaffordable. This is because 1) mining is actually a race where only the winner who solves the PoW task first can verify digital transactions, which needs a sufficient amount of computational power; 2) solving cryptographic puzzles is a probabilistic process, implying that no one would win the race with certainty even though it is computationally powerful.

In return for mining blocks successfully, miners are rewarded in proportion to the computational powers they invested. However, due to significant computational resources needed and probabilistic factors involved in the mining process, a solo miner has low expected revenue as well as high volatility in the reward. For example, Bitcoin system now sets the difficulty of mining such that one block is generated every 10 minutes. Hence, a solo miner often has to wait 687 days in expectation to mine a block [7].

To tackle the above issue, solo miners join coalitions in the form of *mining pools*, gathering their computational powers to seek the solution of PoW puzzles and sharing the rewards proportionally to their contributions. This undoubtedly increases the chance of solving cryptographic puzzles successfully and makes the mining process more predictable. Hence, pooled mining can benefit miners from high payoffs and low variance in rewards. At present, nearly 80 percent of the computing power in Bitcoin and 60 percent of that in Ethereum belong to less than 8 and 3 mining pools, respectively.

The dominant position of pooled mining leads it to become a valuable target to be attacked. Many pools have an open trait, allowing any miner to join them through public Internet interfaces [8], which makes matters worse. Such a nature of openness makes pooled mining susceptible to attacks. There are mainly three kinds of security attacks in pooled mining: the selfish mining attack [9], [10], [11], the block withholding attack [8], [12], [13] and the pool-hopping

- *Hongwei Shi and Shengling Wang are with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China. E-mail: hongweishi@mail.bnu.edu.cn, wangshengling@bnu.edu.cn.*
- *Qin Hu is with the Department of Computer and Information Science, Indiana University - Purdue University Indianapolis, Indianapolis, IN 46202 USA. E-mail: qinhu@iu.edu.*
- *Xiuzhen Cheng is with the Department of Computer Science, George Washington University, Washington, DC 20052 USA. E-mail: cheng@gwu.edu.*
- *Junshan Zhang is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA. E-mail: junshan.zhang@asu.edu.*
- *Jiguo Yu is with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China , and also with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250014, China. E-mail: jiguoyu@sina.com.*

Manuscript received 9 Dec. 2019; revised 28 June 2020; accepted 24 Aug. 2020. Date of publication 4 Sept. 2020; date of current version 9 July 2021. (Corresponding author: Shengling Wang.) Digital Object Identifier no. 10.1109/TDSC.2020.3021686

attack [14]. The first two attacks can be well solved through the state-of-the-art approaches [10], [11], [12], [13], [15], and hence, we focus on the pool-hopping attack.

The pool-hopping attack was first proposed by Rosenfeld [14], in which the malicious miners strategically switch among the pools to obtain a higher payoff. This attack is cost-efficient and straightforward because of no more extra operations (e.g., keeping the block secret, dropping full proof of work or forking) needed. Studies have proved that a miner has no incentive to stay in a pool without pool hopping or redistributing the computing power [7], [12], [15]. As estimated in [14], a hopper can obtain a higher payoff up to nearly 28 percent, depending on the ratio between the hashrates of hoppers and honest miner in a pool. Pool hopping definitely casts down the mining power of a pool, resulting in its declined expected revenue. In addition, the pool-hopping attack also jeopardizes the interests of honest miners, who join in a pool continuously without switching to other pools. According to [14], honest miners in the attacked pool will receive 43 percent less payoff in the worst-case theoretically, which is unfair for them.

However, little research has studied the pool-hopping attack. PPS and PPLNS [14] are pioneer countermeasures. Considering that the unbalanced distribution of reward over time makes room for miners' strategic hopping, the key idea of PPS and PPLNS is reducing the variance of reward in time series. Typically, in a PPS pool, a miner will be rewarded as long as she submits a share (her contribution) to the pool, regardless of whether a block is mined successfully or not (we denote the pool as "he" and the miner as "she" for easy differentiation in this paper). PPLNS, one of the most prevailing reward mechanisms [16], drops the concept of "round", focusing on N shares submitted to the pool recently and distributing rewards according to the shares in proportion.

Essentially, the difference between PPS and PPLNS lies in that the former is driven by events while the latter is triggered by time. In detail, PPS rewards a miner once the event of receiving her share happens; PPLNS evaluates whether a miner should be awarded when the paying time arrives. The common feature of PPS and PPLNS is that they pay miners proportionally to their contribution, regardless of whether a block is mined successfully or not. Due to the uncertainty of mining results, the pool takes the full risk when no block is mined. Therefore, both PPS and PPLNS charge miners some fees when they join in a pool to alleviate such a risk, which also increases the hopping cost so as to further resist the pool-hopping attack. Obviously, the higher the fee, the higher the cost of the miner joining in the pool, and the smaller the motivation to mine and vice versa.

In a nutshell, the mainstream countermeasures to the pool-hopping attack, namely PPS and PPLNS, pose a risk to the pool as well as the cost to miners. Therefore, we propose a hopping-proof pooled mining with free fee in this paper, which can hedge pool hopping without any fee charged if the miner does not switch in pools strategically. The proposed pooled mining strategy has a wide scope of application since it can be employed in both prepaid and postpaid mechanisms. The former rewards once share is submitted, no matter whether there is a success mining or not; the latter awards only when the full cryptographic puzzle is solved.

It is challenging to realize the hopping-proof pooled mining without fee charged. The reasons behind the fact are: a) the strategic transferring among different pools is the instinctive demand of a miner. Especially when no fee is charged, costless hopping easily arouses miners to switch among pools; b) in the postpaid mode, mining risk is completely transferred from the pool to miners. In this situation, it is non-trivial to motivate miners to still work without hopping.

To tackle these challenges, we take advantage of the zero-determinant (ZD) theory to design an incentive mechanism for pooled mining, where cooperation (i.e., mining without hopping) is the dominant strategy of a rational miner in all situations. The ZD theory was first developed in [17] by Press and Dyson, in which the player who adopts the ZD strategy (i.e., the ZD adopter) can unilaterally set its adversary's utility no matter what strategy the adversary takes. The power of the ZD strategy endows the pool to dominate the game with any miner, rewarding her cooperation and punishing the defection, to lure the cooperation of the miner.

The main contributions of this paper can be summarized as follows:

- The interaction between the pool and any miner is formulated as an iterated prisoner's dilemma (IPD) game and the corresponding conditions are also identified. The generality of our model empowers the proposed pooled mining to have a wide scope of application, implying that it is suitable to both prepaid and postpaid mechanisms. When applied in a postpaid mechanism, the proposed pooled mining can incentivize miners to work without hopping while keeping the pool away from the risk of no block mined successfully.
- We investigate in detail whether the pool can be a ZD adopter and how he plays the ZD strategy. We draw a conclusion that the pool can unilaterally control the miner's payoff rather than his own one. The specific expected payoff of a miner that the pool can set is characterized.
- An incentive mechanism based on the ZD theory is proposed for motivating miners to work without hopping. Specifically, the proposed mechanism empowers the pool to encourage the miner to behave cooperatively by increasing her short-term payoff without any additional payment in the long run.
- Both theoretical analyses and simulations demonstrate the effectiveness of the proposed incentive mechanism. More importantly, we find the proposed pooled mining is fair, implying that even the pool can dominate the game with any miner, he has to cooperate when the miner works collaboratively. The fairness of our scheme makes it have long-term sustainability.

The rest of the paper is organized as follows. The related literatures are listed in Section 2. Section 3 describes the formulation of our problem. The ZD strategy for the pool in an iterated prisoner's game is deduced in Section 4. Based on which, we propose an incentive mechanism in light of the ZD theory in Section 5. We evaluate the mechanism both

theoretically in Section 6 and experimentally in Section 7. Section 8 concludes our paper finally.

2 Related Work

At present, the researchers mainly focus on three kinds of security attacks in pooled mining: the selfish mining attack, the block withholding (BWH) attack and the pool-hopping attack.

In detail, a selfish mining attacker [18] keeps its mined block secret and intentionally forks the main blockchain. Specifically, the selfish miner mines on its private branch instead of working on the public chain as the honest miners. When the public ledger approaches it's private chain, the selfish miner advertises its concealed chain to the public, leading to wasting resources of the honest miners on resolving cryptopuzzles which ends up gaining no rewards. Several defense mechanisms have been proposed to block this selfish manner as well as its variants. For example, Saad *et al.* [10] introduced the notion of truth state for every block and a novel parameter named expected transaction confirmation height in each transaction to detect selfish mining attack. Additionally, they proposed a new defensive algorithm to enforce miner's fair mining by continuously checking the length of transaction confirmation height and block publishing height. Without any additional modification, Zhang *et al.* [11] presented a backward-compatible defense mechanism fighting for selfish mining attack which neglects the untimely released blocks but concentrates more on the incorporate links to competing blocks of their predecessors.

The BWH attackers pretend to devote their computational capabilities into the target pool and then obtain payoffs. However, they send only partial proof of work, not full proof of work, resulting in reward reduction to other miners in the pool. This kind of attack was first proposed in [14], after which, Courtois *et al.* [19] summarized its concept and Eyal modeled the confrontation between the pools as a prisoner's dilemma in [8]. Specifically, in [8], a Nash equilibrium was established, where the rational pools would attack each other, resulting in a lose-lose situation. Besides, the pools are trapped into an iterative prisoner's dilemma, in which the pool chooses to attack or not is the so called miner's dilemma. Ongoing researches on avoiding this attack have proposed some efficient and cheap defense mechanisms. For example, Bag *et al.* [12] brought in two mechanisms to effectively defend it, the cryptocurrency commitment scheme and the hash function scheme. Both of the mechanisms are capable of protecting pools from misbehaved miners by distinguishing between a partial proof of work and a full proof of work. Considering that existing studies stress solely on the pure and equal strategy of conventional game, Hu *et al.* [13] analyzed the ZD strategy that may be utilized in block withholding attack between two pools. Based on which, different conditions for the pools playing the ZD strategy individually and simultaneously have been demonstrated comprehensively. In addition, a novel computational power splitting (CPS) game is established by Luu *et al.* [15], where they found the popular pool protocols are insecure when facing the block withholding attack. They initiated several public proposals to mitigate this attack and left more room for further investigation.

We focus on the pool-hopping attack in this work. The pool-hopping attack was first proposed by Rosenfeld [14], in which the malicious miners strategically switch among pools to obtain a higher payoff. In particular, pool-hopping denotes the malfeasance of malicious miners (or named by attackers, hoppers, adversaries) to mine in a pool only when the pool pays well and leave to join in other pools when it is not the case. As such, the hoppers can earn more from different pools than the honest ones who contribute their mining power consistently into one pool.

Technical speaking, to launch the pool-hopping attack successfully, the attacker needs to scrutinize the reward mechanism of each pool to figure out which pool pays well during which period of time for easy differentiation of hopping preferences. Since reward mechanisms are quite stable in each pool, such kind of examination can be derived straightforwardly and costlessly compared with other kinds of attacks, such as selfish mining attack, block withholding attack and forking attack, since no more extra operations (e.g., keeping the block secret, dropping full proof of work or forking) are needed.

Studies have proved that a miner has no incentive to stay in a pool without pool hopping or redistributing the computing power [7], [15]. The cooperation game constructed in [7] depicts that any arbitrary reward allocation mechanism creates an incentive for miners to give up the current pool and participate in others to increase the corresponding expected reward, verifying the effectiveness and inevitability of such a hopping behavior. Note that the hoppers may strategically jump among pools during various periods, they can also perform such kind of attack by cunningly splitting and redistributing their mining powers into different pools. In [15], Luu *et al.* analyzed the aforementioned malicious actions as a power splitting game, based on which, they derived the optimal strategies for such a player to maximize its utility. Consequently, this kind of greedy and opportunistic manner definitely casts down the mining power of a pool, resulting in its declined expected revenue.

Pioneer countermeasures for the pool-hopping attack are PPS, PPLNS and their variants, including the Slush's method, maximum pay-per-share (MPPS), and pay-once-PPLNS. Detailedly, the pool manager can calculate the score of each share based on the exponential score function $s = e^{\frac{T}{c}}$, in which s represents the score of the share given in time T and c denotes the scaling parameter. Due to the share's score, the pool hopping behavior can be alleviated in mining pools by reducing the score of shares at the earlier stage of the round while increasing the score of shares later on. Such kind of score-based method is recognized as the Slush's method and has been applied in the mining pools such as Slushpool [20]. Besides, in the maximum pay-per-share method, two balances are kept for each miner, that is, a PPS balance and a proportional balance [14]. To be specific, if the miner offers a share, her PPS share balance is increased as if the pool is a PPS pool. When the pool generates a block, the proportional balances of the miners are increased as if they have joined a proportional pool. Based on which, the reward paid for each miner is the minimum between the PPS balance and the proportional balance. In pay-once-PPLNS, every share is rewarded at most once [14]. In other words, the share is deleted after it is paid, leading a higher probability to the elder shares to be paid for

future blocks. If a share is partially paid, it will be deleted partially. However, theoretical analysis on the above mechanisms are lacking and their effectiveness in preventing pool-hopping attacks still remain an open issue [21].

3 Game Formulation

In this section, we introduce our game model to formulate the interaction between the pool and the miner. Generally, we define the strategy space of each player as a dichotomous space, namely cooperation (c) and defection (d). In the PoW mining scenario, the pool is considered as a cooperator if he decides to pay the highest payoff to the miner; otherwise, he is regarded as a defector. On the other hand, the miner can devote herself wholeheartedly to the current pool by providing her total computational power to the pool without hopping, defined as cooperation, or contribute herself halfheartedly through offering partial computing ability or switching to other pools strategically, denoted by defection. We denote the actions of the pool and the miner as $x, y \in \{c, d\}$, respectively. Therefore, there are four possibilities of states in each round between the pool and the miner, i.e., $XY = (cc, cd, dc, dd)$, where X and Y denote the state of the pool and that of the miner, respectively. It is worth to note that the terminal of a mining round mentioned in our model can be defined as the time a block is mined successfully or the paying time similar to that in PPLNS. Hence, the proposed scheme can be applied in both prepaid and postpaid mechanisms.

Each state will correspond to specific payoffs for both players, which can be derived as follows:

- if both the pool and the miner are collaborative with the pool providing the highest payoff and the miner offering her entire computing power to the current pool, the payoffs of them are represented as K_p and K_m, respectively;
- when the miner defects while the pool cooperates, the miner will get an increase of $\sigma > 0$ based on her original payoff K_m, while the pool may obtain a decrease of $\pi > 0$ on K_p;
- in the case that the defective pool plays against a cooperative miner, the payoff of the pool increases by $\mu > 0$, while the miner receives a loss of $\rho > 0$;
- when both players behave maliciously, the payoffs of the pool and the miner are $K_p - \pi + \mu$ and $K_m + \sigma - \rho$, respectively.

Subsequently, the payoff vectors of the pool, denoted as $\mathbf{S}_p = (S_p^{xy})$, and the miner, denoted as $\mathbf{S}_m = (S_m^{xy})$, $x, y \in \{c, d\}$, can be presented as follows:

$$\mathbf{S}_p = (S_p^{cc}, S_p^{cd}, S_p^{dc}, S_p^{dd}) = (K_p, K_p - \pi, K_p + \mu, K_p - \pi + \mu),$$

$$\mathbf{S}_m = (S_m^{cc}, S_m^{cd}, S_m^{dc}, S_m^{dd}) = (K_m, K_m + \sigma, K_m - \rho, K_m + \sigma - \rho),$$

which are also shown in Table 1.

Next, some insightful theorems are introduced to characterize the game in the following.

Theorem 3.1. *If $\pi > \mu, \rho > \sigma, \mu < \rho, \sigma < \pi$, a prisoner's dilemma (PD) game can be modeled to depict the confrontation between the pool and the miner.*

TABLE 1
Payoff Matrix of the Pool and the Miner

Pool \ Miner	Cooperation	Defection
Cooperation	K_p, K_m	$K_p - \pi, K_m + \sigma$
Defection	$K_p + \mu, K_m - \rho$	$K_p - \pi + \mu, K_m + \sigma - \rho$

Proof. To become a PD game, two fundamental conditions should be satisfied. In detail, 1) the stable state occurs when both players defect, i.e., $XY = dd$ is the Nash equilibrium; 2) mutual cooperation is the best outcome with respect to the social welfare, which means $XY = cc$ outperforms other states from an overall perspective.

The game between the pool and the miner satisfies the first condition. To be specific, if the miner is friendly, the pool will get a lower payoff as K_p when he cooperates than his payoff of $K_p + \mu$ when he defects; besides, if the pool challenges with a malicious miner, the payoff when he defects, i.e., $K_p - \pi + \mu$, is also larger than that of his cooperation, i.e., $K_p - \pi$. Thus, as a rational decision maker, the pool will always choose to defect rather than cooperation when facing an adversary with uncertain actions. With similar analysis, we can find the only feasible option for a rational miner is also to behave viciously. Accordingly, both the pool and the miner will select defection as the stable state. Therefore, the Nash equilibrium of this game comes to be $XY = dd$.

In order to investigate the second condition clearly, we denote the social welfare in each state as W_{cc}, W_{cd}, W_{dc} and W_{dd}. Thus, we have $W_{cc} = K_p + K_m$, $W_{cd} = K_p + K_m + \sigma - \pi$, $W_{dc} = K_p + K_m - \rho + \mu$, and $W_{dd} = K_p + K_m + \sigma + \mu - \rho - \pi$. Then the second condition is satisfied when $W_{cc} > W_{cd}, W_{cc} > W_{dc}, W_{cc} > W_{dd}$ hold. It is obvious that when $\pi > \mu, \rho > \sigma, \mu < \rho, \sigma < \pi$, the above inequalities can be satisfied. Based on the analyses above, as self-regarding players, the pool and the miner will choose malicious behavior to maximize their payoffs, leading to mutual defection as the stable state in the game consequently. However, the most favorable outcome of the confrontation turns out to be mutual cooperation. Therefore, a PD game is formed when $\pi > \mu, \rho > \sigma, \mu < \rho, \sigma < \pi$. □

Notably, the miner may stay in the current pool for a long time without hopping to others. Hence, in this case, the PD game mentioned above can become an iterated one if some conditions are satisfied, which are summarized in the following theorem.

Theorem 3.2. *If $\pi > \mu, \rho > \sigma, \mu < \rho, \sigma < \pi$, the confrontation between the pool and the miner can be modeled as an iterated prisoner's dilemma (IPD) game.*

Proof. A PD game becomes an iterated one when the payoff of any player's persistence on cooperation is larger than hopping between cooperation and defection. In other words, the inequalities below should hold

$$\begin{cases} 2K_p > K_p + \mu + K_p - \pi, \\ 2K_m > K_m + \sigma + K_m - \rho. \end{cases} \tag{1}$$

Hence, when $\pi > \mu, \rho > \sigma, \mu < \rho$ and $\sigma < \pi$, the game between the pool and the miner can be modeled as an IPD one. □

In light of the above analyses, we can find that the miner and the pool may be trapped into the iterated prisoner's dilemma, where the Nash equilibrium is far away from mutual cooperation, leading to low efficiency and distrust for Blockchain system in the long run. To tackle this problem, we employ the powerful ZD strategy to drive the players to cooperate so as to reach the win-win situation. As introduced in Section 1, the ZD adopter can unilaterally set its adversary's payoff no matter what strategy the adversary takes.

Aware of such an effective strategy, the pool is attracted to use the ZD strategy to resist a hopping miner. In this case, however, we are facing the following problems: *is the pool capable of being a ZD adopter? if yes, how does the ZD strategy work?* To address these questions, we conduct the following analyses.

4 ZD Strategy for the Pool

In this section, we examine whether the pool can play the ZD strategy, and if yes, how to achieve that. First, a Markov game is established between the pool and the miner. As mentioned in Section 3, there are four possible game results, i.e., $XY = (cc, cd, dc, dd)$, in each round. We define the pool's mixed strategy as $\mathbf{p} = (p_1, p_2, p_3, p_4)$, where p_1 represents the probability of choosing cooperation in this round based on the previous outcome cc. Similarly, when the previous outcome is cd, dc or dd, the probability of the pool to cooperate in this round is p_2, p_3 or p_4. Accordingly, the probability of the pool being defective in each round is $(1-p_1, 1-p_2, 1-p_3, 1-p_4)$ corresponding to different game results in last round. Comparably, in the cases that the miner chooses to cooperate when cc, cd, dc or dd happens previously, her strategy can be denoted as $\mathbf{q} = (q_1, q_2, q_3, q_4)$, while the probability of defecting is $(1-q_1, 1-q_2, 1-q_3, 1-q_4)$.

With the above-defined strategies of the pool and the miner, the Markov matrix in each round can be derived as follow:

$$\mathbf{A} = \begin{bmatrix} p_1q_1 & p_1(1-q_1) & (1-p_1)q_1 & (1-p_1)(1-q_1) \\ p_2q_2 & p_2(1-q_2) & (1-p_2)q_2 & (1-p_2)(1-q_2) \\ p_3q_3 & p_3(1-q_3) & (1-p_3)q_3 & (1-p_3)(1-q_3) \\ p_4q_4 & p_4(1-q_4) & (1-p_4)q_4 & (1-p_4)(1-q_4) \end{bmatrix},$$

where each element denotes the probability of state transition. For example, if the previous outcome is cc, combining the cooperation probabilities of the pool and the miner, i.e., p_1 and q_1, the probability of $XY = cc$ in this round is p_1q_1, so do other elements in $\mathbf{A}$.

Denote $\mathbf{v}$ as the stationary vector of matrix $\mathbf{A}$, then $\mathbf{v}^T\mathbf{A} = \mathbf{v}^T$ and $\mathbf{v}^T\mathbf{M} = \mathbf{0}$, where $\mathbf{M} = \mathbf{A} - \mathbf{I}$ ($\mathbf{I}$ is the identity matrix). According to the Cramer's rule, the equation $Adj(\mathbf{M})\mathbf{M} = det(\mathbf{M})\mathbf{I} = \mathbf{0}$ holds, where $Adj(\mathbf{M})$ and $det(\mathbf{M})$ represent the adjugate matrix and the determinant of $\mathbf{M}$. Subsequently, the equation above indicates that every row of $Adj(\mathbf{M})$ is in proportion to $\mathbf{v}$[17]. Thus, if the dot product of $\mathbf{v}$ with any vector $\mathbf{f} = (f_1, f_2, f_3, f_4)^T$ is conducted, the determinant remains unchanged with some elementary column transformation, such as adding the first column to the second and the third columns. Thus, we have:

$$\mathbf{v} \cdot \mathbf{f} = D(\mathbf{p}, \mathbf{q}, \mathbf{f}) = det \begin{bmatrix} p_1q_1 - 1 & p_1 - 1 & q_1 - 1 & f_1 \\ p_2q_2 & p_2 - 1 & q_2 & f_2 \\ p_3q_3 & p_3 & q_3 - 1 & f_3 \\ p_4q_4 & p_4 & q_4 & f_4 \end{bmatrix}.$$

It is evident that the second column of the above determinant is only related to the pool's strategy. Based on this, the expected payoffs of the pool (S_p) and the miner (S_m) can be derived as

$$S_p = \frac{\mathbf{v} \cdot \mathbf{S}_p}{\mathbf{v} \cdot \mathbf{1}} = \frac{D(\mathbf{p}, \mathbf{q}, \mathbf{S}_p)}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})},$$
$$S_m = \frac{\mathbf{v} \cdot \mathbf{S}_m}{\mathbf{v} \cdot \mathbf{1}} = \frac{D(\mathbf{p}, \mathbf{q}, \mathbf{S}_m)}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})}. \tag{2}$$

Hence, the linear relationship between the pool and the miner's expected payoffs holds as follows:

$$\alpha S_p + \beta S_m + \gamma = \frac{D(\mathbf{p}, \mathbf{q}, \alpha\mathbf{S}_p + \beta\mathbf{S}_m + \gamma\mathbf{1})}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})}, \tag{3}$$

where $\alpha,\ \beta,\ \gamma$ are coefficients.

Therefore, if the pool sets his strategy the same as $\alpha\mathbf{S}_p + \beta\mathbf{S}_m + \gamma\mathbf{1}$, the determinant in the numerator equals 0, because there exists two identical columns. In this case, $\alpha S_p + \beta S_m + \gamma = 0$, implying that a linear relation is established between the expected payoffs S_p and S_m, where the corresponding strategy is therefore called *Zero-Determinant Strategy*, denoted as $\hat{\mathbf{p}}$ below.

Specifically, when the pool sets $\hat{\mathbf{p}} = \beta\mathbf{S}_m + \gamma\mathbf{1}$ (i.e., $\alpha = 0$), the pool can control the miner's expected payoff independently as $S_m = -\frac{\gamma}{\beta}$; while when he exerts his strategy as $\hat{\mathbf{p}} = \alpha\mathbf{S}_p + \gamma\mathbf{1}$ by setting $\beta = 0$, he can set his own expected payoff at $S_p = -\frac{\gamma}{\alpha}$. The following theorem demonstrates the effectiveness of the ZD strategy adopted by the pool.

Theorem 4.1. *The pool can unilaterally control the miner's expected payoff as $S_m = \frac{(1-p_1)S_m^{dd} + p_4S_m^{cc}}{1-p_1+p_4}$, while he is not able to set his own expected payoff independently.*

Proof. First, if the pool wants to control his adversary's expected payoff as $S_m = -\frac{\gamma}{\beta}$ by setting $\alpha = 0$, the specific ZD strategy of the pool should satisfy $\hat{\mathbf{p}} = \beta\mathbf{S_m} + \gamma\mathbf{1}$, according to which, we can deduce p_2 and p_3 with respect to p_1 and p_4

$$\begin{cases} p_2 = \frac{p_1(S_m^{cd} - S_m^{dd}) - (1+p_4)(S_m^{cd} - S_m^{cc})}{S_m^{cc} - S_m^{dd}}, \\ p_3 = \frac{(1-p_1)(S_m^{dd} - S_m^{dc}) + p_4(S_m^{cc} - S_m^{dc})}{S_m^{cc} - S_m^{dd}}. \end{cases} \tag{4}$$

It is evident that p_2 and p_3 are meaningful as they belong to [0, 1]. Therefore, it is clear that being a ZD player, the pool can set the miner's expected payoff unilaterally. And the miner's expected payoff comes to be

$$S_m = -\frac{\gamma}{\beta} = \frac{(1-p_1)S_m^{dd} + p_4S_m^{cc}}{1-p_1+p_4}. \tag{5}$$

As (5) consisting of a weighted average of S_m^{cc} and S_m^{dd} with weights p_4 and $1-p_1$, we can conclude that the

expected payoff of the miner can be set in the range of $[S_m^{dd}, S_m^{cc}]$ by the pool's ZD strategy.

Second, when it comes to the case that the pool sets his own expected payoff, the ZD adopter's strategy should meet $\hat{\mathbf{p}} = \alpha \mathbf{S_p} + \gamma \mathbf{1}$ ($\beta = 0$). Using p_1 and p_4 to represent α and γ, we have

$$\begin{cases} \alpha & = \frac{p_1 - p_4 - 1}{S_p^{cc} - S_p^{dd}}, \\ \gamma & = \frac{(1-p_1)S_p^{dd} + p_4 S_p^{cc}}{S_p^{cc} - S_p^{dd}}. \end{cases} \tag{6}$$

And we can use p_1 and p_4 to describe p_2 and p_3 as

$$\begin{cases} p_2 & = \frac{(1+p_4)(S_p^{cc} - S_p^{cd}) - p_1(S_p^{dd} - S_p^{cd})}{S_p^{cc} - S_p^{dd}}, \\ p_3 & = \frac{-(1-p_1)(S_p^{dc} - S_p^{dd}) - p_4(S_p^{dc} - S_p^{cc})}{S_p^{cc} - S_p^{dd}}, \end{cases} \tag{7}$$

which indicates $p_2 \geq 1$ and $p_3 \leq 0$. Under this condition, the pool's strategy is feasible in only one case, i.e., $\hat{\mathbf{p}} = (1, 1, 0, 0)$, resulting in $\alpha = 0$ and $\gamma = 0$ according to (6). Thus, as a ZD player, the pool cannot control his payoff.□

5 Incentive Mechanism Based on the ZD Strategy

In this section, we propose a ZD-based incentive mechanism for the pooled mining to hinder pool-hopping attacks. Theorem 4.1 reveals the capability of the pool as a ZD player to set the miner's expected payoff unilaterally. However, whether the pool can take advantage of such a capability to regulate the miner depends on her strategy. If the miner's strategy is irrelevant to her payoff, such as all-cooperation (ALLC, $\mathbf{q} = (1, 1, 1, 1)$), all-defection (ALLD, $\mathbf{q} = (0, 0, 0, 0)$), tit-for-tat (TFT, $\mathbf{q} = (1, 1, 0, 0)$), the pool cannot employ the ZD strategy to motivate the cooperative behavior of the miner. Hence, the proposed ZD-based incentive mechanism is suitable for the case that the strategy is laid down by the miner in light of her payoff. Win-stay-lose-shift (WSLS, $\mathbf{q} = (1, 0, 0, 1)$) and evolutionary strategies are typical payoff-driven examples.

A WSLS player will keep the same strategy as the previous round in which the outcome is good, that is so called "win-stay". Otherwise, it will adopt the strategy opposite to the one in the previous round, which is therefore named as "lose-shift". Hence, WSLS can be regarded as a particular case of the evolutionary strategy. In this work, we take the evolutionary strategy as the representative for further analysis, which can be categorized into two kinds: *non-memorial* and *memorial*. We introduce them in detail as follows.

5.1 Evolutionary Strategies

The non-memorial evolutionary (**E**) strategy is featured by the fact that an **E** player may develop the strategy only based on its expected payoff. Specifically, as a rational player, if the cooperative behavior brings about a higher payoff than the defective one, the **E** player will choose to collaborate and vice versa. A typical non-memorial evolutionary strategy can be formulated as follow [22]:

$$q^t(c|\mathbf{p}) = \frac{e^{\epsilon[E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p})]}}{1 + e^{\epsilon[E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p})]}}, \tag{8}$$

where $q^t(c|\mathbf{p})$ denotes the non-memorial **E** player's cooperation probability in round t based on the pool's strategy $\mathbf{p}$ and $\epsilon > 0$ is a scaling parameter. Besides, $E_m^t(c|\mathbf{p})$ and $E_m^t(d|\mathbf{p})$ represent the expected payoffs of the miner who acts cooperatively and defectively.

Different from the non-memorial evolutionary strategy, the memorial evolutionary strategy is associated with not only the expected payoff but also its strategy in the previous round, which we call it *memory*. That is to say, informed of the previous strategy and the expected payoff, the memorial **E** player may adjust its strategy more rationally.

Inspired by [23], we present the memorial evolutionary strategy as following: if the cooperation probabilities of the pool and the miner are denoted as p^t and q^t in round t, then the miner's cooperation probability q^{t+1} in the next round evolves as

$$q^{t+1} = q^t \cdot \frac{W_c^t}{E_m^t}, \tag{9}$$

where W_c^t indicates the expected payoff of the miner when she cooperates and E_m^t implies the expected payoff of the miner in round t. Accordingly, W_c^t and E_m^t can be calculated by

$$\begin{aligned} W_c^t &= p^t \cdot S_m^{cc} + (1 - p^t) \cdot S_m^{dc}, \\ E_m^t &= q^t \cdot W_c^t + (1 - q^t) \cdot W_d^t, \end{aligned} \tag{10}$$

where $W_d^t = p^t \cdot S_m^{cd} + (1 - p^t) \cdot S_m^{dd}$ is the miner's expected payoff when she defects.

5.2 ZD Incentive Mechanism

From Equations (8) and (9), it is clear that if the miner obtains more payoff as a cooperative player, her cooperation probability will increase. That is to say, the miner is more likely to devote her computing power entirely to the pool without hopping if such an action brings about a higher payoff. Therefore, as a ZD player, the pool may reward the cooperation of a miner with a higher payoff while punishing her defection with the lower one. Based on this, we propose a ZD-based incentive mechanism for the pool to coerce the miner's collaborative action, thereby deterring the hopping behavior of the miner, which is detailed in the following.

As shown in Algorithm 1, in the first round, we offer the reward to each miner i ($i = 1, 2, \ldots, N$) proportionally to her contribution to the pool. The historical best computing power B_i is recorded as the initial computation power of each miner i, namely m_i^1 (Lines 1-4). In practice, whether a miner behaves cooperatively or defectively can not be deduced without any side information, since it is the private information of the miner. Hence, the pool has to differentiate a collaborate or defective miner based on the observation of the difference of computational powers between two continuous rounds. This requires the pool to record the computation power m_i^j of any miner i at the end of each round j (Line 7), so that the pool can obtain the difference of the devoted computational power of miner i between round $j - 1$ and

round j, i.e., $\Delta m_i^j = m_i^j - m_i^{j-1}$ (Line 8). If $\Delta m_i^j \geq 0$, miner i is considered to be a cooperative player and vice versa.

The case $\Delta m_i^j < 0$ indicates that the miner splits her computing power into other pools, thus implying she is a pool-hopping attacker (the situation where the miner is unavailable due to some reasons such as lacking of electricity is out of our consideration). Her payoff is therefore needed to be reduced in order to hinder such an attack. Under this situation, the pool will exert the ZD strategy, setting the attacker's payoff as the minimum one, i.e., L (Lines 9-10). If $\Delta m_i^j = 0$, the pool provides the same payoff to the miner as that in the last round (Lines 11-12). When $\Delta m_i^j > 0$, the pool would update B_i if needed (Lines 14-16). Since this case indicates the miner behaves more cooperatively, the pool will increase her payoff as $E_i^j = H * \frac{e^{\zeta \cdot y}}{1+e^{\zeta \cdot y}}$, where $y = (\frac{\Delta m_i^j}{B_i} + 1) \cdot E_i^{j-1}$ and $\zeta > 0$ represents a scaling parameter (Line 17-18). It is worth to note that the more increment of computational power relative to B_i is, the higher reward the miner can obtain, which is up to the maximum payoff that the pool can offer, namely H.

Algorithm 1. The ZD-Based Incentive Mechanism

Require:
The total number of iterations, M;
The number of miners, N;
The initial computation power of miner i, m_i^1;
The minimum and maximum payoffs that the pool can offer, L and H;

1: **for** $i = 1$ to N **do**
2: Calculate the initial reward according to $\frac{m_i^1}{\sum_{i=1}^{N} m_i^1} \cdot [H - L] + L$
3: $B_i = m_i^1$
4: **end for**
5: **for** $i = 1$ to N **do**
6: **for** $j = 2$ to M **do**
7: Update computation power m_i^j
8: $\Delta m_i^j = m_i^j - m_i^{j-1}$
9: **if** $\Delta m_i^j < 0$ **then**
10: Calculate $\mathbf{p}^j$ which makes $E_i^j = L$
11: **else if** $\Delta m_i^j = 0$ **then**
12: $\mathbf{p}^j = \mathbf{p}^{j-1}$ which makes $E_i^j = E_i^{j-1}$
13: **else if** $\Delta m_i^j > 0$ **then**
14: **if** $B_i < m_i^j$ **then**
15: $B_i = m_i^j$
16: **end if**
17: $y = (\frac{\Delta m_i^j}{B_i} + 1) \cdot E_i^{j-1}$
18: Calculate $\mathbf{p}^j$ which makes $E_i^j = H * \frac{e^{\zeta \cdot y}}{1+e^{\zeta \cdot y}}$
19: **end if**
20: **end for**
21: **end for**

6 Theoretical Analysis

In this section, we analyze the proposed incentive mechanism theoretically.

Theorem 6.1. *For any non-memorial evolutionary miner who is motivated by the ZD incentive mechanism, it is conceivable that the miner's cooperation probability will be maximized.*

Proof. To maximize $q^t(c|\mathbf{p})$ according to (8), we turn to prove that $E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p})$ rises with the increase of game round t if the miner is a cooperative one. According to Algorithm 1, if any miner i behaves more cooperatively than the previous round, we have

$$\begin{aligned} E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p}) &= H * \frac{e^{\zeta \cdot y}}{1+e^{\zeta \cdot y}} - L \\ &= H * \frac{e^{\zeta \cdot (\frac{\Delta m_i^t}{B_i}+1) \cdot R_i^{t-1}}}{1+e^{\zeta \cdot (\frac{\Delta m_i^t}{B_i}+1) \cdot R_i^{t-1}}} - L. \end{aligned} \tag{11}$$

Since $(\frac{\Delta m_i^t}{B_i} + 1) \cdot R_i^{t-1}$ keeps raising because of the miner's collaborative behavior, $\frac{e^{\zeta \cdot (\frac{\Delta m_i^t}{B_i}+1) \cdot R_i^{t-1}}}{1+e^{\zeta \cdot (\frac{\Delta m_i^t}{B_i}+1) \cdot R_i^{t-1}}}$ becomes to one at last, leading $E_m^t(c|\mathbf{p}) - E_m^t(d|\mathbf{p})$ equals to $H - L$ consequently. Hence, driven by the proposed ZD incentive mechanism, $q^t(c|\mathbf{p})$ can evolve to the maximum. □

Theorem 6.2. *For any memorial evolutionary miner who is motivated by the ZD incentive mechanism, her cooperation probability tends to 1 gradually.*

Proof. In light of (9), a memorial evolutionary miner can calculate her cooperation probability according to W_c^t and E_m^t, which can be deduced by (10). In practice, we use the cooperative frequencies f_p^t and f_m^t to approximate p^t and q^t. Specifically, f_p^t indicates the number of rounds the pool cooperates divided by the total number of rounds, while f_m^t denotes that of a miner.

Based on the ZD incentive mechanism, we consider the following two cases, where the miner chooses to cooperate or defect [24].

a) if the miner is considered as cooperative, the pool may reward her, resulting in $E_m^{t+1} \geq E_m^t$. In this case, with the increase of E_m^{t+1} and f_m^{t+1}, W_c^{t+1} turns to

$$W_c^{t+1} = \frac{E_m^{t+1} - (1 - f_m^{t+1}) W_d^{t+1}}{f_m^{t+1}}. \tag{12}$$

Hence, $\lim_{t \to +\infty} W_c^{t+1} = \frac{E_m^{t+1} - (1-f_m^t) W_d^{t+1}}{f_m^t} > W_c^t$ because of $W_d^{t+1} = W_d^t$.

b) when the miner is regarded as a defective miner, then we have $E_m^{t+1} \leq E_m^t$, and the decrease of E_m^{t+1} and f_m^{t+1} will lead to

$$W_d^{t+1} = \frac{E_m^{t+1} - f_m^{t+1} W_c^{t+1}}{1 - f_m^{t+1}}. \tag{13}$$

Comparably, $\lim_{t \to +\infty} W_d^{t+1} = \frac{E_m^{t+1} - f_m^t W_c^{t+1}}{1-f_m^t} < W_d^t$ because of $W_c^{t+1} = W_c^t$.

To sum up, Case a) indicates that W_c^t increases and W_d^t remains unchanged and Case b) implies that W_d^t declines while W_c^t remains steady. Thus, $\exists T^* \in \mathbb{Z}^+$, such that $\forall t > T^*$, $W_c^t > W_d^t$ holds. Based on this, E_m^t can be derived as

$$\begin{aligned} E_m^t &= f_m^t W_c^t + (1 - f_m^t) W_d^t \\ &< f_m^t W_c^t + (1 - f_m^t) W_c^t = W_c^t. \end{aligned} \tag{14}$$

In light of (14), we can conclude that $q^{t+1} = q^t \frac{W_c^t}{E_m^t} \to 1$ with the increase of game round t. That is to say, the

memorial evolutionary miner will gradually increase the cooperation probability to one eventually. □

Conclusively, the non-memorial and memorial evolutionary miner will be encouraged to behave cooperatively by the proposed ZD incentive mechanism in the end.

Another essential nature of the proposed incentive mechanism is that it can be employed into the prepaid mechanism as well as the postpaid mechanism, with the former rewards the miner when a share is submitted and the latter defines the terminal of a mining round as the time a block is mined successfully. Noteworthily, the ZD incentive mechanism is free-fee charged for miners in both prepaid and postpaid cases due to their wholehearted devotions. More importantly, in the postpaid mechanism, the proposed incentive mechanism can hinder pool hopping attackers without putting any risk on the pools since our mechanism enables the miners to mine wholeheartedly until a block is generated successfully.

Now that such a powerful strategy the pool can employ, he has an overwhelmingly dominant position compared with the miner, then *is the pool capable of getting a higher payoff greedily through defecting when the miner collaborates?* We use the following theorem as a response to the above concern.

Theorem 6.3. *When the miner chooses to cooperate, the only rational strategy of the pool who employs the ZD incentive mechanism is to collaborate.*

Proof. As demonstrated in Theorems 6.1 and 6.2, the miner will choose to contribute her maximum computational power into the pool because of the effectiveness of the proposed ZD incentive mechanism. In this case, the pool will provide the miner with the maximal payoff. Therefore, we will discuss what the ZD strategy is when the pool sets the expected payoff of the miner as the optimal value in the following.

According to Section 4, the miner's expected payoff can be set as $S_m = \frac{(1-p_1)S_m^{dd}+p_4 S_m^{cc}}{1-p_1+p_4}$, which belongs to $[S_m^{dd}, S_m^{cc}]$. Due to

$$\begin{aligned}\frac{\partial S_m}{\partial p_1} &= \frac{p_4(\rho-\sigma)}{(1-p_1+p_4)^2},\\ \frac{\partial S_m}{\partial p_4} &= \frac{(1-p_1)(\rho-\sigma)}{(1-p_1+p_4)^2},\end{aligned} \tag{15}$$

$\frac{\partial S_m}{\partial p_1} > 0$ and $\frac{\partial S_m}{\partial p_4} > 0$ because of $\rho > \sigma$ as indicated in Theorem 3.2, implying a monotonically increasing relationship between S_m and p_1, p_4. Hence, when $p_1 = 1$, $p_4 = 1$, the pool can maximize the miner's expected payoff. Furthermore, according to (4), if p_1 and p_4 are equivalent to 1, the only possible value of p_2 is 1 because p_2 should lie in [0,1] to be a probability, so as for p_3. That is to say, the pool can set $\mathbf{p} = (1, 1, 1, 1)$ to maximize the payoff of a miner.

In light of the above analysis, once the miner cooperates, the pool will set his ZD strategy as $\mathbf{p} = (1, 1, 1, 1)$ to maximize a collaborative miner's expected payoff. That is to say, whenever the miner cooperates, the pool will collaborate subsequently. □

In summary, the pool will be collaborative in return if the miner offers her maximum computing power. Thus, the proposed ZD incentive mechanism is fair to both sides, which makes it be long-term sustainable. Such an aim is achieved via controlling the miner's short-term expected payoff by the pool. Then, *what are the players' actual payoffs over the long run?* This question can be answered by the following two theorems.

Theorem 6.4. *In the long run, the miner's actual payoff equals to K_m based on our proposed ZD incentive mechanism.*

Proof. a) For a non-memorial evolutionary miner, $\exists \tau \in \mathbb{Z}^+$, such that $\forall t \geq \tau$, q^t can be maximized. That is to say, when $t \geq \tau$, the expected payoff of the miner is identical to K_m, which is the maximum payoff for a cooperative miner. In light of this, the actual payoff of the miner P_m^A can be derived as the average of the expected payoff E_m^i in each round i, where $i < \tau$ and the expected payoff K_m after round τ. Therefore, P_m^A can be written as

$$P_m^A = \lim_{t\to\infty} \frac{\Sigma_{i=1}^{\tau-1} E_m^i + \Sigma_{i=\tau}^{t} K_m}{t} = K_m. \tag{16}$$

b) The actual payoff of a memorial evolutionary miner is

$$\begin{aligned}P_m^A &= \lim_{t\to\infty} \frac{W_c^t - (1-p^t)(K_m+\sigma-\rho)}{p^t} = \lim_{t\to\infty} W_c^t\\ &= \lim_{t\to\infty} \frac{E_m^t - (1-q^t)W_d^t}{q^t} = \lim_{t\to\infty} E_m^t = K_m.\end{aligned} \tag{17}$$

□

By inspecting Theorem 6.4, the miner will receive the actual payoff P_m^A as K_m over the long run. Then, *is it possible for the pool to own more payoff by greedy behavior?* This question can be resolved by the following theorem.

Theorem 6.5. *In the long run, the pool's actual payoff P_p^A is equivalent to K_p based on our proposed ZD incentive mechanism.*

Proof. According to Theorem 6.3, the pool will behave cooperatively to reward a collaborative miner, implying that $XY = cc$ is the stable state for the game. In such a case, $P_p^A = K_p$ holds according to Table 1. □

In light of Theorems 6.4 and 6.5, the pool and miner will obtain the actual payoffs as K_m and K_p, respectively. That is to say, neither the pool nor the miner can receive higher reward by noncooperative manner over the long run, which is quite fair for both sides.

7 Performance Evaluation

To testify the effectiveness of the ZD incentive mechanism proposed in Section 5, we conduct numerical simulations in this section. To be specific, we set the payoff vectors of the pool and the miner as $\mathbf{S}_p = (3, 0, 5, 2)$ and $\mathbf{S}_m = (3, 5, 0, 2)$, which is a typical example of the prisoner's dilemma. We also carry out the simulations with other parameter settings and derive the comparable results. So we omit to present those results to avoid redundancy. Note that each simulation is repeated 100 times to get the average value for statistical confidence.

In detail, if the pool is a ZD adopter competing with a miner who employs four classical strategies, i.e., ALLC,

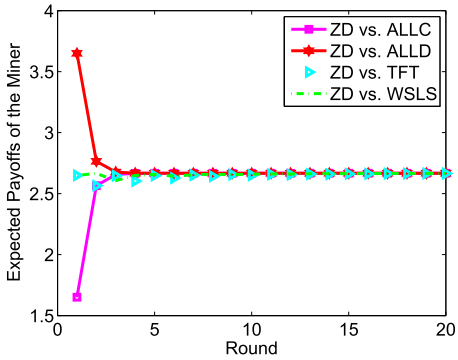
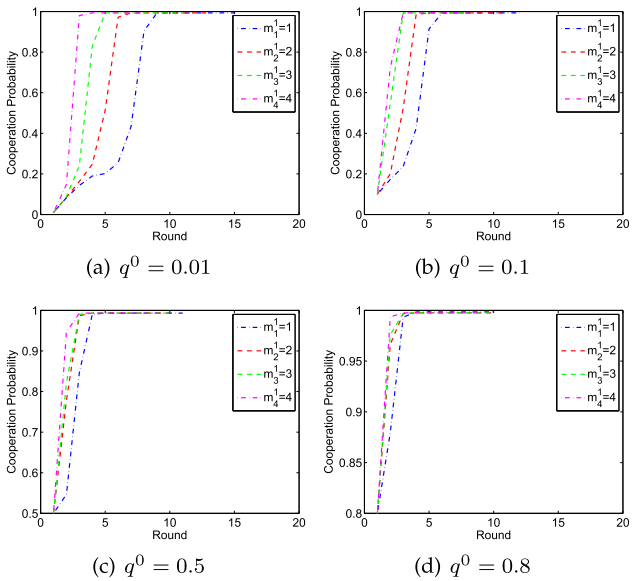


Fig. 1. The expected payoffs of the miner when she adopts ALLC, ALLD, TFT, WSLS strategies, and the pool employs the ZD strategy.

ALLD, TFT and WSLS, the miner's expected payoffs can be set at a fixed value as shown in Fig. 1. Taking the specific ZD strategy of the pool $\mathbf{p} = (0.9, 0.3, 0.8, 0.2)$ as an example, no matter what strategies the miner employs, her expected payoff will finally become to a constant. That is to say, the adversary's outcome can be controlled unilaterally by the ZD adopter because of his effective strategy.

As mentioned in Section 5, the classical strategies ALLC, ALLD and TFT are out of our consideration because the strategies are irrelevant to the payoff of the player. Moreover, WSLS is regarded as a special evolutionary strategy. Hence, only the simulations of the evolutionary miners who compete with a ZD pool are included in this work, which are demonstrated as follows.

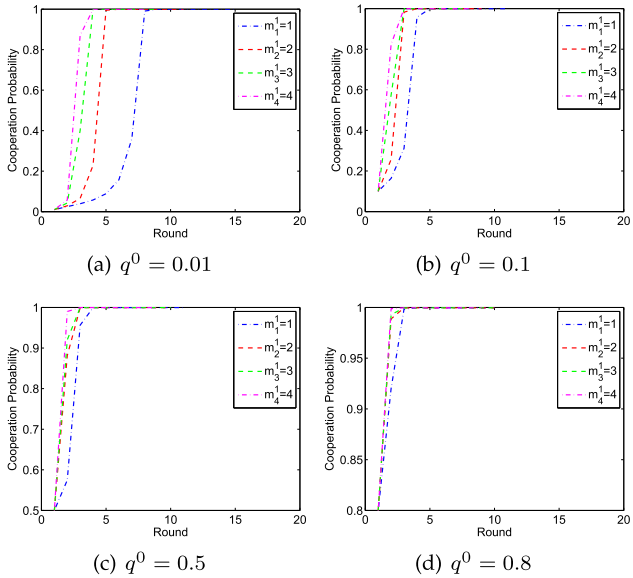
In our simulation, we assume there are four miners in a pool, whose initial computational powers are respectively $m_1^1 = 1, m_2^1 = 2, m_3^1 = 3, m_4^1 = 4$. It is worthy noting that the cases in which more miners exist in a ZD pool share the same conclusion, so we omit it for reducing repetition. Setting the original cooperation probabilities (CPs) $q^0 = 0.01, q^0 = 0.1, q^0 = 0.5$ and $q^0 = 0.8$, Figs. 2 and 3 respectively show how the CPs of the non-memorial evolutionary miners evolve according to the proposed ZD incentive mechanism when $\epsilon = 5$ and 8. In particular, ϵ is set to be big



(a) $q^0 = 0.01$ (b) $q^0 = 0.1$

(c) $q^0 = 0.5$ (d) $q^0 = 0.8$

Fig. 2. The evolutions of the CPs of the non-memorial evolutionary miners when $\epsilon = 5$.



(a) $q^0 = 0.01$ (b) $q^0 = 0.1$

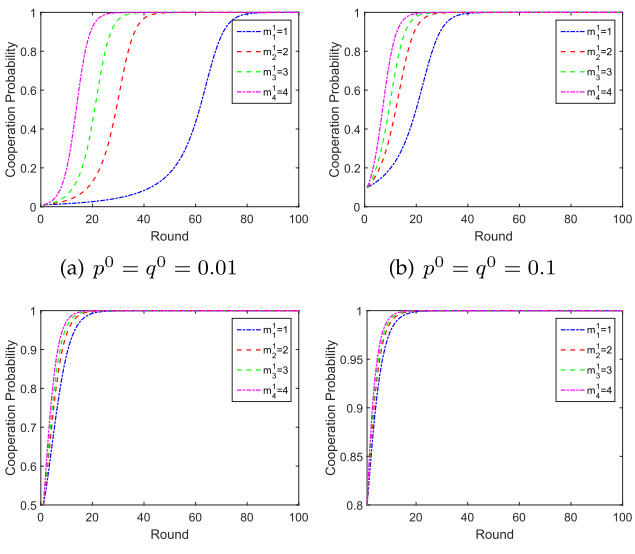
(c) $q^0 = 0.5$ (d) $q^0 = 0.8$

Fig. 3. The evolutions of the CPs of the non-memorial evolutionary miners when $\epsilon = 8$.

enough here so that the maximum cooperation probability of a non-memorial evolutionary player (calculated by (8)), can approach to 1.

Through further observation of Figs. 2 and 3, we can conclude that the CPs of the non-memorial evolutionary miners converge to one with different speeds, which is mainly because of different initial computational investments and the scaling parameter ϵ. To be specific, a miner with a larger initial computing investment would be more inclined to accelerate the cooperation process due to the higher growth of payoff. Intuitively, a higher ϵ brings about a faster convergence speed of the CP according to (8).

Fig. 4 plots the CPs of a memorial evolutionary miner driven by the ZD-based incentive mechanism, where the CPs go up to 1 gradually with the initial values as $p^0 = q^0 = 0.01, 0.1, 0.5$, and 0.8. In detail, each subfigure shows that the CP of the miner with a small initial input con-



(a) $p^0 = q^0 = 0.01$ (b) $p^0 = q^0 = 0.1$

(c) $p^0 = q^0 = 0.5$ (d) $p^0 = q^0 = 0.8$

Fig. 4. The evolutions of the CPs of the memorial evolutionary miners.

verges slowly compared with other miners, even though they share the same initial cooperation probability. The reason may lie in that the miner with a smaller initial computing investment may get a relatively lower payoff in the beginning, leading a slow growth of the expected payoff. Thus, her CP would rise slower comparably. Moreover, considering the CPs of a miner with the same initial investment but having different initial cooperation probabilities, for example, the blue lines in subfigures (a)-(d), the result is that the higher the initial CP is, the faster it is converged to one, which is mainly caused by the *memory* we mentioned above in light of (9).

8 Conclusion

In this paper, we propose a fee-free pooled mining fighting for the pool-hopping attack in Blockchain. To that aim, we formulate the interaction between the pool and any miner as an IPD game and identify the corresponding conditions. Based on the model, we take advantage of the ZD theory to empower the pool to unilaterally control the miners payoff, which can be used to motivate the cooperation of miners through the proposed ZD incentive mechanism. Our work is featured by three traits, which are *fee-free*, *wide applicability* and *fairness*. Both theoretical analyses and numerical simulations demonstrate the effectiveness of the ZD incentive mechanism.

Acknowledgments

This work was supported by National Key R&D Program of China (No. 2019YFB2102600), National Natural Science Foundation of China (No. 61772080, 61672321, 61771289, 61832012, and 62072044), the Blockchain Core Technology Strategic Research Program of Ministry of Education of China (No. 2020KJ010301), BNU Interdisciplinary Research Foundation for the First-Year Doctoral Candidates (No. BNUXKJC2022), the International Joint Research Project of Faculty of Education, Beijing Normal University, and Engineering Research Center of Intelligent Technology and Educational Application, Ministry of Education.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008.
[2] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.
[3] H. Zhou, Y. Niu, J. Liu, C. Zhang, L. Wei, and Y. Fang, "A privacy-preserving networked hospitality service with the bitcoin blockchain," in *Proc. Int. Conf. Wireless Algorithms Syst. Appl.*, 2018, pp. 696–708.
[4] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat, and Z. Zhao, "A blockchain based witness model for trustworthy cloud service level agreement enforcement," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1567–1575.
[5] N. Papadis, S. Borst, A. Walid, M. Grissa, and L. Tassiulas, "Stochastic models and wide-area network measurements for blockchain design and analysis," in *Proc. Conf. Comput. Commun.*, 2018, pp. 2546–2554.
[6] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2016, pp. 305–320.
[7] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2015, pp. 919–927.
[8] I. Eyal, "The miner's dilemma," in *Proc. IEEE Symp. Secur. Privacy*, 2015, pp. 89–103.
[9] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2016, pp. 515–532.
[10] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in *Proc. Int. Conf. Comput. Netw. Commun.*, 2019, pp. 360–364.
[11] R. Zhang and B. Preneel, "Publish or perish: A backward-compatible defense against selfish mining in bitcoin," in *Proc. Cryptographers Track RSA Conf.*, 2017, pp. 277–292.
[12] S. Bag, S. Ruj, and K. Sakurai, "Bitcoin block withholding attack: Analysis and mitigation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1967–1978, Aug. 2017.
[13] Q. Hu, S. Wang, and X. Cheng, "A game theoretic analysis on block withholding attacks using the zero-determinant strategy," in *Proc. IEEE/ACM 27th Int. Symp. Qual. Service*, 2019, pp. 1–10.
[14] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," Tech. Rep., 2011.
[15] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, "On power splitting games in distributed computation: The case of bitcoin pooled mining," in *Proc. IEEE 28th Comput. Secur. Found. Symp.*, 2015, pp. 397–411.
[16] P. Chatzigiannis, F. Baldimtsi, I. Griva, and J. Li, "Diversification across mining pools: Optimal mining strategies under pow," *Workshop Econ. Inf. Secur.*, pp. 966–974, 2019.
[17] W. H. Press and F. J. Dyson, "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent," *Proc. Nat. Acad. Sci. United States America*, vol. 109, no. 26, pp. 10 409–10 413, 2012.
[18] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Commun. ACM*, vol. 61, no. 7, pp. 95–102, 2018.
[19] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," *Comput. Res. Repository*, Tech. Rep., 2014.
[20] Slushpool. Accessed: Sep. 9, 2020. [Online]. Available: http://slushpool.com/home/
[21] W. Wang *et al.*, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.
[22] H. P. Young, "The diffusion of innovations in social networks," *Economy Evolving Complex Syst. III: Current Perspectives Future Directions*, vol. 267, 2006, Art. no. 39.
[23] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary game for mining pool selection in blockchain networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 760–763, Oct. 2018.
[24] Q. Hu, S. Wang, L. Ma, R. Bie, and X. Cheng, "Anti-malicious crowdsourcing using the zero-determinant strategy," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 1137–1146.

Hongwei Shi received the BS degree in computer science from Beijing Normal University, Beijing, China, in 2018. She is currently working toward the MS degree in computer science at Beijing Normal University, Beijing, China. Her research interests include blockchain, game theory, and combinatorial optimization.

Shengling Wang (Senior Member, IEEE) received the PhD degree from Xi'an Jiaotong University, Xi'an, China, in 2008. She is a full professor with the School of Artificial Intelligence, Beijing Normal University. After that, she did her postdoctoral research with the Department of Computer Science and Technology, Tsinghua University. Then, she worked as an assistant and associate professor from 2010 to 2013 with the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include mobile/wireless networks, game theory, and crowdsourcing.

Qin Hu received the PhD degree in computer science from the George Washington University, Washington, DC, in 2019. She is currently an assistant professor with the Department of Computer and Information Science, Indiana University - Purdue University Indianapolis. Her research interests include wireless and mobile security, crowdsourcing/crowdsensing, and blockchain.

Xiuzhen Cheng (Fellow, IEEE) received the MS and PhD degrees in computer science from the University of Minnesota, Twin Cities, Minnesota, in 2000 and 2002, respectively. She is a professor with the Department of Computer Science, George Washington University, Washington, DC. Her current research interests focus on privacy-aware computing, wireless and mobile security, dynamic spectrum access, mobile handset networking systems (mobile health and safety), cognitive radio networks, and algorithm design and analysis. She has served on the editorial boards of several technical publications and the Technical Program Committees of various professional conferences/workshops. She has also chaired several international conferences. She worked as a program director for the U.S. National Science Foundation (NSF) from April to October 2006 (full time), and from April 2008 to May 2010 (part time). She published more than 170 peer-reviewed papers.

Junshan Zhang (Fellow, IEEE) received the PhD degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, in 2000. He joined the School of Electrical, Computer, and Energy Engineering, Arizona State University, in August 2000, where he has been fulton chair professor since 2015. His research interests fall in the general field of information networks and data science, including communication networks, machine learning for Internet of Things (IoT), fog/edge computing, optimization/control of cyber physical systems, and smart grid. He is a recipient of the ONR Young Investigator Award in 2005 and the NSF CAREER award in 2003. His papers have won a few awards, including the Best Student Paper at WiOPT 2018, the Kenneth C. Sevcik Outstanding Student Paper Award of ACM SIGMETRICS 2016, the Best Paper Runner-up Award of IEEE INFOCOM 2009 and IEEE INFOCOM 2014, and the Best Paper Award at IEEE ICC 2008 and ICC 2017. Building on his research findings, he co-founded Smartiply Inc., in 2015, a fog computing startup company delivering boosted network connectivity and embedded artificial intelligence. He was TPC co-chair for a few major conferences in computer networks, including IEEE INFOCOM 2012 and ACM MOBIHOC 2015. He was general chair for ACM/IEEE SEC 2017 and WiOPT 2016.

Jiguo Yu (Senior Member, IEEE) received the PhD degree from Shandong University, Jinan, China, in 2004. He became a full professor with the School of Computer Science, Qufu Normal University, Shandong, China, in 2007. Currently, he is a full professor with the Qilu University of Technology (Shandong Academy of Sciences) and Shandong Computer Science Center (National Supercomputer Center in Jinan). His main research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. Particularly, he is interested in designing and analyzing algorithms for many computationally hard problems in networks. He is a member of the ACM and a senior member of the China Computer Federation (CCF).