Mining Pool Selection Problem in the Presence of Block Withholding Attack

Kentaro Fujita, Yuanyu Zhang, Masahiro Sasabe, and Shoji Kasahara
Graduate School of Science and Technology, Nara Institute of Science and Technology,
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan.
fujita.kentaro.fk0@is.naist.jp, {yy90zhang, m-sasabe, kasahara}@ieee.org

***Abstract*—Mining, the process where multiple miners compete to add blocks to Proof-of-Work (PoW) blockchains, is of great importance to maintain the tamper-resistance feature of blockchains. In current blockchain networks, miners usually form groups, called mining pools, to improve their revenues. When multiple pools exist, a fundamental mining pool selection problem arises: which pool should each miner join to maximize its revenue? In addition, the existence of mining pools also leads to another critical issue, i.e., Block WithHolding (BWH) attack, where a pool sends some of its miners as spies to another pool to gain extra revenues without contributing to the mining of the infiltrated pool. This paper therefore aims to investigate the mining pool selection issue (i.e., the stable population distribution of miners in the pools) in the presence of BWH attack from the perspective of evolutionary game theory. We first derive the expected revenue density of each pool to determine the expected payoff of miners in that pool. Based on the expected payoffs, we formulate replicator dynamics to represent the growth rates of the populations in all pools. Using the replicator dynamics, we obtain the rest points of the growth rates and discuss their stability to identify the Evolutionarily Stable States (ESSs) (i.e., stable population distributions) of the game. Simulation and numerical results are also provided to corroborate our analysis and to illustrate the theoretical findings.**

***Index Terms*—Blockchain, Mining Pool, Evolutionary Game Theory, Block Withholding Attack**

I. Introduction

Blockchain, the key enabler of modern crypto currency systems like Bitcoin [1], has been identified as the most representative distributed ledger technology. Although initially developed for managing financial transactions, blockchain has also found potential applications in other fields, like access control [2], data sharing [3] and supply chain management [4]. Blockchain, at its core, is a distributed database managed over a Peer-to-Peer (P2P) network, where the data (i.e., transactions) are stored in blocks, which are chained together through cryptographic hashes. In addition to a set of transactions, a block also contains a header, which consists of the Merkle tree root of the transactions, the hash value of the previous block header, a timestamp indicating the block generation time, a *Difficulty* representing the difficulty of mining and a *Nonce* used in the mining. A block is usually limited in size. For example, in Bitcoin, the block size limit is about 1 MB [5].

Blocks are appended to the blockchain at regular intervals (e.g., about 10 minutes in Bitcoin), which is achieved by a process called mining. In each round of the mining process, peers called miners compete to generate the next valid block. To be specific, a miner first collects a set of transactions and encapsulate them along with other fields (e.g., hash of previous block, timestamp, *Difficulty* and *Nonce*) into a block [6]. Next, the miner repeats to calculate the hash of the block header by varying the *Nonce* value, until it finds a valid block whose hash value satisfies the *Difficulty* condition. These two steps are conducted by all miners and the miner that first finds a valid block wins in this round and will be rewarded a certain amount of money [7].

Mining requires a huge amount of computation power and thus it is difficult for individual miners to gain revenues regularly in general. Therefore, in current blockchain networks, miners usually form groups called mining pools and aggregate their computation power to improve their revenues. A pool has a manager, who sets another relaxed *Difficulty* that is easier to satisfy. All miners in this pool are required to find the blocks that meet the relaxed *Difficulty* and report them to the manager. The number of reported blocks will be used to measure the contributions of the miners. Once the manager receives a valid block (i.e., a block whose hash value satisfies the true *Difficulty*), it broadcasts the valid block to the whole P2P network. If the pool wins, the revenue will be distributed to all its miners according to their contributions. Therefore, when multiple mining pools exist in the network, the fundamental mining pool selection problem arises: which pool should each miner join to maximize its revenue?

Despite the benefits brought by mining pools, a pool can infiltrate another to illegally gain more revenue by launching the so-called Block WithHolding (BWH) attack [8]. Fig. 1 illustrates a typical BWH attack in the case with two pools, where one pool (say Pool 1) attacks the other (say Pool 2). When launching the BWH attack, Pool 1 sends some of its miners as spies to Pool 2. Like normal miners, the spies participate in the mining process of Pool 2, reporting non-valid blocks and receiving revenues. However, when the spies find valid blocks, they will not report them to the manager of Pool 2. Instead, they will withdraw the blocks. In addition, after receiving the revenues from Pool 2, the spies send them back to Pool 1, which then redistributes the revenues among its miners including the spies. Under this typical BWH attack, the authors in [9] investigated the impacts of BWH attack on the revenues of mining pools. In particular, they focused on

978-0-7381-0495-9/20/$31.00 ©2020 IEEE
DOI 10.1109/Blockchain50366.2020.00047

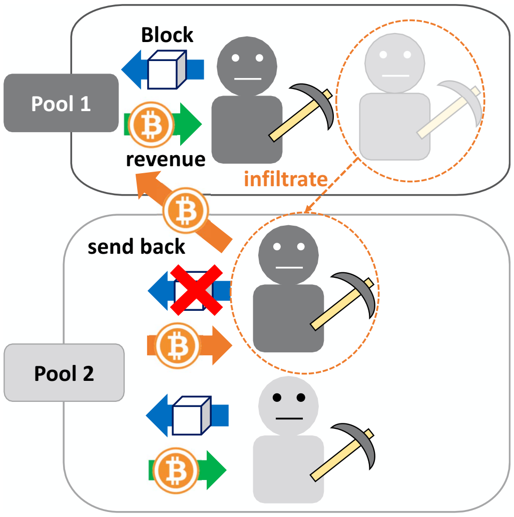


Fig. 1: Illustration of BWH attack.

the case with two pools and modeled the miner's revenue density for each pool. The results showed that pools can improve their revenues by launching the BWH attack to other pools.

Without considering the BWH attack, the authors in [10] addressed the mining pool selection issue (i.e., the population state of miners in the pools) from the perspective of evolutionary game theory [11]. The mining pool selection issue in the presence of BWH attack has also been investigated in [12] based on the evolutionary game theory as well. However, for simplicity of analysis, the authors ignored the revenue from infiltrated pools, rendering the incentive for launching the BWH questionable. This paper therefore takes the revenue from infiltrated pools into consideration and solves the mining pool selection problem in the presence of BWH attack. We first derive the expected revenue density of each pool to determine the expected payoff of miners in that pool. Based on the expected payoffs, we formulate replicator dynamics to represent the growth rates of the populations in all pools. Like [10] and [12], for simplicity, we then focus on the case with two pools where one pool attacks the other. Using the replicator dynamics, we obtain the rest points of the growth rates for the special case and discuss their stability to identify the Evolutionarily Stable State (ESS) (i.e., stable population states) of the game. Simulation and numerical results are also provided to corroborate our analysis and to illustrate the theoretical findings.

The rest of the paper is organized as follows. Section II introduces the related work. In Section III, we model the mining pool selection problem in the presence of the BWH attack as an evolutionary game. We focus on the analysis of the special case in Section IV. In Section V, we provide simulation and numerical results for the special case. Finally, we conclude this paper in Section VI.

II. Related Work

Game theory has been recognized as the most widely used approach to model interactions among rational miners or mining pools in the blockchain network [13]. Classical models include non-cooperative game, extensive-form game, coalition formation games and evolutionary game, which have been applied to address different problems in the blockchain context. For example, non-cooperative games can be applied to the BWH attack [9], [14], coalition formation games have found applications in the mining pool formation problem, i.e., how miners form mining pools [15], [16] and evolutionary games are popular in the mining pool selection problem to investigate the pool selection behaviors of miners [10], [12]. In this section, we mainly focus on the studies that applied game theory (especially the evolutionary game model) to analyze the mining pool selection problem in the presence and absence of BWH attacks. For the detailed introduction of the application of game theory to the blockchain, the readers are referred to [13].

A. *Mining Pool Selection without BWH Attack*

The authors in [10] formulated the mining pool selection problem as an evolutionary game, while they focused on a case without the BWH attack. Each pool adopts different parameters (e.g. , block size, minimum required hash rate) as its strategies to attract miners to join. Based on these parameters, each miner selects the pool to join to maximize its revenue/payoff. As a result, each pool attracts a fraction of the miners and the resultant population states of miners are of great interest. To investigate the properties of the population state, the authors applied the evolutionary game theory to model the pool selection process of miners and determined the ESS (i.e., stable population state) for the special case with two pools.

B. *Mining Pool Selection under BWH Attack*

The authors in [12] considered the mining pool selection under the BWH attack. In this research, they investigated the strategies of mining pools under the BWH attack from the perspective of mining pool administrators. The strategies of mining pool are the population of attacker and the size of blocks to be mined by pool.Based on these parameters, miners measure their profits and select pools to join. The authors also applied the evolutionary game theory to model the pool selection in order to investigate the properties of the population state. They described how pool administrators change the mining strategies to drive the population of miners to ESS. However, the authors neglected the revenues from the infiltrated pools for simplicity of analysis, making the incentive for launching the BWH attack arguable.

III. Evolutionary Game Formulation

In this section, we address the mining pool selection issue in the presence of BWH attack based on the evolutionary game theory. We formulate the expected payoff of miners in Section III-A and show the details of evolutionary game-theoretic analysis for mining pool selection in Section III-B.

A. Expected Payoffs of Miners

Similar to [10], we consider a blockchain network consisting of N miners and M pools. Each pool i adopts the following parameters as its mining strategies:

- ω_i: the minimum computation power (or hash rate) of miners required to join the pool. To simplify the analysis, all miners in this pool are assumed to use the same hash rate for mining.
- s_i: the size of blocks to be mined by pool.

Let $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_M]$ denote the hash rate requirement profile of the pools. In addition, to launch the BWH attack, each pool i uses a fraction a_{ij} $(0 < a_{ij} < 1)$ of its total hash rate to infiltrate pool j. We denote the attack profile of pool i by $\boldsymbol{a}_i = [a_{i1}, \ldots, a_{iM}]$ with $\sum_{j=1,j\neq i}^{M} a_{ij} \leq 1$ and the total attack profile of all the pools by $\boldsymbol{a} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M]$. We denote the population fraction of miners in pool i by x_i $(0 \leq x_i \leq 1, \sum_{i=1}^{M} x_i = 1)$ and the population profile of the pools by $\boldsymbol{x} = [x_1, \ldots, x_M]^\top$. According to [9], the probability that pool i finds a valid block is given by

$$P_i^{mine}(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{a}) = \frac{N x_i \omega_i (1 - \sum_{j=1}^{M} a_{ij})}{\sum_{j=1}^{M} N x_j \omega_j (1 - \sum_{k=1,k\neq j}^{M} a_{jk})}. \quad (1)$$

After a pool finds a block, the pool must broadcast the block to the entire network so that it can be added to the blockchain managed by other pools. However, if multiple mining pools find different new blocks at the same time, each pool acknowledges the block that arrives first and discards the subsequent blocks. The block propagation time is determined mainly by the propagation delay on each link and the transaction verification time on each relay node. For a block of size s, the propagation delay can be modeled as $\tau_p(s) = s/(\gamma c)$, where γ is the parameter related to the scale of the network and c is the average effective channel capacity of each link. The transaction verification time can be modeled as a linear function $\tau_v(s) = bs$, where b is a parameter determined by the scale of the network and average approval time between each node. Thus, the average propagation time can be expressed as

$$\tau(s) = \tau_p(s) + \tau_v(s) = \frac{s}{\gamma c} + bs. \quad (2)$$

We assume that the average block generation interval T remains constant. Thus, the occurrence of effective block discarding due to the propagation time of blocks can be modeled as a Poisson process with mean $1/T$ [17]. As a result, the probability of orphaning block of size s is

$$P_r^{orphan}(s) = 1 - e^{-\tau(s)/T} = 1 - e^{-(\frac{s}{\gamma c} + bs)/T}.$$

From (1) and (2), the probability of pool i winning the block mining competition with a block of size s_i is

$$P_i^{win}(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a}) = P_i^{mine}(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{a}) e^{-(\frac{s_i}{\gamma c} + bs_i)/T}.$$

Once winning the mining competition, the winner pool will be rewarded a certain amount of money, which includes a fixed revenue from the coinbase of the new block and the revenue from transaction fees. We use C to denote the fixed revenue from the coinbase. We assume that the transaction size and fee are fixed, and thus the total mining revenue for pool i can be modeled a linear function of block size s_i. Denoting ρ the transaction fee per unit block size, we can formulate the revenue of pool i obtained from the transaction fees of a block of size s_i as ρs_i.

In addition to the mining revenue, each pool also receives extra attacking revenue from the infiltrated pools. In [12], in order to simplify the analysis, such attacking revenue was ignored by simply assuming that it is contained in the mining revenue. However, since the attacking revenue would motivate pools to launch BWH attacks, it should be carefully taken into consideration. Thus, this paper follows the method in [9] to formulate the attacking revenue. First, we introduce a new concept called *revenue density* to define the revenue of unit hash rate, which is calculated by dividing the total revenue (i.e., mining revenue plus attacking revenue) of a pool by its total hash rate including those of the spies. We use r_i to denote the revenue density of pool i. Given the mining revenue, attacking revenue and revenue density, we now formulate the total expected revenue of pool i as

$$\begin{aligned} R_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a}) &= (C + \rho s_i) P_i^{win}(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a}) \\ &\quad + N x_i \omega_i \sum_{j=1,j\neq i}^{M} a_{ij} r_j \end{aligned}$$

and the revenue density r_i as

$$r_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a}) = \frac{R_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a})}{N x_i \omega_i + \sum_{j=1,j\neq i}^{M} N x_j \omega_j a_{ji}}. \quad (3)$$

Thus, each miner in pool i will receive revenue of amount $\omega_i r_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a})$.

Since mining requires enormous computation power, miners must also consider the cost of power consumption due to hash calculation in the mining process. Assuming that the power charge required for unit hash rate is p, we can express the expected payoff of each miner in pool i as follows:

$$y_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a}) = (r_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a}) - p)\omega_i. \quad (4)$$

Note that miners in the same pool use the same hash rate ω_i, obtaining identical expected payoffs.

B. Evolutionary Game for Mining Pool Selection

Each miner aims to maximize the payoff given in (4) by selecting a mining pool to join. Therefore, we model mining pool selection of each miner under the BWH attack as an evolutionary game based on [10]. The game can be defined as $\mathcal{G} = <\mathcal{N}, \mathcal{M}, \boldsymbol{x}, y_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a})>$, where

- $\mathcal{N}$ is the set of miners with $|\mathcal{N}| = N$.
- $\mathcal{M} = \{1, \ldots, M\}$ is the set of mining pools.
- $\boldsymbol{x} = [x_1, \ldots, x_M]^\top \in \mathcal{X}$ is the population profile of mining pools with x_i being the population fraction of pool i and $\mathcal{X}$ denotes the set of all population profiles.
- $\{y_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a})\}_{i\in\mathcal{M}}$ is the set of miners' expected payoffs.

Each pool i uses ω_i, s_i, and $\boldsymbol{a}_i$ as its parameters to attract miners. All the parameters are pre-fixed before the game and will remain unchanged during the playing of the game.

1) Replicator Dynamics: The growth rate of the population fraction of each pool can be described by the replicator dynamics. Based on the pairwise proportional imitation protocol, the replicator dynamics of the population fractions can be expressed by the following system of Ordinary Differential Equations (ODEs) [18]:

$$\begin{aligned}\dot{x_i}(t) &= f_i(\boldsymbol{x}(t), \boldsymbol{\omega}, s_i, \boldsymbol{a})\\ &= x_i(t)(y_i(\boldsymbol{x}(t), \boldsymbol{\omega}, s_i, \boldsymbol{a}) - \overline{y}(\boldsymbol{x}(t), \boldsymbol{\omega}, \boldsymbol{s}, \boldsymbol{a})),\end{aligned} \quad (5)$$

where $\dot{x}_i(t)$ represents the growth rate of the population of pool i at time t and $\overline{y}(\boldsymbol{x}) = \sum_{i=1}^{M} y_i(\boldsymbol{x}, \boldsymbol{\omega}, s_i, \boldsymbol{a}) x_i$ represents the average expected payoff of all the miners. We can see from (5) that the population of pools whose miners' expected payoffs are larger than the average payoff will increase, meaning that these pools will attract more miners to join.

2) Nash Equilibrium: Consider the population states $\boldsymbol{x} = [x_1, \ldots, x_M]^\top \in \mathcal{X}$ as mixed strategies that each miner may choose, where each x_i denotes the probability of choosing pool i. A Nash Equilibrium (NE) is then a mixed strategy $\boldsymbol{x}^*$ that is a best reply to itself. That means under the condition that other miners choose $\boldsymbol{x}^*$, a miner cannot gain more revenue by choosing any other $\boldsymbol{x} \in \mathcal{X}$ rather than $\boldsymbol{x}^*$. Formally, an NE satisfies the following inequality [19]:

$$(\boldsymbol{x}^* - \boldsymbol{x})^\top Y(\boldsymbol{x}^*) \geq 0, \quad \forall \boldsymbol{x} \in \mathcal{X},$$

where $Y(\boldsymbol{x}) = [y_1(\boldsymbol{x}), \ldots, y_M(\boldsymbol{x})]^\top$ with $y_i(\boldsymbol{x})$ given by (4). In an NE, the temporal growth rate of the population fraction of each pool i is zero. That is, $\forall i \in \mathcal{M}, f_i(\boldsymbol{x}(t), \boldsymbol{\omega}, s_i, \boldsymbol{a}) = 0$ holds [20]. Thus, we need to solve the ODEs of the replicator dynamics (i.e., find the rest points) to identify the NEs.

3) Evolutionary Stable Strategy: Note that only the stable NEs are ESSs. Thus, to find the ESSs, we need to further investigate the stability of the NEs. Again, we consider the population states as mixed strategies. Suppose the whole population adopt the strategy $\boldsymbol{x}^*$ and there exits another mutant strategy $\boldsymbol{x}'$ trying to invade a fraction $\epsilon \in (0, \overline{\epsilon})$ of the population. Then, $\boldsymbol{x}^*$ is an ESS, if the following inequality holds:

$$\sum_{i \in \mathcal{M}} x_i^* y_i((1-\epsilon)\boldsymbol{x}^* + \epsilon \boldsymbol{x}') \geq \sum_{i \in \mathcal{M}} x_i' y_i((1-\epsilon)\boldsymbol{x}^* + \epsilon \boldsymbol{x}').$$

More precisely, $\boldsymbol{x}^*$ is an ESS if it meets the following two conditions: [19].

1) $(\boldsymbol{x}^* - \boldsymbol{x})^\top Y(\boldsymbol{x}^*) \geq 0, \quad \forall \boldsymbol{x} \in \mathcal{X}$
2) If $(\boldsymbol{x}^* - \boldsymbol{x})^\top Y(\boldsymbol{x}^*) = 0$, then $(\boldsymbol{x}^* - \boldsymbol{x})^\top Y(\boldsymbol{x}) > 0$ holds.

The first condition indicates that an ESS must be an NE. The second condition means that if a miner choosing a strategy $\boldsymbol{x}$ can gain as much revenue as a miner choosing the NE strategy when other miners choose the NE, then a miner choosing the NE must gain more revenue than a miner choosing $\boldsymbol{x}$ when other miners choose $\boldsymbol{x}$.

4) Mining Pool Selection Algorithm: Algorithm 1 shows how each miner selects the pool to join based on the pairwise proportional imitation protocol. To start the algorithm, each miner initially joins a pool at random. After the initialization, each miner i first chooses a pool $j \in \mathcal{M}$ at random and computes the expected miners' payoffs of pool j and its current pool (say k). Then, the miner will move to pool j with probability $\rho_{k,j}$. These two steps will be repeated until the population profile $\boldsymbol{x}$ converges.

Algorithm 1 Mining Pool Selection Algorithm

```
Initial:    $t \leftarrow 1$;
while $\boldsymbol{x}$ is not converged and $t <$ MAX_COUNTER do
    for all $i \in \mathcal{N}$ do
        $k \leftarrow$ Current pool of miner $i$
        $j \leftarrow \mathsf{rand}(1, M)$    ▷ choose a pool $j$ at random
        Move from pool $k$ to $j$ with probability
        $\rho_{k,j} = x_j \max(y_j(\boldsymbol{x}, \boldsymbol{\omega}, s_j, \boldsymbol{a}) - y_k(\boldsymbol{x}, \boldsymbol{\omega}, s_k, \boldsymbol{a}), 0)$

    $t \leftarrow t + 1$
```

IV. Analysis in Case of Two Mining Pools

In this section, we analyze the case of one-side attack, where one pool attacks the other. We assume that Pool 1 attacks Pool 2, i.e., $a_{12} > 0$ and $a_{21} = 0$.

1) Miner's Revenue Density: According to (3), the miner's revenue density of each pool can be expressed as follows:

$$r_1 = \frac{\alpha(1 - a_{12}) + N a_{12} r_2 (x_1 \omega_1 (1 - a_{12}) + x_2 \omega_2)}{N(x_1 \omega_1 (1 - a_{12}) + x_2 \omega_2)}, \quad (6)$$

$$r_2 = \frac{\beta \omega_2 x_2}{N(x_1 \omega_1 a_{12} + x_2 \omega_2)(x_1 \omega_1 (1 - a_{12}) + x_2 \omega_2)}, \quad (7)$$

where

$$\alpha = (C + \rho s_1) e^{-(\frac{s_1}{\gamma c} + b s_1)/T},$$

and

$$\beta = (C + \rho s_2) e^{-(\frac{s_2}{\gamma c} + b s_2)/T}.$$

2) Miner's Expected Payoff: According to (4), (6) and (7), the miner's expected payoff of each pool can be described as follows:

$$y_1 = \omega_1 \left(\frac{\alpha(1 - a_{12})(a_{12} \omega_1 x_1 + \omega_2 x_2) + a_{12} \beta \omega_2 x_2}{N((1 - a_{12}) \omega_1 x_1 + \omega_2 x_2)(a_{12} \omega_1 x_1 + \omega_2 x_2)} - p \right),$$

$$y_2 = \omega_2 \left(\frac{\beta \omega_2 x_2}{N((1 - a_{12}) \omega_1 x_1 + \omega_2 x_2)(a_{12} \omega_1 x_1 + \omega_2 x_2)} - p \right).$$

3) Replicator Dynamics: According to (5), the system of the ODEs for the replicator dynamics can be expressed as follows:

$$\dot{x_1} = x_1(y_1 - \overline{y}),$$
$$\dot{x_2} = x_2(y_2 - \overline{y}),$$

where the expected payoff of miners $\overline{y}$ can be expressed as:

$$\overline{y} = x_1 y_1 + x_2 y_2.$$

Using the fact $x_2 = 1 - x_1$, we simplify the system of ODEs into the following equation by letting $x_1 = x$ and $x_2 = 1 - x$:

$$\dot{x_1} = x(1-x)(y_1 - y_2).$$

Thus, the population state can be expressed as $(x, 1-x)$.

4) Rest Points: Solving $\dot{x_1} = 0$ yields rest points in the form of $(x_1^*, x_2^*) = (x^*, 1 - x^*)$ where x^* is given by

$$x^* \in \left\{0, 1, \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}\right\}, \tag{8}$$

where

$$\begin{aligned} A &= a_{12}^2 Np\omega_1^3 - a_{12} Np\omega_1^3 - a_{12}^2 Np\omega_2\omega_1^2 + a_{12} Np\omega_2\omega_1^2 \\ &\quad + Np\omega_2\omega_1^2 - 2Np\omega_2^2\omega_1 + Np\omega_2^3, \\ B &= \alpha a_{12}\omega_1\omega_2 - \alpha a_{12}^2\omega_1^2 + \alpha a_{12}\omega_1^2 - a_{12}\beta\omega_1\omega_2 \\ &\quad - \alpha\omega_1\omega_2 + \beta\omega_2^2 - 2Np\omega_2^3 + 3Np\omega_1\omega_2^2 - Np\omega_1^2\omega_2, \\ C &= -\alpha a_{12}\omega_1\omega_2 + a_{12}\beta\omega_1\omega_2 + \alpha\omega_1\omega_2 - \beta\omega_2^2 + Np\omega_2^3 \\ &\quad - Np\omega_1\omega_2^2. \end{aligned}$$

5) Evolutionary Stability of Rest Points: From Section III-B3, not all rest points are evolutionarily stable, and their evolutionary stability needs to be proved. However, this is extremely difficult under our game model. Thus, the evolutionary stability of the rest points are discussed based on the phase portrait of the replicator dynamics in Section V.

V. Numerical Results

In this section, we verify the correctness of the analysis by simulations and also show the impacts of BWH attack on the population states of the mining pools. We consider a blockchain network with two pools (i.e., $M = 2$) and 5000 miners. We set the propagation delay parameter as $\frac{1}{\gamma c} + b = 0.005$, the transaction fee per unit block size as $\rho = 2$, the block generation interval as $T = 600$, the fixed revenue from coinbase as $C = 1000$ and the unit power charge per hash rate as $p = 0.01$. In the simulations, we execute Algorithm 1 to obtain the population states after convergence.

A. No-Attack Case

First, we consider the case without BWH attack (i.e., $a_{12} = 0$) with pool hash rate requirements $(\omega_1, \omega_2) = (30, 20)$ and block sizes $(s_1, s_2) = (100, 100)$. Fig. 2 shows the phase portraits of the replicator dynamics. From the figure, we can see the population states converges to $(0.4, 0.6)$ for all initial states, which is consistent with the results in [10].

B. One-Side Attack Case

In this subsection, we consider the one-side attack case with attack size $a_{12} = 0.015$ from Pool 1 to Pool 2, i.e., Pool 1 attacks Pool 2. Other parameters are set as those in Fig. 2. In this case, we also consider three initial population states: $(x_1, x_2) = (0.20, 0.80)$, $(x_1, x_2) = (0.75, 0.25)$ and $(x_1, x_2) = (0.85, 0.15)$. Fig. 3a shows the change of population states over time via simulations. From the figure, it can be seen that the population state after convergence differs depending on the initial population states. The results in this figure show that the game in this scenario converges two population states: $(1, 0)$ and $(0.55, 0.45)$. Calculating (8), we obtain four rest points: $(0, 1)$, $(1, 0)$, $(0.55, 0.45)$ and $(0.77, 0.23)$. This indicates that only the rest points $(1, 0)$ and $(0.55, 0.45)$ are stable, i.e., they are the ESSs. To verify this observation, we plot the phase portraits of the replicator dynamics in Fig. 3b. We can see that the population states converge to $(0.55, 0.45)$ for initial states with $x_1 < 0.77$. For other initial states, the population states converge to $(1, 0)$, which is consistent with the observation obtained from the simulations. Comparing the stable population states of the cases with BWH attack and without BWH attack, we can see that the population fraction of Pool 1 increases from 0.4 (without BWH attack) to 0.55 or to 1.0 (with BWH attack). This indicates that launching BWH attack attracts miners to join the attacking pools.

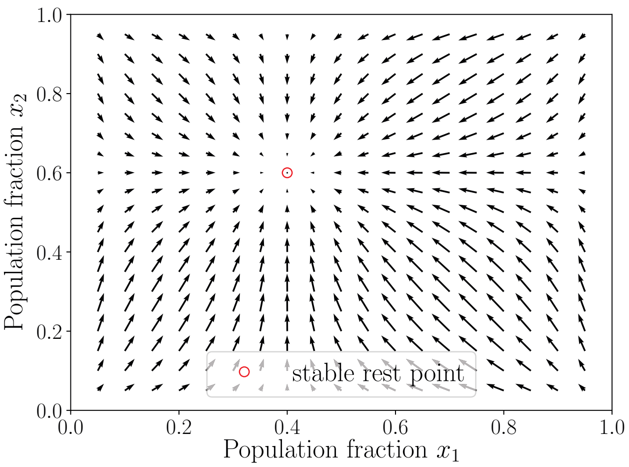
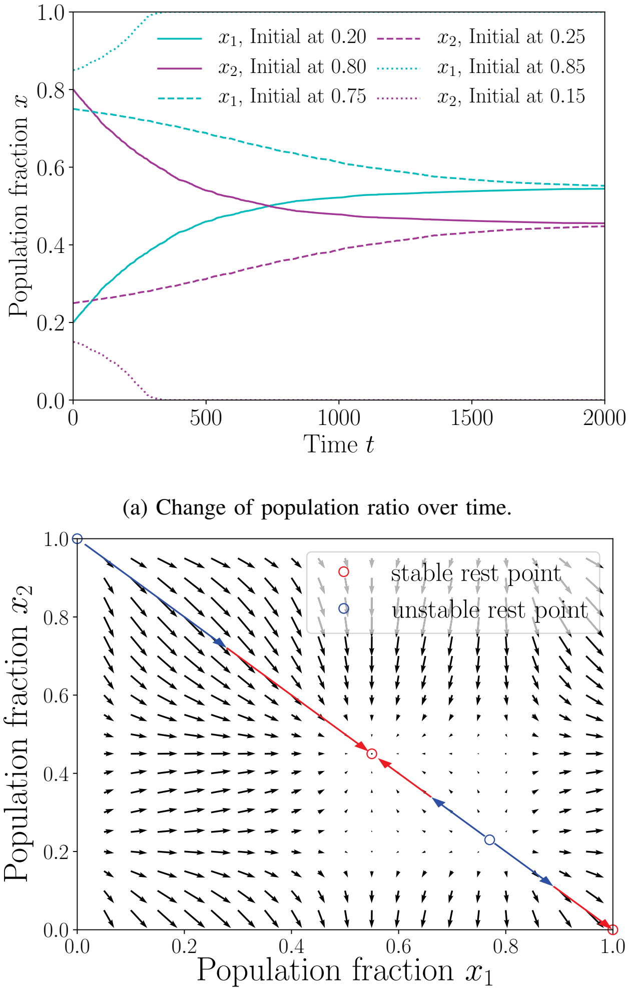


Fig. 2: Stable population states vs. attack size a_{12}.

Next, we investigate the impacts of the attack size on the mining pool selection. Fig. 4 shows how the stable population state changes when the attack size increases from 0 to 0.03. The initial state is fixed as $(0.50, 0.50)$. We can see from the figure that as the attack size increases, the population fraction of Pool 1 increases, while that of Pool 2 decreases. This implies that using more hash rate for attack attracts more miners to join the pool. We can also observe that using even only 1–3% of the total hash rate will lead to an increase of the population. This shows the significant impacts of the BWH attack on the mining pool selection of miners.

VI. Conclusions

In this paper, we investigate the mining pool selection problem in the presence of BWH attack and model the pool selection of miners by the evolutionary game theory framework. In the case with two pools, we obtained the rest points of the game and verified their stability using simulations. The results in this paper show that launching BWH attack attracts miners to join the pool, and the more hash rate (computation



(a) Change of population ratio over time.

(b) Phase portraits of replicator dynamics.

Fig. 3: One-side attack case with $a_{12} = 0.015$.

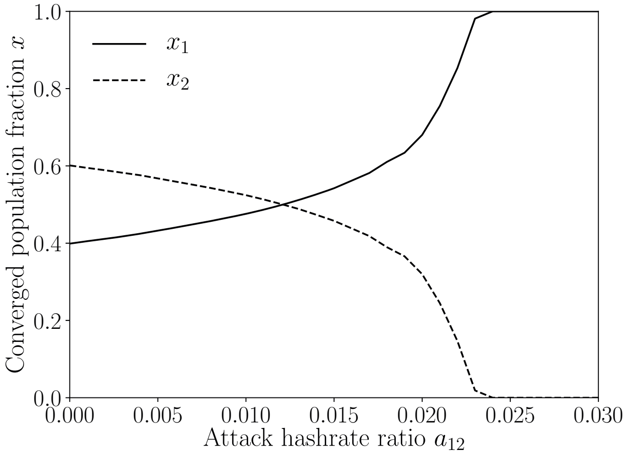


Fig. 4: Stable population states vs. attack size a_{12}.

power) used for attack, the more miners will be attracted. In addition, using even a small amount of hash rate for attack will greatly increase the population of the miners in the attacking pool, indicating the significant impacts of BWH attack on the mining pool selection of miners.

Acknowledgments

This work was supported in part by KAKENHI (A) under Grant 19H01103, SCAT Research Grant and The Telecommunications Advancement Foundation.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain Based Access Control," in *Proc. of IFIP International Conference on Distributed Applications and Interoperable Systems*, 2017, pp. 206–220.

[3] Q. I. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.

[4] N. Kshetri, "Blockchain's Roles in Meeting Key Supply Chain Management Objectives," *International Journal of Information Management*, vol. 39, pp. 80–89, 2018.

[5] F. Tschorsch and B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[6] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *Proc. of IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557–564.

[7] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden, "Incentive Compatibility of Bitcoin Mining Pool Reward Functions," in *Proc. of International Conference on Financial Cryptography and Data Security*, 2016, pp. 477–498.

[8] N. T. Courtois and L. Bahack, "On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency," *arXiv preprint arXiv:1402.1718*, pp. 1–15, 2014.

[9] I. Eyal, "The Miner's Dilemma," in *Proc. of IEEE Symposium on Security and Privacy*, 2015, pp. 89–103.

[10] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary Game for Mining Pool Selection in Blockchain Networks," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 760–763, 2018.

[11] R. B. Myerson, *Game Theory*. Harvard University Press, 2013.

[12] S. Kim and S.-G. Hahn, "Mining Pool Manipulation in Blockchain Network Over Evolutionary Block Withholding Attack," *IEEE Access*, vol. 7, pp. 144 230–144 244, 2019.

[13] Z. Liu, N. C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "A Survey on Applications of Game Theory in Blockchain," *arXiv preprint arXiv:1902.10865*, 2019.

[14] 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. of 2015 IEEE 28th Computer Security Foundations Symposium*. IEEE, 2015, pp. 397–411.

[15] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proc. of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 919–927.

[16] L. Brünjes, A. Kiayias, E. Koutsoupias, and A.-P. Stouka, "Reward sharing schemes for stake pools," *arXiv preprint arXiv:1807.11218*, 2018.

[17] J. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," in *Proc. of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2015, pp. 281–310.

[18] J. W. Weibull, *Evolutionary Game Theory*. MIT Press, 1997.

[19] J. Hofbauer and W. H. Sandholm, "Stable Games and Their Dynamics," *Journal of Economic Theory*, vol. 144, no. 4, pp. 1665–1693, 2009.

[20] J. Hofbauer and K. Sigmund, "Evolutionary Game Dynamics," *Bulletin of the American Mathematical Society*, vol. 40, no. 4, pp. 479–519, 2003.