Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(1): 309–327.
DOI: 10.3934/mbe.2020017
Received: 19 August 2019
Accepted: 25 September 2019
Published: 09 October 2019

Research article

A type of block withholding delay attack and the countermeasure based on type-2 fuzzy inference

Liang Liu[1,2], Wen Chen[1,*], Lei Zhang[1], JiaYong Liu[1] and Jian Qin[3]

[1] College of Cybersecurity, Sichuan University, Chengdu, China
[2] College of Electronics and Information Engineering, Sichuan University, Chengdu, China
[3] Chengdu No.7 high school, Chengdu, China

* **Correspondence:** Email: wenchen@scu.edu.cn; Tel: +86-28-85405568.

Abstract: We proposed a new type of bitcoin withholding attack named block withholding delay (BWD). It is different from the traditional withholding attacks which always drop valid blocks. BWD attackers never discard blocks but they delay the submissions of blocks to the pool managers, resulting the pool failed in the mining competitions and loss of rewards. We analyzed the optimum strategy of a BWD attacker who split its computing power into two parts, one was utilized to launch BWD attacks on the victim pools, while the other part was used for solo mining. We present detailed quantitative analysis of the maximum incentive that an attacker can earn by carefully splitting its computing power, and demonstrated that the attacker can obtain higher incentives than its contribution to the network in different conditions. Furthermore, we proposed a countermeasure against BWD based on the interval type-2 Takagi-Sugeno-Kang fuzzy inference system (IT2-TSK-FIS). The principle is to modify the private payoff scheme of pools to increase the risk of losing revenues of the rogue miners who deliberately delay block submissions. The scheme dealing the uncertain cause of block delay using fuzzy inference, and it is so designed that it does not require modifications of public mining protocols or data structures of the bitcoin network, which makes it applicable in practical pools.

Keywords: Bitcoin security; block delay; block withholding attack; pool mining; selfish miner

1. Introduction

Bitcoin is a worldwide cryptocurrency, which was proposed by Nakamoto [1] in 2008. Compared with traditional currency, bitcoin has many advantages, including easy authentication and transaction, anonymous, and etc [2]. In addition, the coins are generated by distributed mining of blocks in the bitcoin community [3], which does not depend on the permits of governments or banks, thus they cannot be excessively issued and naturally have the property of anti-inflation [4]. Bitcoin has become

important social wealth, according to the report from CoinDesk [5], the value of bitcoin is over 300 billion dollars.

Salah and et al. [6, 7] proposed that blockchain based technologies, such as smart contracts, can be key enablers to solve many security problems including internet, IoT and etc. However, due to the open nature of blockchain, bitcoin is vulnerable to some kinds of attacks [8, 9], including 51% attack, double-spending, selfish attack, network layer attack, stubborn mining and etc. A 51% attack [10] refers an attack on the blockchain, for which such an attack is launched by a group of miners controlling more than 50% of the network's mining hashrate (computing power). The attackers would be able to prevent new transactions from gaining confirmations, allowing them to discard others' transactions. They would also be able to reverse transactions that were completed, meaning they could even double-spend bitcoins [11, 12]. The attacker first mined a block, but instead of broadcasting this block to the network, it surreptitiously mines atop the block to form a private branch. The attacker then spends some bitcoins on the original branch, while issuing a conflicting transaction on its private branch. For the attack to succeed, the private branch must keep up with and overtake the original branch after at least six conformations have been reached, which makes it hard to be implied in the real bitcoin community. However, recently, some new variation such as fast payment double spending has become a real threaten to the bitcoin users.

Salah [13] discussed open research challenges of utilizing blockchain technologies, and warnned that the decentralized power found in blockchain can suffer from abuses and misuses. Selfish attack is another potential threat found by Eyal [14]. With selfish mining, the attackers keep a secret chain to themselves and keep mining on the top of the chain until it is ahead of the public fork. The purpose of selfish mining is to waste the efforts of honest miners mining on the public fork and increase the attackers' relative hashrate and revenues. Eyal proved that through selfish mining the hashrate needed to dominate the blockchain can be largely reduced from 51% to 25% [14]. Hasan [15] proposed that the attacker may assumes many illegitimate and fake identities with the purpose of gaining more control and influence within a community of users.

Some network attacks try to undermine the P2P connections of bitcoin networks and control the transmissions of coin transactions. The typical attacks on the bitcoin network layer include DDOS [16], eclipse [17] and etc. Furthermore, Kartik [18] proposed stubborn mining, which is a combination of selfish mining and network-level attacks. During stubborn mining even if the public fork is ahead, the attackers still keep mining on the private chain with the purpose of catching up through eclipse attacks on the network layer.

It should be noted that among all the types of bitcoin attacks, bitcoin withholding attack (BWH) [19, 20] is a real threat, which has been publicly confirmed by the bitcoin community. A BWH attack conducted on a bitcoin pool in April 2014 indeed be well-incentivized, and the attack caused nearly 200,000 USD in damage to the victim pool [22]. The BWH attackers join a victim pool, imitating honest miners and sharing the reward from the pool, but they never submit valid blocks to the pool manager, which seriously undermine the hashrate and interests of the pool. In [23] the BWH attackers conspire with some malicious pools, which sponsor the attackers to attack opponents and increase the win probability of themselves. In [24] the attackers split their computing powers, and launch BWH attacks on a set of victim pools while keep mining on their private pools to maximize the revenues. Therefore, the attackers earn higher incentives at the expense of other honest miners's revenue. Courtois [19] studied the detection methods of BWH and pointed out that the pool managers can try to

detect the attacks by observing the deviation between the suspicious miners' contributions and their computing powers. Rosenfeld [20] suggested a fix of the block structure that would allow the pool managers to detect BWH attacks by a honey-pot technique. Liu and et al. [21] proposed a generalized model in BWH attack based on game theory, the discussed the reward and cost of a BWH attacker in a scenario that only two participants in a pool. They deeply discussed the equilibrium of the two parts in different resources allocation cases both with pure strategy and mixed strategy. Furthermore, they also analyzed the possible way to counter against BWH by information asymmetry based on information conceal mechanism.

In this paper we discussed a new type of bitcoin withholding attack that the rogue miners deliberately delay the submission of valid blocks, named block withholding delay (BWD). The BWD attack is proposed based on the principle of bitcoin consensus rules [1]: in each mining cycle, the miners continue mining after the received valid block on the top of the blockchain, and the latter arrived blocks will be discard from the main chain ultimately. Therefore, although the BWD attackers finally submitted blocks to the pool managers, the submission delay still results in discarding of the blocks without reward from the community.

BWD differs from traditional BWH in the following ways:(a) BWH drops all of the valid blocks, on the contrary BWD always submits blocks to the pool managers; (b) the BWD attackers try to delay block submissions to prevent the victim pool from claiming rewards of the blocks. The BWD attackers share reward of the pool members but nearly make no contribution to the pool. Therefore compared with BWH, BWD is a more concealed attack, which undermines the fairness of the pools and discredit the bitcoin ecosystem in a secret manner.

1.1. Related work

According to the mining rules of bitcoin [1], in each mining cycle the transactions of the network are collected in a publicly verifiable ledger named a block chain. The participants in the bitcoin community are called miners who are responsible for solving cryptographic puzzles as a proof-of-work (PoW), which contains the hash of a new block. If the first n bits of the hash value are continuous zeros, then the block is valid and the finder can claim reward from the bitcoin network. The number n is called mining difficult and it is dynamically adjusted to regulate the flow of bitcoins, such that a valid block is created once in approximately 10 minutes.

With the increasing difficulty of bitcoin mining, the miners join mining pools to complete the PoW tasks together so as to obtain stable incomes. Each pool runs a private payoff protocol to estimate the computing power of pool members using partial work proof as a basis for allocating profits [24]. However, the power evaluations may not match with the miners' real contributions to the pools, and some dishonest miners may claim excessive gains by exploiting this vulnerability, such as BWH attacks proposed by Rosenfeld in 2011 [20]. The BWH attackers in a victim pool discard valid blocks, so as to degenerate the pool's hashrate. It was proved that by carrying out BWH attacks on a victim pool the attacker indirectly increased the rewards of other pools. Courtois [25] generalized the BWH attack introduced by Rosenfeld [20]. They proposed that the BWH attackers can earn more profit in a long term, and proved that if an attacker has α percent of the total computing power of the network measured in hashes per second, and it mines for its own account with half capacity, while apply the other half capacity to attacking pools, the excessive gain could be $\alpha/(1-4\alpha)$. Bag [24] focus on the optimal strategies of a BWH attacker who does private mining with a fixed fraction of its mining power

and uses the rest of its power in attacking pools. They derived various expressions for the attackers' revenues under different settings and proposed optimum strategies for their power splitting based on the distribution of pools' hashrate in the bitcoin community.

Some researchers try to analysis the pool competitions under BWH attacks using game models. Laszka [26] provided a game theoretic analysis of the BWH attack and discussed the long term viability of a Nash equilibrium between pools. Luu [22] investigated the utility of BWH using a general CPS (computing power splitting) game model of bitcoin, and evaluated the maximum reward of the attackers under different scenarios. They proposed that the attackers always obtain more reward by mining dishonestly regardless of their mining powers. Therefore, the attackers have an incentive to carry out BWH attacks. Eyal [23] analyzed a game where identical mining pools attack each other. They demonstrated that there does not exist a Nash equilibrium when one of the pools is attacked by BWH, finally all the pools face a version of the Prisoner's Dilemma, resulting all of the pools have to attack each other and earn less than what they should have if none had attacked. These works have shown that block withholding attack is seriously harmful for the bitcoin community, and the ultimate consequence of the attacking is to degenerate the credit and profitability of the whole community.

In the paper, we proposed a new type of withholding delay attack BWD. The attackers join pools and pretend to be honest miners, but they delay the block submissions, resulting in the victims' failure of the mining competitions. The rest of the paper is organized as follows, in section 2, we analyzed the incentives of an attacker whose target is a single pool, and discussed the conditions of excessive incentives. In section 3, we investigated the revenues of an attacker when it divided its power to attack multiple pools simultaneously, we also analyzed the optimum power splitting strategy using a differential evolution algorithm. In section 4, we simulated the BWD attacks based on the real bitcoin pools' sizes and computing powers to analyze the excessive incentives; finally, we initiated a study on effective countermeasures against BWD attacks.

1.2. Contribution

The contribution of this paper including two aspects:

- Proposed a new type of bitcoin withholding attack BWD. It was the first time to analyze bitcoin withholding delay attack in the pools, which is more concealed than traditional BWH attacks. BWD was systematically studied under different conditions, and the maximum excessive revenues of the attackers were quantitatively analyzed. We also explained why block withholding delay can be well-incentivized for the rogue miners, providing an algorithmic strategy to gain higher rewards than honest mining. We confirmed our findings by experiments simulated the power distribution of current real bitcoin pools.
- A countermeasure against BWD. We proposed a new payoff scheme of pools based on type-2 TSK fuzzy inference system, which dynamically distribute reward to the pool members according to their fuzzy delay times. The underlying idea is to increase the risk of attacker's loss of revenues, such that the more number of delay the less reward it will earn. We proved that the schemes can make bitcoin mining immune against block withholding delay attack without change of the existing mining rules.

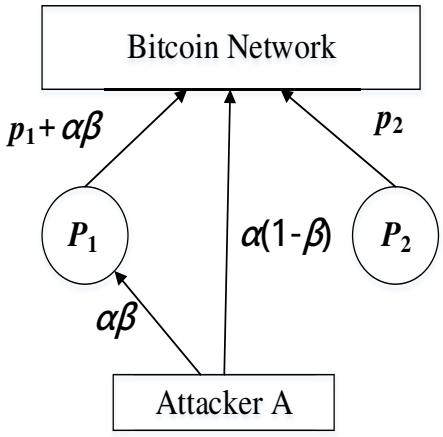

Figure 1. The BWD attacks on a single pool with respect to an attacker with α fraction of total computing power in the network.

2. BWD attacking on a single pool

In this section we investigate the incentives of an attacker who attacks a single pool in the bitcoin network. Let us simplify the problem and consider a situation when there are only two mining pools P_1 and P_2 in the entire network and a BWD attacker A. As shown in Figure 1, P_1 is the victim pool, and other miners in the network constitute another pool P_2 .The computing power of P_1, P_2 and A is p_1, p_2 and α respectively. And the total computing power in the network is

$$p_1 + p_2 + \alpha = 1 \tag{2.1}$$

The attacker A pretends to be an honest miner in P_1 with β fraction of its computing power for attacking and the rest of its computing power is utilized to mining solo. It should be remind that when A is mining in P_1, it cannot directly submit valid blocks to the bitcoin system, because the coinbase transaction contained in the blocks will cause the mining reward to go directly to the wallet of P_1's manager [22]. If A indeed found a valid block at time t, it does not immediately submit the block to P_1, actually it intentionally delays the submission until the following two cases: 1)it's at the expiration time t_e of the mining cycle; 2)some other miners have broadcasted their findings of valid blocks at time $t < t_e$, in this case A will submit its block at time t_s which uniformly distributes in $[t, t_e]$ to make the submission delay irregular and hard to detect. In both cases the attacker finally submit valid blocks but only in case 2 its computing powers is really wasted.

Let E_A and $E_i, i = 1, 2$ represent the events that A or any honest miner in P_i found a valid block respectively. To simplify the discussion, we ignore the extremely low probability event that both of the two split parts of A's computing power mined a valid block respectively in one mining cycle. Then the probability of A's power wasting is:

$$P_w = P(E_1, E_2, E_A) + P(\bar{E_1}, E_2, E_A) + P(E_1, \bar{E_2}, E_A) \tag{2.2}$$

The attacker A and the pool members of P_1 and P_2 mining blocks independently, thus $P(E_1, E_2, E_A) = P(E_1) \cdot P(E_2) \cdot P(A) = p_1 \cdot p_2 \cdot \alpha\beta$. Similarly $P(\bar{E_1}, E_2, E_A) = (1 - p_1) \cdot p_2 \cdot \alpha\beta$ and $P(E_1, \bar{E}2, E_A) = p_1 \cdot (1 - p_2) \cdot \alpha\beta$. Finally we get $P_w = \alpha\beta(p_1 + p_2 - p_1 \cdot p_2)$.

Theorem 1. *The gain of P_1 when it is attacked by the BWD attacker A is given by* $\frac{p_1+\alpha\beta(1-P_w\alpha\beta)}{1-P_w\alpha\beta} \cdot \frac{p_1}{p_1+\alpha\beta}$

Proof. If the attacker A does not use its computing power to launch BWD attacks on P_1, the computing work force of the entire network would have been 1. But when A use β fraction of its computing power for attacking P_1, the wasted power of A is $P_w \cdot \alpha\beta$ and the effective computing power of the network is reduced to $1 - P_w \cdot \alpha\beta$. Furthermore, P_1 still expected that its computation power is $p_1 + \alpha\beta$ while its real power is $p_1 + \alpha\beta(1 - P_w)$. According to the payoff schemes, P_1 will give $\frac{\alpha\beta}{p_1+\alpha\beta}$ fraction of its revenue to A, and hence the remain profit of P_1 would be $\frac{p_1+\alpha\beta(1-P_w)}{1-P_w\alpha\beta} \cdot \frac{p_1}{p_1+\alpha\beta}$. □

Let's calculate the gain of A. From the proof of Theorem 1 we know that the active computing power of the network is $1 - P_w \cdot \alpha\beta$. Apparently the gain of A is twofold. Firstly, the gain from mining privately is equal to $G_1 = \frac{\alpha(1-\beta)}{1-P_w\alpha\beta}$, and the expected share of incentive from P_1 is $G_2 = \frac{p_1+\alpha\beta(1-P_w\alpha\beta)}{1-P_w\alpha\beta} \cdot \frac{\alpha\beta}{p_1+\alpha\beta}$, therefore, the total revenue of A is

$$\begin{aligned} G_A = G_1 + G_2 &= \frac{\alpha(1-\beta)}{1-P_w\alpha\beta} + \frac{p_1+\alpha\beta(1-P_w)}{1-P_w\alpha\beta} \cdot \frac{\alpha\beta}{p_1+\alpha\beta} \\ &= \frac{\alpha(p_1+\alpha\beta-P_w\alpha\beta^2)}{(1-P_w\alpha\beta)\cdot(p_1+\alpha\beta)} \end{aligned} \tag{2.3}$$

It is easy to see that, when A launches BWD attacks on P_1, the gain of P_2 is $\frac{p_2}{1-P_w\cdot\alpha\beta}$, which is higher than that when A is mining honestly. As its relative computing power is increased by A' wasting of computing powers, P_2 would indirectly benefit from the attacks on P_1. Actually the following theorem shows that if A does not carefully split its computing power, the attack will just benefit P_2, and the gain from the attacking may not compensate the loss of A caused by the power wasting.

Theorem 2. *When the attacker A launches BWD attacks on P_1, it will earn excessive incentives if* $0 < \beta < \frac{p_1}{1-\alpha}$.

Proof. We know that when A takes honest strategy, its gain is $G_H = \alpha$. While A launches BWD attacks on P_1, its gain G_A compared with G_H is $G_A/G_H = \frac{(p_1+\alpha\beta-P_w\alpha\beta^2)}{(1-P_w\alpha\beta)\cdot(p_1+\alpha\beta)}$. Obviously, A can obtain excessive incentives when $G_A/G_H > 1$, which implies that:

$p_1 + \alpha\beta - P_w\alpha\beta^2 > (1 - P_w\alpha\beta) \cdot (p_1 + \alpha\beta)$
$\Leftrightarrow P_w\alpha\beta(p_1 + \alpha\beta - \beta) > 0$
$\Leftrightarrow p_1 > \beta(1 - \alpha)$
$\Leftrightarrow p_1/(1 - \alpha) > \beta$
Therefore, we proved that the attacker can get excessive incentives when $0 < \beta < \frac{p_1}{1-\alpha}$ □

As an example in Figure 2, we calculated the excessive incentives of A using $Inc = (G_A - G_H) * M$, M is the worth of bitcoins, (when we writing this paper a single bitcoin is worth 3,418 USD and nearly 20% of the 21 million bitcoins are to be mined). From the figure we can see that, Inc increases when α and β satisfy $\beta < p_1/(1-\alpha)$, and the maximum of Inc is more than 125.64 million dollars. However, if β continually increases, Inc begins to decrease and even becomes negative. The reason is that when β is too large, A has to use most of its power for BWD attacking, then both P_1 and A generate less blocks, resulting the incentive from P_1 cannot compensate the loss of A caused by its wasting of powers.

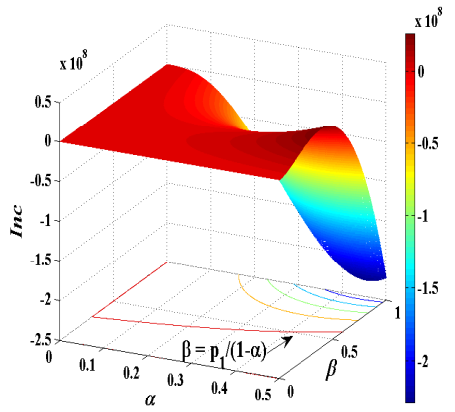

Figure 2. GA. with different values of α and β, with $p_1 = 0.3$.

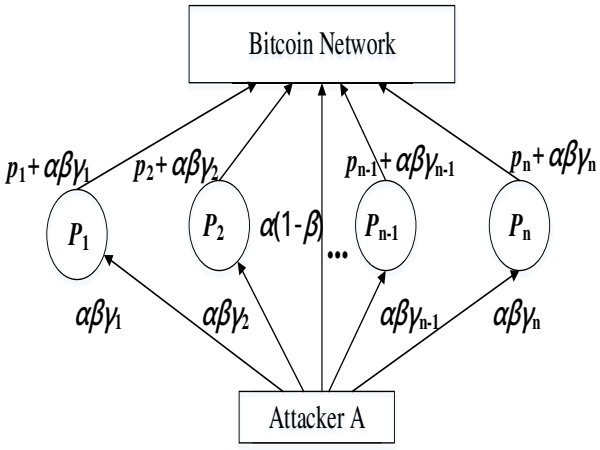

Figure 3. The BWD attacking on multiple pools with respect to an attacker with α fraction of total computing power in the network.

3. BWD attacking on multiple pools

Since A can acquire additional incentives by attacking a single pool, it may try to attack other pools to maximize its incomes by similar strategies. Now we consider a more complicated situation that the attacker A attacks multiple pools simultaneously. As shown in Figure 3, there are n pools, namely $P_1, P_2, ..., P_n$, each with computing power p_i, and satisfy $\sum_{i=1}^{n} p_i + \alpha = 1$. Similarly, A uses $1-\beta$ fraction of its computing power for mining solo while the remaining part is divided, such that $\beta\gamma_i$ fraction of its computing power is used for attacking pool P_i , and satisfy $\sum_{i=1}^{n} \gamma_i = 1$.

If A finds a valid block in pool P_i, it will delay the block submission till other valid blocks have been reported in the networks; but if no other blocks are broadcasted at the end of the mining cycle, then A has to submit its block to P_i to avoid exposure of its attacking behaviors. In this manner, A wastes its mining power in P_i with probability

$$P_{wi} = \alpha\beta\gamma_i \cdot (1 - \prod_{i=1}^{n}(1 - p_i)) \tag{3.1}$$

where $\alpha\beta\gamma_i$ is the probability of A finds a valid block in P_i, and $1-\prod_{i=1}^{n}(1-p_i)$ represents the probability that at least one pool reported a valid block in a mining cycle. Therefore, the total mining power of the network is $1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}$.

The incentive of A consists of the gain from mining privately $\frac{\alpha(1-\beta)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}}$, and the reward from pools $\sum_{i=1}^{n}(\frac{p_i+\alpha\beta\gamma_i(1-Pwi)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}}\cdot\frac{\alpha\beta\gamma_i}{p_i+\alpha\beta\gamma_i})$, where $\alpha\beta\gamma_i(1-Pwi)$ is the effective power of A to make contribution to P_i. However, the pool P_i still believes that A contributes $\frac{\alpha\beta\gamma_i}{p_i+\alpha\beta\gamma_i}$ fraction of the pool's computing powers, as it does not know A's attacking behaviors.

From the previous section, we can see that the factor β is important for the attacker to earn excessive incentives from the pools, and if β is not set properly then the gain may less than that of honest strategy. Therefore, we will further discuss the condition of β for excessive incentives when A attacks multiple pools.

Theorem 3. *When the attacker A launches BWD attacks on P_1, P_2, ... P_n, it will earn excess incentives if $0<\beta<\frac{p_i}{\gamma_i(1-\alpha)}$, for $i=1,2,...,n$.*

Proof. The gain of the attacker A when it launches BWD attacks on the n pools is $G_A=\frac{\alpha(1-\beta)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}}+\sum_{i=1}^{n}(\frac{p_i+\alpha\beta\gamma_i(1-Pwi)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}}\cdot\frac{\alpha\beta\gamma_i}{p_i+\alpha\beta\gamma_i})$. On the other hand when A mining honestly, its gain G_H is α. Therefore, A can obtain excessive incentive if $G_A/G_H>1$, which means

$$\frac{(1-\beta)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}}+\sum_{i=1}^{n}(\frac{p_i+\alpha\beta\gamma_i(1-Pwi)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}}\cdot\frac{\beta\gamma_i}{p_i+\alpha\beta\gamma_i})>1 \tag{3.2}$$

Let $T_i=\alpha\gamma_i\cdot(1-\prod_{i=1}^{n}(1-p_i))$,$\alpha\sum_{i=1}^{n}\gamma_i T_i=U$, then $P_{wi}=\beta T_i$ and $1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}=1-\beta^2 U$, and (3.2) can be rewritten as

$$\frac{1-\beta}{1-\beta^2 U}+\sum_{i=1}^{n}(\frac{p_i+\alpha\beta\gamma_i(1-\beta T_i)}{1-\beta^2 U}\cdot\frac{\beta\gamma_i}{p_i+\alpha\beta\gamma_i})>1 \tag{3.3}$$

Remember that $\sum_{i=1}^{n}\gamma_i=1$, then (3.3) is transformed to

$$\begin{aligned}&\frac{1}{1-\beta^2 U}\sum_{i=1}^{n}((1-\beta)\gamma_i+\frac{(p_i+\alpha\beta\gamma_i(1-\beta T_i))\cdot\beta\gamma_i}{p_i+\alpha\beta\gamma_i})>1\\ \Leftrightarrow\ &\frac{1}{1-\beta^2 U}\sum_{i=1}^{n}(\gamma_i-\frac{\alpha\beta^3\gamma_i^2 T_i}{p_i+\alpha\beta\gamma_i})>1\\ \Leftrightarrow\ &U-\sum_{i=1}^{n}\frac{\alpha\beta\gamma_i^2 T_i}{p_i+\alpha\beta\gamma_i}>0\end{aligned} \tag{3.4}$$

Remember that $T_i=\alpha\gamma_i\cdot(1-\prod_{i=1}^{n}(1-p_i))$,$\alpha\sum_{i=1}^{n}\gamma_i T_i=U$, substitute them into (3.4) we have equally condition (3.5).

$$\begin{aligned}&\sum_{i=1}^{n}[\alpha^2\gamma_i^2(1-\prod_{i=1}^{n}(1-p_i))-\frac{\alpha^2\beta\gamma_i^3}{p_i+\alpha\beta\gamma_i}\cdot(1-\prod_{i=1}^{n}(1-p_i))]>0\\ \Leftrightarrow\ &\sum_{i=1}^{n}[\alpha^2\gamma_i^2(1-\prod_{i=1}^{n}(1-p_i))\cdot(1-\frac{\beta\gamma_i}{p_i+\alpha\beta\gamma_i})]>0\end{aligned} \tag{3.5}$$

As the left part $\alpha^2\gamma_i^2(1-\prod_{i=1}^{n}(1-p_i))$ is larger than 0, it is easy to very that if $\beta < \frac{p_i}{\gamma_i(1-\alpha)}$, for i = 1,2,...,n, then the right part $1-\frac{\beta\gamma_i}{p_i+\alpha\beta\gamma_i}$ is also bigger than 0, which means the condition (3.5) can be satisfied and the attacker A will earn excessive gains from the attacking. □

Then the optimal values of β and γ_i are discussed to estimate the maximum incentives of the BWD attacker. The combinatorial optimization problem of G_A with respect to β and γ_i are expressed as follows

$$\begin{aligned}\max_{\beta,\gamma_i} G_A =& \frac{\alpha(1-\beta)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}} + \\ & \sum_{i=1}^{n}\left(\frac{p_i+\alpha\beta\gamma_i(1-Pwi)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}}\cdot\frac{\alpha\beta\gamma_i}{p_i+\alpha\beta\gamma_i}\right) \\ subject\ to :& 0<\gamma_i<1, \sum_{i=1}^{n}\gamma_i = 1 \\ & 0<\beta<\frac{p_i}{\gamma_i(1-\alpha)}, i=1,2,...,n\end{aligned} \tag{3.6}$$

Obviously G_A is a non-monotonic function with respect to β and $\gamma_i, 1 \le i \le n$, and it is difficult to solve (3.6) through numerical methods. Therefore, a differential evolution (DE) algorithm is applied to find the optimal combination of these $n+1$ parameters.

Differential evolution (DE) algorithm is an efficient random search technology that is mainly used for continuous global optimization problems [27]. In the paper, DE is applied to search the global optimum solution of β and γ_i in (3.6).

Firstly, the possible solutions are coded into real-valued vectors (chromosomes), whose properties represent the parameters β and γ_i. Then the population evolved through mutations which ensures the variety of the individuals. Usually the lower the individual's fitness is, the more genetic modification it will have. After the mutations, some better offsprings are generated and then DE starts a new round of evolution until it reached the expected number of generations.

Initialization An initial population $X_0 = \{x_1, x_2, ..., x_m\}$ is randomly generated under the constraints of (3.6), and each individual $x_j, 0 \le j \le m$, includes n+1 elements in (3.7)

$$x_j =< p_{j0}, p_{j1}, ..., p_{jn} >, 0 \le p_{ji} \le 1 \tag{3.7}$$

where p_{j0} and $p_{ji}, 1 \le i \le n$, represent possible value of β and γ_i respectively

Fitness function In the paper, the incentives G_A is directly taken as a fitness function to evaluate the individuals as shown in (3.8).

$$\begin{aligned}f(x_j) =& \frac{\alpha(1-p_{j0})}{1-\alpha p_{j0}\sum_{i=1}^{n}p_{ji}P_{wi}} + \\ & \sum_{i=1}^{n}\left(\frac{p_i+\alpha p_{j0}p_{ji}(1-P_{wi})}{1-\alpha p_{j0}\sum_{i=1}^{n}p_{ji}P_{wi}}\cdot\frac{\alpha p_{j0}p_{ji}}{p_i+\alpha p_{j0}p_{ji}}\right)\end{aligned} \tag{3.8}$$

Mutation: It simulated the biological chromosome mutations to generate new individuals. The traditional mutation strategies include: crossover mutation, single(multi) point(s) mutation, and etc. However, remember that the solutions are subject to $0<\gamma_i<1, \sum_{i=1}^{n}\gamma_i = 1, 0<\beta<\frac{p_i}{\gamma_i(1-\alpha)}$. Therefore, we designed some new mutation rules to satisfy the constrains.

Table 1. The Parameters OF DE.

Name	Value	Description
N	300	Number of chromosomes
G	300	Number of generations
chromosome	X(1,2,...10)	$X(1)$ is β, X(2...10) represents γ_i, i=1...9
α	0.252	The hash power of attacker A
P_i, i=1...9	(0.112, 0.102, 0.081, 0.073, 0.054, 0.05, 0.05, 0.048, 0.03)	The hash power of the 9 pools
P_e	0.35	The rate of exchange strategy
P_m	0.35	The rate of modification strategy

Table 2. The best solution of power splitting.

β	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7	γ_8	γ_9	γ_{10}
49.60%	18.62%	18.52%	14.97%	14.87%	8.80%	8.55%	8.14%	5.32%	2.21%

- **Exchange strategy:** two genes p_{ji} and $p_{jk}, 1 \leq i, k \leq n$ are randomly selected from an individual x_j to exchange their positions.
- **Modification strategy:** two genes p_{ji} and $p_{jk}, 1 \leq i, k \leq n$ are randomly selected from an individual x_j, then the genes are modified such that: $\Delta = min\{\omega \cdot p_{ji}, 1 - p_{jk}\}, \omega \in [0, 1]$, and $p_{ji} = p_{ji} - \Delta, p_{jk} = p_{jk} + \Delta$
- **β-update strategy:** firstly the upper bound of β: $T = min\{\frac{p_i}{p_{ji}(1-\alpha)}, 1 \leq i \leq n\}$ is updated then a new β is randomly selected from $(0, T]$ and assigned to p_{j0} of x_j.

After the mutation process, the individuals are selected to the next generation based on their fitness. Usually the chance to be selected is proportional to the individual's fitness. Although we have given the conditions of excessive incentives of BWD both for an attack on a single pool or attacks on multiple pools simultaneously in theorem 2 and 3. It should be noted that, in the real cases the power ratios of the pools dynamically change, so do the attackers' power shares in each pool. Therefore, we just simplify the discussion in a relatively static situation, if the BWD attack is launched in real network, the new power ratio should be timely updated during the BWD process.

4. Experiment

In the first group of experiment, the BWD attacks are simulated based on the real bitcoin pools' sizes and computing powers. As shown in Figure 4 [28], currently the 10 biggest bitcoin pools P_1 to P_{10} have more than 85% computing power of the whole bitcoin networks. We assumed that P_1 ("Antpool") whose computing power is 25.2% attacks other pools using BWD. The maximum incentive G_A of P_1 was optimized by the DE algorithm with parameters shown in Table 1.

The maximum G_A in each generation is shown in Figure 5, and the final best solution is shown in Table 2. From the results we can see that using the best solution of power splitting, the attacker P_1 can increase its revenue G_A to 25.207%. Compared with the honest strategy, although the attacker only got 0.007% excessive incentives, it means about 1,004,892$ additional revenues from the BWD attacks (currently there are about 4,200,000 to be mined and each is worth 3,418$).

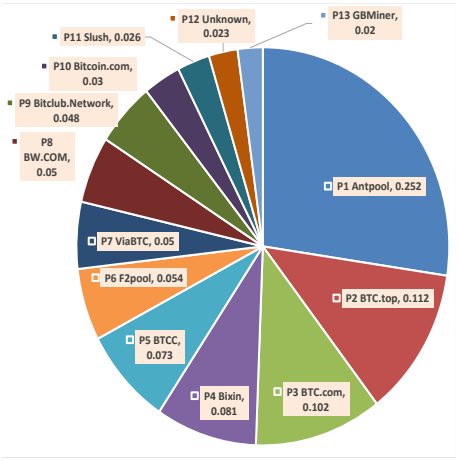

Figure 4. The power share of bitcoin Pools[28].

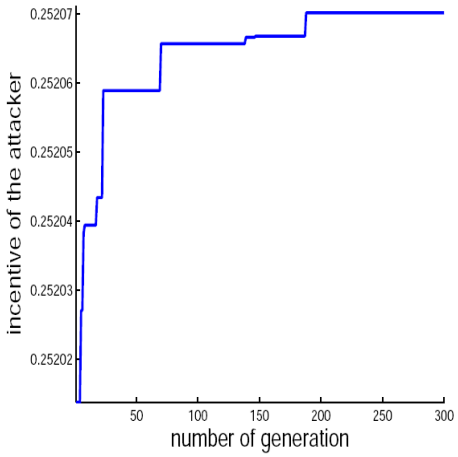

Figure 5. The maximum incentive of each generation.

To further test the optimized BWD strategy, we designed another group of experiments. Suppose that there are six pools P_1 to P_6 with computing power 0.49, 0.45, 0.40, 0.35, 0.3, and 0.25 respectively. Each time one pool P_i, $1 \leq i \leq 6$ is selected to launch BWD attacks on the others. The optimum power splitting strategy returned by DE as well as the related maximum incentives of the attackers are shown in Table 3.

From the results we can see that all the BWD attackers (even with small number of computing powers) can obtain higher revenues than their real power share of the community. Thus all the pools tend to launch BWD attacks on each other and no one would take honest strategy any more, resulting a similar situation of Prisoner's Dilemma in [23] that the pools have to attack each other and finally earn less than that if all of them have chosen honest strategy. Table 3 shows that the excessive gain of BWD increases with the attacker's computing powers, the more the powers is, the more the additional gain will be. Therefore, the big pools will more incline to get additional incentives through BWD attacks, which makes the attackers more powerful and finally the hash power and coin shares may concentrate to some huge pools. However, the optimized solutions in the table also expose that the BWD attackers are incline to select big pools as their targets. That's because usually big pools can earn more rewards, and so does the incentives obtained from these pools through BWD attacks.

Table 3. The results of BWD attacks launched by different pools.

Attacker	α	GA	Excessive gain	Best attacking strategy optimized by DE
P_1	0.49	0.5052	0.0152	$< \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6 >=< 0.4526, 0.3560, 0.1914, 0, 0 >$
P_2	0.45	0.4620	0.0120	$< \gamma_1, \gamma_3, \gamma_4, \gamma_5, \gamma_6 >=< 0.4637, 0.3194, 0.2169, 0, 0 >$
P_3	0.40	0.4084	0.0084	$< \gamma_1, \gamma_2, \gamma_4, \gamma_5, \gamma_6 >=< 0.4339, 0.3658, 0.2003, 0, 0 >$
P_4	0.35	0.3554	0.0054	$< \gamma_1, \gamma_2, \gamma_3, \gamma_5, \gamma_6 >=< 0.3979, 0.3539, 0.2481, 0, 0 >$
P_5	0.3	0.3033	0.0033	$< \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_6 >=< 0.4110, 0.3257, 0.2633, 0, 0 >$
P_6	0.25	0.2518	0.0018	$< \gamma_1, \gamma_2\ \gamma_3, \gamma_4, \gamma_5 >=< 0.3632, 0.3575, 0.2782, 0.0009, 0.0001 >$

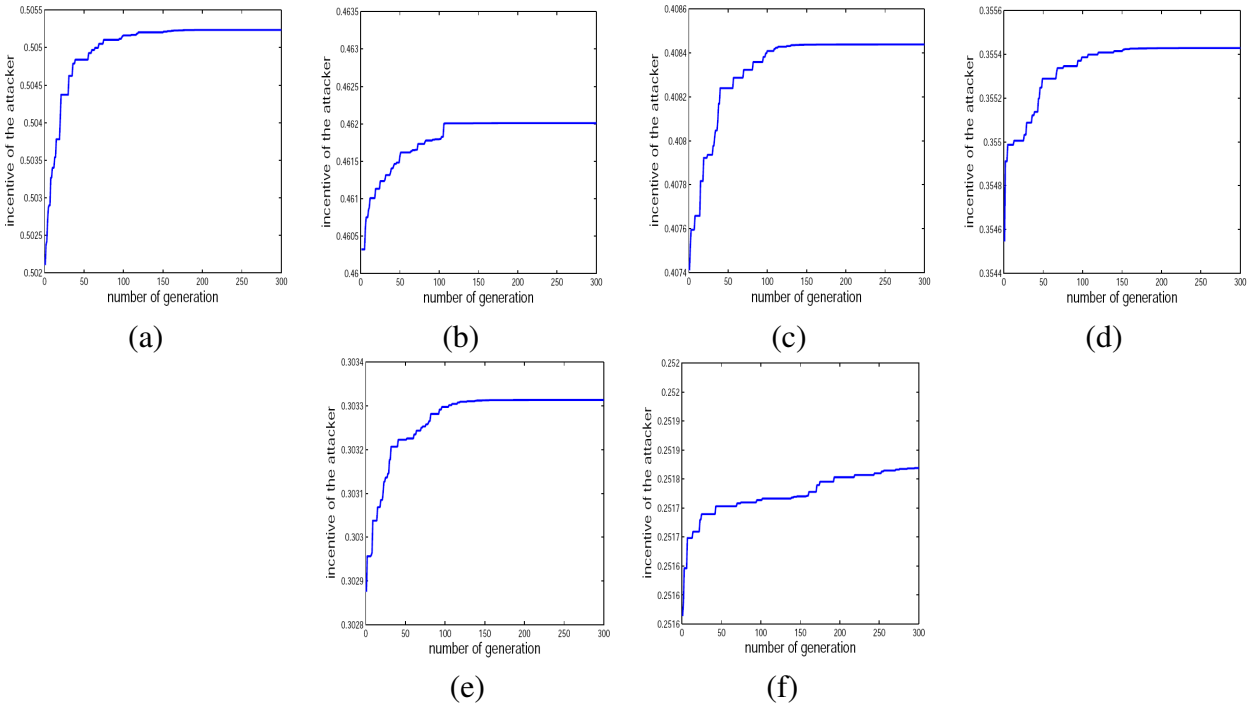

Figure 6. The incentives of BWD attackers using the optimized solutions from DE. (a) P_1 with hash power 0.49, (b) P_2 with hash power 0.45, (c) P3 with hash power 0.4, (d) P4 with hash power 0.35, (e) P5 with hash power 0.30, (f) P6 with hash power 0.25.

The simulated results demonstrated that with the optimized strategies the attackers can acquire non-ignorable excessive incentives by BWD attacks. In the next section we will discuss the countermeasures against BWD attacks.

5. Discussion on defenses

As the BWD attackers pretend to be honest miners who always submit full proof of work to the pool manager, it is hard to judge whether a miner is an attacker or not. Because, even if a miner delayed the submission of a valid block, we cannot directly mark him as a rogue miner, since the delay can be attributed to deliberately attacks, network congestions or limited hash powers. By now, there has

been no effective defense against this type of attack. Thus, BWD may be popularly used and becomes disastrous for any open mining pool that allows untrusted miners to join pool anytime.

In this section we will compare some existing defense strategy of similar block withhold attacking, and discuss about how to defeat a possible BWD attacker in a pool.

5.1. Desired properties of defense strategy

A defense to become immediately practical on the existing bitcoin network, it should be non-intrusive, i.e., should require no incompatibility with the existing bitcoin protocol. Luu [22] proposed the desired properties for the pools and defense strategies to determine whether a fix is adequate and practical.

- P_1. It does not favor either big or small player, and should treat them equally as long as they are honest.
- P_2. It disincentives both pool operator and player to drop (or delay submitting) valid blocks.
- P_3. It preserves the existing bitcoin block chain.
- P_4. It is compatible with existing mining hardware.
- P_5. It does not affect miners who are not in the pool.
- P_6. It requires a minor bitcoin protocol's change.
- P_7. It does not make the pool violate P_1 or P_2.

Obviously, the current pooled mining protocol does not satisfy P_2, thus making the fix necessary. Besides, the miners have participated a fierce competition to solve the hash hard problems, the resources of computation and transmission band should be carefully preserved, thus we add an additional required property:

- P_8. it does not require too much resources of miners and pools.

5.2. Related works of countermeasure against block withholding attacking

The problem of withholding valid blocks has aroused attentions of researchers. Some proposed to detect rogue miners using traditional security techniques. Courtois [19] proposed an inspection scheme which records all the mining events and calculates the standard deviation of each miner's contributions, which are compared with their statistical records to find out whether there exist some withholding attackers. Rosenfeld [20] discussed a technique uses Honeypots for luring rogue miners into a trap. However, in these schemes the pool managers and miners have to expend their computing power for doing much more computation wasting their valuable resources. Thus, does not satisfy P_8.

One of the main reasons that make the block withholding delay attack profitable is that every power share has the same value from the miner' perspective. Thus, Schrijvers [29] and Bag [30] proposed to counter BWH attack by making the reward strategy incentive compatible. In their measures, in order to well-incentivize the miners to submit blocks, the finders of the valid blocks can have extra reward in addition to the normal gains proportional to their computing powers. Although this technique satisfies the above properties P_3 to P_6, it suffers from several drawbacks, especially P_1 breaks down: normal shares are worth significantly lesser. Thus, compared with present experience in pools, variance increases for all pool participants and especially for smaller ones [22].

The source of block withholding is that the miners can check whether they have found a full solution or a partial proof of work. Some researchers proposed fixes of the bitcoin proof-of-work (PoW)

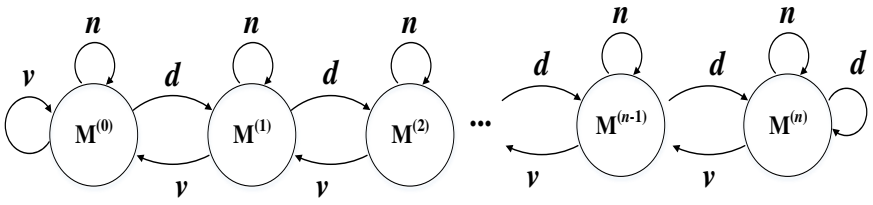

Figure 7. Sate machine of the dynamical payoff scheme.

protocol [22] [24]. The general idea of these approaches is to not allow miners to recognize which solution is a valid block, thus preventing dishonest miners from withholding blocks. For example, in the scheme of [24], each pool privately selects two binary strings r and s, then the pool manager broadcast the hash value $p = Hash(r||s)$ to the pool members, after that:

- Each miner in the pool try to find a block that includes p as a new field of the header of the block, and the hash value of the bock must starts with z' zero bits.
- The pool receives pool shares (partial proofs) from its members and computes hashes until it finds a block with a hash value $0^{z'}||r||0, 1*$. If the pool finds such a block, it broadcasts the block along with $r||s$.
- After a block is sent by a mining pool along with the combined nonce $r||s$, every bitcoin node checks if the hash of the pad ($Hash(r||s)$) is included in the block and that the first $z - z'$ bits of the combined nonce are same as r.

Since every partial work needs to have z' leading zero bits, no bitcoin miner can distinguish between a block with a partial PoW and the same with a full PoW. Thus the miners cannot initiate block withholding attack against the pool. Also, since the pool has to include p in the block header, there is no way it can diminish the amount of computation needed to solve the puzzle. However, the schemes required modification of the existing block structure and PoW protocols, which violates property P_6 and hard to be applied in the existing bitcoin system.

5.3. Change the payoff scheme of pools

As the purpose of BWD attackers is to obtain higher revenues. From this point, we proposed to change the payoff scheme of pools to increase the risk of deliberately delay the submission of blocks, such that there is a high probability of the BWD attackers to earn less compared with mining honestly. It should be noted that each pool can implement its private payoff scheme, which is accepted by the pool members to distribute gains among them. Thus the fix does not affect the public PoW or other mining rules and satisfies the above property of P_6.

Currently, the pool members should obtain rewards proportional to their contributions (or computing powers) to the pools [1]. However, from Theorem 3 we can see that the attacker A can get excessive gains through BWD attacks, and its total revenue is higher than the computing power α. Therefore, in order to disincentive the attackers to launch BWD attacks, we proposed to modify the pool payoff schemes in a way such that the pool managers record the submission delay events. And the miners' reward is dynamically adjusted with respect to their delay times, so as to decrease the BWD attackers' incentives.

As Figure 7 shows, the dynamical pay off scheme is described by a state machine, where $M^{(i)}$

represents a miner M has delayed i blocks, and v is the input that M successfully submit a valid block, n represents M failed to find a block in a mining cycle, while d means that M submitted a block after other miners and the pool earn nothing by the block.

Nevertheless, the delay of the block submission can attribute to many reasons, especially the uncertain and dynamic congestion of the network or poor calculation power. Therefore, an intellectual approach is required to deal with the complexity and uncertainness of the block delay cases. Fortunately, Fuzzy logic control (FLC) can provide such an intellectual approach, which is capable of mimicking human thought and dealing the uncertainness in the form of fuzzy calculations. It is commonly known that, using three-dimensional membership the type-2 fuzzy sets can efficiently deal with large uncertainties. In the paper, we proposed a fuzzy based payoff scheme to disincentive the attackers.

Taken the following attributes into consideration: x_1:nubmer of delays , x_2:how long for the delay, ..., x_n:the recent contribution of the miner, we can construct an interval type-2 Takagi-Sugeno-Kang (TSK) FLS. An interval type-2 TSK FLS is also described by fuzzy *If-Then* rules that denote input-output relations of a system. In general, a first-order interval type-2 TSK [31] models with a rule base of M rules, each having n antecedents, the kth rule can be expressed as follows:

$$\begin{aligned} &R^k : IF\ x_1\ is\ \tilde{A}\ and ... x_i\ is\ \tilde{A}_i^k\ and\ x_n\ is\ \tilde{A}_n^k \\ &THEN\ y^k\ is\ \tilde{G}_l = \sum_{i=1}^{n} C_{k,i}x_i + C_{k,0} \end{aligned} \tag{5.1}$$

where k=1,...,M, $C_{k,i}$ $(i = 0, 1, ..., n)$ are consequent interval type-1 fuzzy sets; $\tilde{G}_l$ is the output of the kth rule, it is also an interval type-1 fuzzy set; and $\tilde{A}_i^k$ $i = 1, ...n$ are interval type-2 antecedent fuzzy sets. Furthermore, the corresponding membership functions (MF) of kth input variable and output variable can be expressed by the way of $\mu_{\tilde{F}_i^k}(x_i)$ and $\mu_{\tilde{G}^k}(y)$. These rules simultaneously take into account the uncertainty about antecedent membership functions and consequent parameter values.

In an interval type-2 TSK FLS with meet $\sqcap$ under product or minimum t-norm [31], the firing set of the kth rule $F^k(x)$ is defined as

$$F^k(x) = [\underline{f}^k(x), \overline{f}^k(x)], \tag{5.2}$$

where

$$\begin{aligned} \underline{f}^k(x) &= \sqcap_{i=1}^{n} [\underline{\mu}_{\tilde{A}_i^k}(x_i) \sqcap \underline{\mu}_{\tilde{F}_i^k}(x_i)] \\ \overline{f}^k(x) &= \sqcap_{i=1}^{n} [\overline{\mu}_{\tilde{A}_i^k}(x_i) \sqcap \overline{\mu}_{\tilde{F}_i^k}(x_i)] \end{aligned} \tag{5.3}$$

$\mu_{\tilde{A}_i^k}$ is the membership function of fuzzification. Accordingly, the inference result of the kth fired rule is simplified as:

$$\mu_{\tilde{B}^k}(y) = \mu_{\tilde{G}^k}(y) \sqcap F^k(x) \tag{5.4}$$

Finally, the inference results of the all fired rules are described as:

$$\mu_{\tilde{B}}(y) = \sqcup_{i=1}^{M} \mu_{\tilde{B}_i}(y) \tag{5.5}$$

After the fuzzy inference process, the output from an interval type-2 fuzzy set is required. Type-reducer is employed to convert interval type-2 fuzzy set into type-1 fuzzy set output, which is then

converted in a numeric output through running the defuzzifier. For the case of our payoff FLS, center of sets (cos) type reduction is employed as follows

$$\begin{aligned} Y_{cos} &= [y_l, y_r], \ where \\ y_l &= \sum_{i=1}^{M} \frac{\underline{f}^k(x)^i y^i}{\sum_{i=1}^{M} \underline{f}^k(x)^i}, \\ y_r &= \sum_{i=1}^{M} \frac{\overline{f}^k(x)^i y^i}{\sum_{i=1}^{M} \overline{f}^k(x)^i} \end{aligned} \tag{5.6}$$

The output interval of the type-2 FLC system can be calculated based on the values of y_l and y_r. After type reduction, we can get an interval set Y_{cos}, which is applied to defuzzifier to calculate the crisp output:

$$f(x) = (y_l + y_r)/2 \tag{5.7}$$

Let p and α represent the hash power of a pool and M respectively, Δ denotes the total reward of the pool. As the submission delay caused by mining competitions or network congestions is typically a small probability event, if there have been x blocks are delayed by M, its incentives from the pool should be decreased to $\Delta * (\alpha/p)^{(C \cdot f(x))}$, $f(x)$ is calculated by (5.7), $C \in [1, \infty)$ is a penalty factor. Obviously, as the increasing of the delay times i, the gain of M will decrease dramatically. On the contrary, if M successfully submitted a valuable block in state $M^{(x)}$, then it will jump to state $M^{(x-1)}$ and get higher reward.

Theorem 4. *With the new payoff scheme, the BWD attacker will get less revenues than the honest strategy if it continuously delay blocks.*

Proof. Based on Theorem 3, it is easy to see that the revenue of a BWD attacker A who successively delayed x blocks is

$$\begin{aligned} \frac{\alpha(1-\beta)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}} + \sum_{i=1}^{n}(\Delta_i \cdot (\frac{\alpha\beta\gamma_i}{p_i + \alpha\beta\gamma_i})^{C \cdot f(x_i)}), \\ \sum x_i = x \end{aligned} \tag{5.8}$$

Obviously, the first part $\frac{\alpha(1-\beta)}{1-\alpha\beta\sum_{i=1}^{n}\gamma_i P_{wi}}$ is less than the gain α by honest strategy; and we can specifically set the penalty factor C to make sure that the second part $\sum_{i=1}^{n}(\Delta_i \cdot (\frac{\alpha\beta\gamma_i}{p_i+\alpha\beta\gamma_i})^{(C \cdot f(x_i))}$ will quickly decrease to zero as the increases of the delay times. □

Therefore, the new payoff scheme raised the risk of BWD attackers such that they may earn much less rewards than honest strategy. Furthermore, our method does not require modifications of the public protocol of the current bitcoin network, and it can be applicable to the pools as their private payoff schemes of incentives to discourage the rogue miners.

6. Conclusion

In this paper, we investigated the utility of a new type of bitcoin attack called block withholding delay (BWD). BWD leverages the vulnerabilities of current consensus protocol of bitcoins, and the attackers try to delay the submissions of valid blocks to the pool managers, which decreases the victims'

probability of winning monetary rewards from the bitcoin network. On the same time, the attackers regularly mining on their private pools. Thereby they can get undue profits at the cost of undermining the overall earnings of all the honest miners in the victim pools. The optimum splitting strategy of computing powers was quantitatively analyzed for the BWD attacker, and we proposed the conditions under which BWD is well-incentivized. BWD is a real threat to the viability of pooled mining with existing protocols in cryptocurrencies. It should be noted that the BWD attack may also be carried out in other blockchain platforms, as long as the pool's administrators can not check whether the miners' real contributions match their reward. We also discussed the countermeasures against BWD attacks based on the tracking of pool members' delay behavior using type 2 fuzzy inference system, and proposed a new payoff scheme of pools to prevent the BWD attacks by increasing the risk of earning less incentives if the blocks are not immediately submitted. The scheme does not require modifications of the public protocols of the existing bitcoin networks, which makes it applicable to the practical pools.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant nos. 2016YFB0800605 and 2016YFB0800604) and Natural Science Foundation of China (Grant nos.61872255, U1736212 and 61572334).

Conflict of interest

The authors declares no conflict of interest.

References

1. S. Nakamoto, *bitcoin: A peer-to-peer electronic cash systemTech*, **2008**(2008). Available from: https://bitcoin.org/bitcoin.pdf.
2. M. K. M. Ly, Coining bitcoin's Legal-Bits: Examining the regulatory framework for bitcoin and virtual Currencies, *Harv. J. L. Tech.*, **27** (2014), 587–608.
3. A. Zohar, bitcoin: Under the Hood, *Communicat. ACM*, **58** (2015), 104–113.
4. V. Savvas, P. Perikles, R. Maria, bitcoin value analysis based on cross-correlations, *J. Int. Bank. Commerce*, **22** (2017), 1–12.
5. S. Higgins. *$300 Billion: Bitcoin price boosts crypto market value to record high*, 2018. Available from: https://www.coindesk.com/.
6. M. A. Khan and K. Salah, IoT security: Review, blockchain solutions, and open challenges, *Future Generat. Comput. Syst.*, **82** (2018), 395–411.
7. A. Suliman, Z. Husain, M. Abououf, et al. , Monetization of IoT Data using Smart Contracts, *IET Networks*, **8** (2019), 32–37.
8. M. Conti, C. Lal and S. Ruj, *A Survey on Security and Privacy Issues of bitcoin.*, 2017. Available from: https://arxiv.org/abs/1706.00916.
9. Y. X. WANG and J. T. Gao, A Regulation Scheme Based on the CiphertextPolicy Hierarchical Attribute-Based Encryption in Bitcoin System, *IEEE Access*, **6** (2018), 16267–16278.

10. A. Miller, A. Kosba, J. Katz, et al., Nonoutsourceable Scratch-Off Puzzles to Discourage bitcoin Mining Coalitions, *Proc. 22nd ACM SIGSAC Conf. Comput. Comm. Secur. (CCS)*, Denver, Colorado, USA, October, **2015** (2015), 680–691.
11. G. O. Karame, E. Androulaki, M. Roeschlin, et al., Misbehavior in bitcoin: A study of double-spending and accountability, *ACM T. Ioform. Syst. SE.*, **18** (2015), 1–32.
12. G. O. Karame, E. Androulaki, M. Roeschlin, et al., Misbehavior in Bitcoin: A Study of Double-Spending and Accountability *ACM Transact. Inform. Syst. Secur.*, **18** (2015), 1–32 .
13. Khaled Salah, M. H. U. Rehman, N. Nizamuddin, Blockchain for AI: Review and open research challenges, *IEEE Access*, **7** (2019), 10127–10148.
14. I. Eyal and E. G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, *Proc. Int. Conf. Fina. Cryptol. Secur.*, Christ Church, Barbados, March, **2014** (2014), 436–454.
15. H. R. HASAN and K. Salah, Combating deepfake videos using blockchain and smart contracts, *IEEE Access*, **7** (2019), 41596–41606.
16. B. Johnson, A. Laszka, J. Grossklags, et al., Game-Theoretic analysis of DDoS attacks against bitcoin mining pools, *Proc. Int. Conf. Fina. Cryptol. Secur.*, Christ Church, Barbados, March, **2014** (2014), 72–86.
17. E. Heilman, A. Kendler, A. Zohar, et al., Eclipse attacks on bitcoin's peer-to-peer network, *Porc. 24th USENIX Symp. Secur.*, Washington, D.C., USA, **2015** (2015), 129–144.
18. K. Nayak, S. Kumar, A. Miller, et al., Stubborn mining: Generalizing selfish mining and combining with an eclipse attack, *Proc. IEEE Euro. Symp. Secur* Privacy (Euro S&P), Saarbrucken, Germany, March, **2016** (2016), 1–16.
19. N. T. Courtois and L. Bahack, *On subversive miner strategies and block withholding attack in bitcoin digital currency*, 2014. Available from: `https://arxiv.org/abs/1402.1718.`
20. M. Rosenfeld, *Analysis of bitcoin pooled mining reward systems. Accessed: Nov.30, 2018*, 2011. Available from: `https://arxiv.org/abs/1112.4980.`
21. D. Wu, X. D. Liu, X. B. Yan, et al., Equilibrium analysis of bitcoin block withholding attack: A generalized model, *Reliabil. Eng. Syst. Safety*, **5** (2019), 1–32.
22. L. Luu, R. Saha, I. Parameshwaran, et al., On power splitting games in distributed computation: The case of bitcoin pooled mining, *Proc. IEEE Symp. 28th Comput. Secur. Found (CSF)*, Verona, Italy, July, **2015** (2015), 1–18.
23. I. Eyal, The miner's dilemma, *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, July, **2015** (2015), 89–103.
24. S. Bag, S. Ruj and K. Sakurai, Bitcoin block withholding attack: Analysis and mitigation, *IEEE Trans. Inf. Forensic Secur.*, **12** (2017), 1967–1978.
25. N. T. Courtois and L. Bahack, *On subversive miner strategies and block withholding attack in bitcoin digital currency*, 2014. Available from: `Available:https://arxiv.org/abs/1402.1718.`
26. A. Laszka, B. Johnson and J. Grossklags, When bitcoin mining pools run dry, *Proc. Int. Conf. Fina. Cryptol Secur.*, San Juan, Puerto Rico, January, **2015** (2015), 63–77.

27. S. Shen, H. Li, R. Han, et al., Differential game-based strategies for preventing malware propagation in wireless sensor networks, *IEEE Trans. Inf. Forensic Secur.*, **9** (2014), 1962–1973.

28. J. Tuwiner, *Bitcoin Mining Pools*, 2018. Available from: `https://www.buybitcoinworldwide.com`.

29. O. Schrijvers, J. Bonneau, D. Boneh, et al., Incentive compatibility of bitcoin mining pool reward functions, *Proc. Int. Conf. Fina. Cryptol.Secur.*, Christ Church, Barbados, **2016** (2016), 477–498.

30. S. Bag and K. Sakurai, Yet another note on block withholding attack on bitcoin mining pools, *Proc. Int. Conf. Inf. Secur.(ISC)*, Honolulu, HI, USA, **2016** (2016), 167–180.

31. J. S. R. Jangn and C. T. Sun, Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence, *Prentice-Hall Inc.*, Upper Saddle River, NJ, **1996** (1996).

32. M. Mizumoto and K. Tanaka, Some properties of fuzzy sets of type-2, *Inform. Control*, **31** (1976), 312–340.

AIMS Press