Discouragement of Block Withholding Attack in Bitcoin

Zhaojin Li

March 2022

Abstract

Although blockchain, especially Bitcion, has been widely adopted, their fundamental reliability, safety and security is still an open challenge. One key assumption in Bitcoin is the incentive engineering, i.e. all participants would behave honestly. Similar assumptions can be seen in other applications. However, no formal proof or analysis can be found, due to the complex of the blockchain systems. This work focuses on the analysis of one type of incentive attack called the Block Withholding Attack (BWH). Since BWH was proposed, it has been demonstrated by several papers [10][11] to be theoretically profitable to launch in the Bitcoin blockchain. In practice, however, such attacks haven't been observed by the public. This paper presents innovative research to study this phenomenon. We reveal the factors (such as the modification of the puzzle and the expected mining time changes) that impact the implementation of BWH, and measure the impact through theoretical analysis and simulation. We model this problem using game theory and build several strategies among mining pools. We find that the conditions for the attacker to be profitable are nearly impossible in reality. The theories, techniques and frameworks established through this project can be applicable to various analyses in related areas.

1 Introduction

Bitcoin is the first popular decentralized system, and also has been one the largest cryptocurrency platforms [7], its estimated capitalization is over 1 trillion. Bitcoin's incentive is from a process called block mining, anybody can join it with some computation power, participants who are called miners try to solve the hash problems named puzzle with a certain difficulty to upload the block to the ledger, this mechanism is called Proof of Work,which usually needs a large amount of computation to complete it. Once a miner is successful to upload a block, he will be rewarded with a certain extent of Bitcoins, and it's currently 6.25 bitcoins. All other miners will validate the new block, and update it to their local data if the validation passes, then restart their mining to the new block.

When a transaction is created on the chain, it will be sent to the transaction pool, and wait to be picked by the miners. Miners will select many transactions and combine them into a block, and the transactions with higher transaction fees are always in higher priority to be picked. A block contains transactions and a block head, which consists of some parameters, including a hash pointing to the previous block, a timestamp, miner, block size, and the nounce. The aim of PoW is to find a nuonce that can make the hash value of the whole block is less than a certain number. If and only if it completes, this block can be certified as the valid block. Meanwhile, the difficulty of the puzzle will be adjusted regularly, per 2016 blocks, to ensure the mining time is always around 10 minutes.

The rule of the chain is that only the longest chain is the valid chain.

With the difficulty of the puzzle becomes much higher which leads the single miner to wait for a very long time to mine a block, a kind of organization called mining pools appear, all the members of one mining pool compute the puzzle for a same block and share the reward if mining successfully. Usually, a mining pool consists of a pool manager and many member miners, the pool manager doesn't compute the nounce but hang out the mission to member miners, and miners return their results to the manager. In order to decide how to arrange the rewards to each miner, the manager need to measure the amount of work from each miner, and a mechanism the partial proof of work is introduced.

As mentioned above, the usable nounce makes the hash value of the whole block is less than a certain number, it's also called full proof of work under this mechanism. The partial proof of work can not be used practically on block mining, but can be used to measure how many works that miners have spent on the mining. When miners compute for the nounce, they return not only the full proof of work, but also the partial proof of work. The partial proof of work is much easier than the former, usually 1,000 times easier to find. Every time a block is mined, the rewards will be distributed to miners according to their amount of submitted partial proof of work.

This mechanism of rewards allocation can be utilized by other malicious pools with the Block Withholding attack, where a malicious pool manager can allocate its own miners into the targeted mining pool, and allocated miners only submit the partial proof of work but hide the full proof of work. In such this process, the attacker can share the revenue from the targeted pool. Sometimes the attacker benefits from it, but it's also possible to receive a deficit.

Then the arrangement of each Section.

Contributions. In summary, this work makes the following contributions

- We formulate the BWH attack problem as a concurrent game with partial information
- We introduce the parameter of time which is never be considered in previous works

- We set up a model to simulate and find the condition necessary and sufficient conditions in this game to be profitable
- We prove the condition above is nearly impossible in reality, to prove that the BWH attack is actually unprofitable.

Organisation. The remaining par of this paper is organised as follows. Section 2 discussed the previous related works. Following, the problem formulation and the modelling are discussed in the Section 3 and Section 5 respectively. Then in Section 6 we analyzed and summarized the collected data. As for the Section 7, we supposed some interesting tips from the data in Section 6 for discussion. Finally, We presented the Conclusion in Section 8 and also the future works.

2 Related work

Previously, many papers have researched the BWH attack from many aspects. [6]has made a survey of how the game theory can be introduced into the blockchain on many scenery, [10] supposed that pool managers prefer to release the BWH and promot the cooperation between miners, [9] introduced the evolutionarily stable strategy to analyze the equilibrium, [11] supposed an advanced algorithm based on it to improve it's revenue, some calculated the different revenue rates under different situations, [3] tried to discourage the BWH attack, [4] analyze how to choose pools when BWH happens from the sights of miners instead of mining pools, [8] defined it as a non-cooperative game and modeled the competition on a multiple-players model and provide the equilibrium according to the various power ratio, [1] researched the Nash Equilibrium of it under several conditions, [13] introduced the reputation score into the blockchain system to encourage the honest mining, [12] analyze the security threats of it, and some supposed the method to self-detect if the mining pool is under the BWH attack, [5] supposed a method to efficiently detect the BWH attack.

[2] supposed the miners' dilemma from the prisoners' dilemma in the game theory, he proved that the cooperation between miners is not the equilibrium because miners always prefer to attack others to get more profits, while if every miner chooses to attack, everyone doesn't get more when equilibrium and the whole stability of the system decreases. The limitation above is that this paper only considered the two-player model, while in reality there are many mining pools in the Bitcoin system. And also, instead of to counterattack the attackers, attacked pools practically do not know the identities of attackers. In addition, all the previous research don't consider the absolute time parameter, but only consider the round. The round means each period when a block is mined successfully by a miner. However, the time of each round is not static, it changes with the current total effective computation power in the system and the current difficulty of the puzzle. If the power increases, the roundly time

will decrease before the puzzle adjusts itself at the beginning of the next epoch (2016 rounds per epoch), vice versa.

When previous works calculate the revenue, they always use the roundly revenue to judge whether it's profitable or not. Let's say an extreme situation that a mining pool can get the whole rewards from the current round while this round lasts one year to end, that's undoubtedly unacceptable in reality. Therefore, comparing with the roundly revenue, the unit timely revenue can be more practical to explore. Further, we can explore more such as if there are suitable conditions and strategies for pools to get more profits based on the unit time instead of the round.

3 Problem formulation

Firstly, we need to compare the differences between non-time-considered model and time-considered model. We want to find whether the time parameter impacts the strategies and results and how to make the quantification. So, we make some basic setting below.

First, according to the characteristics of Bitcoin, we assume this issue as a **concurrent game with partial information**, because all the miners (players) in Bitcoin behave at the same time and they can only know the information about them selves and some global information, but they can not clearly know other miners' inner information. (some methods can be used to generally detect others, while we don't discuss them here.) Then, we set the final target which is to get the most profits for all the miners.

Also, we assume that both the mining pools have the method to detect if they are in attacks and how much computation power is in the attacks, but they can not know who is(are) the attack(s). They may make some guess by analyzing the global information change, while we don't consider this here.

As for the pools to against the attacks, they would allocate a part of computation power to adopt the detection method to inspect whether they are attacked, such as the firewall application. We assume there are two states in each pool, the normal state and the sensitive state. When the pools are in the normal state, they spend a little (nearly 0) computation power in detection, so they can put nearly all of the power into the puzzle solving to make more profits, while the drawback of this state is that it needs more time to detect the BWH attack; In comparison, if they are in the sensitive state, they can quickly inspect the attack with the more power cost. At present, to be simple, we assume that both the power cost of the two states are 0.

4 Formalisation of the game

Following the definition of concurrent games in [], we define the game formally as a tuple of x elements. The instantiation of the elements models a global state of the game. A game with n players, $\langle S, I, A, T, R, ... \rangle$

Parameters	Description
P	mining pools(players) (p_1,p_2,p_3...p_n)
C	computation power (c_1, c_2, c_3, ...c_n)
K	counterattack strategies (s_1, s_2, s_3)
S	states (normal or sensitive)
t	reaction time of pools in normal state
Pr	profit in reality
Pe	profit in original expectation
R	roundly rewards (r_1, r_2, r_3...r_n)
AM	attack matrix for all the pools, [n * n]

Table 1: Parameters description

power ratio	1:9	2:8	3: 7	4:6	5:5	6:4	7:3	8:2	9:1
roundly model	32%	32.5%	33%	43%	33%	33%	31%	27%	23%
timely model	0	0	0	0	0	0	0	0	0

Table 2: attack allocation proportion of the attacker

where $S = \langle s_1, \ldots, s_n \rangle$ $s_i \in \{A, H, V\}$ A stands for attacking others, V stands for that the node is under attack but did not realised that it is being attacked, H stands for mining without attack.

$A = \langle a_1, \ldots, a_n \rangle$, $a_i = \{C, \}$ where C stands for being counter-attacked, stands for stop attacking, ...

$T : SxA \rightarrow S|$....

$R = \langle r_1, \ldots, r_n \rangle$. The calculation of r_i is defined by a attack metrics. ...

5 Modelling

As mentioned in last section, there are two parts in this section. The first part is comparison between the time-considered model and non-time-considered model, and the second part is to find how BWH attacks can be profitable in the time-considered model.

In the first part, our model is set up based on the miners' dilemma theory in [2], where there are only two mining pools so they can easily ensure the attacker. From the concept of [2], the game is a complete information game, and the logic of action is that the mining pool will launch the attack if it thinks it's profitable in its calculation. Therefore, in the non-time-considered model, the roundly revenue increases when attacking. However, in time-considered model, we introduce the time parameter and the rest remain unchanged, the pseudo code is below:

From the result from the Table [2], we see that for all the power ratio, pools choose to attack in the roundly model while don't attack in the timely model. The actions are completely opposite with the same strategy in two different models, because the mining time for each round will increase when the

Algorithm 1 Simulation for two models

$S \leftarrow sensitive$
P $\leftarrow pools$
K $\leftarrow$ directly counterattack
initial non-time-AM and time-AM
while non-time-AM not converges **do**
    p_1 non-time-attack p_2
    p_2 non-time-attack p_1
end while
while time-AM not converges **do**
    p_1 time-attack p_2
    p_2 time-attack p_1
end while
output the non-time-AM and time-AM

Algorithm 2 Calculation for allocation in the BWH attack

$S \leftarrow sensitive$
P $\leftarrow pools$
best allocation $\leftarrow 0$
best revenue $\leftarrow$ current revenue
for allocation from 0 to c_1 **do**
    current revenue $\leftarrow$ the revenue for this allocation * roundly time rate
    if current revenue $>$ best revenue **then**
        best allocation $\leftarrow$ allocation
        best revenue $\leftarrow$ current revenue
    end if
end for
return best allocation and best revenue

BWH attack happens, which has thinned the revenue of the unit time for both the attacker and the victim, and even made the attacker's unit revenue lower than the previous, so they would not choose to attack. Then we can see, with considering the time and unit revenue, this model is not the miners' dilemma any more, but the lose-lose game. So, in this model, no attack is the Nash Equilibrium, any attack will reduce the revenues of both sides.

However, only the analysis above is not enough to prove that the BWH attack is non-profitable, because if consider that if the attacker keeps attacking and the victim does not realize that it's attacked, in the next epoch, the puzzle difficulty will be adjusted to fit the currently decreased effective computation power, and the mining time will accordingly decrease, so the attacker gets benefits from the concentrated unit revenue since the next epoch. Therefore, we still need to explore in what conditions the attacker can benefit and whether the conditions are practical in reality.

Intuitively, we know from the model above that a mining pool can benefit in attacks if the victim does not react (or reacts too slow), and lose if it's counterattacked. So we would like to find the bound of the reaction time that attackers can earn more if the victim's reaction time is longer than it, and get loss if the victim's reaction time is less than it.

From the results above, we know that the attack loses profits in the first attacking epoch before the puzzle difficulty adjusts itself regularly (per 2016 rounds), and benefits from the second epoch if the victim pool does not counterattack. So, each round we would calculate both the expected revenue without attacks and the practical revenue, and we assume a trigger when the attacker is counterattacked within the first epoch, or under the certain amount of counterattack after the first epoch which causes the practical revenue less than the expected revenue. We set this trigger as the end of this attack, and we then compare the total expected revenue and the practical revenue, to determine whether this attack is profitable.

Then we consider three aspects to analyze and set up the models, they are:

- Number of mining pools in this model?
- How to counterattack when pools realize they are under BWH attacks?
- If stopping to counterattack when the attacker stops?

Firstly, in terms of the number of pools, we consider the classic two-players model and the multiple-players model. In the former, the victim can easily know the attacker as there are only two pools, while in the latter, there is no way to ensure who is the attacker.

Secondly, as for how to counterattack, there are three strategies:

1. No counterattack, similar to the victim does not realize it's under the BWH attack, mainly used in the two-players model
2. Because don't know the identity of the attacker, the victim just consider all the other pools as a whole and attack back, aiming to make more

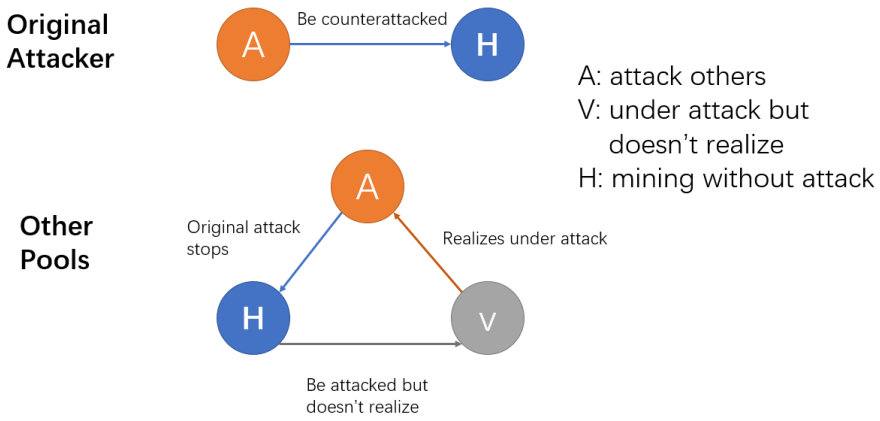

Figure 1: How pools change states

power ratio	24.3	14.3	12.8	11.7	10.7	9.6	4.5	4.5	3.0	2.1
margin	3071	2048	767	512	436	357	290	290	205	191

Table 3: Least profitable reaction time

revenue instead of counterattacking the attacker. And the allocation of the attack power is proportional according to the targets' computation power.

3. With the same reason that to get more profits as above, the victim choose a target which can get the most revenue to attack.

   Fig 1 is how the state changes in the model:

As for the third aspect, if the victims are rational as normal as ideal players in the game, the counterattack is stoppable, and the victims would stop counterattacking when they find the attacker has stopped attacks; Otherwise, the counterattacks are not stoppable.

Based on the setting above, we analyze and find the margin in each combination in the next section.

6 Analysis

According to the algorithms [3] and [4], we pick up the top 10 mining pools in Bitcoin at the date 2022/04/25, which is shown in :

And we select the results:

We just analyze the three aspects: number of pools, how to counterattack, and whether stopping counterattack.

Algorithm 3 Reaction margin finding

```
reaction time ← random r
profit ← Revenue calculation (r)
while abs(profit) > convergence do
    if profit > 0 then
        decrease r
    else
        increase r
    end if
    profit ← Revenue calculation (r)
end while
return r
```

Algorithm 4 Revenue calculation

```
S ← normal
P ← pools
K ← directly counterattack
initial AM
p_1 randomly attack a pool
time ← 0
reaction time ← r
while p_1 is not under counterattack do
    for pool i in P do
        if S_i is under attack and reaction time passes then
            P_i launches attack
        end if
    end for
    calculate the roundly balances, update AM and pools information
    for pool i in P do
        Pr_i ← pr_i + Pr_i (roundly revenue)
        Pe_i ← pe_i + Pe_i (pe * roundly time)
    end for
    time ← roundly time + time
end while
return (Pr_1 - Pe_1)
```

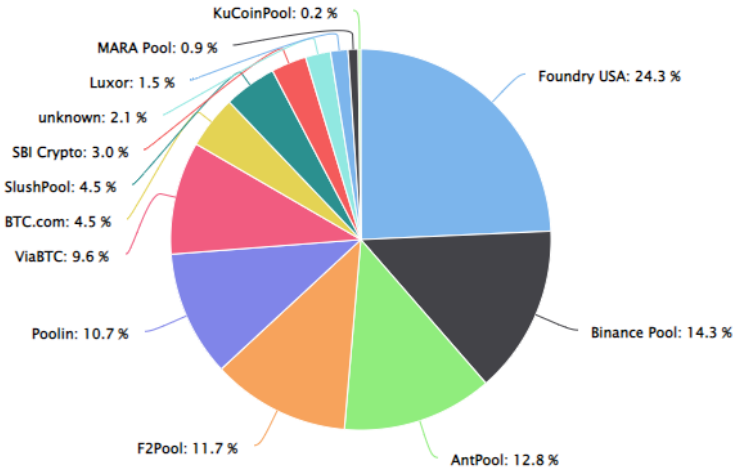

Figure 2: Bitcoin power distribution

1. Whether stopping counterattack when the attacker stops

   we can image that if the victims never stop counterattack, the attack would also not stop attacking, because if it does so the situation is like that it's the victim, and it would lose more. This means that there is no end of attacking, and every pool takes part in attacks until convergence. And we know from the table 1 that only a pool holds more than half of power it can make profits in convergence, otherwise the revenue is nearly the same as no attacks. And in reality, no single pool holds more than a half of computation power. Therefore, if pools don't stop counterattacking when convergence, no pool can gain more, while the total stability of the Bitcoin decreases due to the fall of effective computation power.

2. number of pools

   Here we mainly consider the multiple-players model because it's more practical.

3. how to counterattack

   Because the victim can not know who are the attackers, we ignore the directly counterattack but focus on the ranged-attack and targeted-attack.

   As mentioned above, the former means the victim attack all the other pools according to their computation power, not for counterattacking, but for making profits. With this strategy, the attacker and all the other pools would be attacked at the first reaction round, then all the pools would act at the third reaction round, which would lead to the fast convergence. And as we know from above, a pool can not be profitable in the BWH

counterattack after convergence if the computation power of this pool is less than 50%.

In terms of the latter, the targeted-attack, comparing with the ranged-attack, this model can has more time before it is counterattacked, because much time is needed to form the attack chain. In each reaction round sensitive pools just choose a target to attack where attacking that target can bring the most profit in their calculations. From the results of Table[3] we can see the least reaction time counted by round for every pool, and the pool data is from the instant distribution of computation power of the top 10 pools of Bitcoin shown in the figure[2].

We can find that even under the most suitable conditions to launch a BWH attack, the least reaction time to be profitable is still more than 200 round for the smallest pool, which is easy to be detected by the method in [5]. And non-negligibly, this result has eliminated other factors such as power fluctuates or miners migrate which may further decrease the revenue.

There is also an interesting finding from the Table[3] that the small-power pools are more preferable to launch a BWH attack because their priorities of been counterattacked are lower, and the large pools are more likely to be counterattacked as attacking them would get more revenues.

7 Discuss

There are several tips that we found interesting during our model setting up and simulation.

1. determination of pool managers We have also analyzed the different composition of mining pools–where the computation power of mining pools belongs to? Pool managers? miners? Or mixed?

   - If all belongs to the pool manager, all the revenue from the BWH attack will be thinned by all the computation power, and the revenue rate is in a low level.
   - Extremely, if all belongs to miners and the manager occupies all the revenue, because the pool manager does not share the revenue (expect the management fee), the revenue rate of the manager would be very high due to the low base of it.

     We simply analyzed under a two-player model with equal power, setting the management fee is 1%. Then we found that the revenue rate for a pool manager of a all-owned power pool is 10%, while the revenue rate in a all-miners pool is 1000%.

   We suppose that the low self-power pool is more preferred to launch a BWH attack because of the much higher revenue rate.

2. loyalty of miners As for the individual miner, if it is allocated to another pools for the BWH attack, its direct revenue decreases because it can only directly share the revenue from the attacked pool, and the rest needs to be supplied by the attacker, by transferring the bitcoin to it. Due to this, it has to undertake more risk for the extra operation, such as the attack may refuse to share the extra revenue, so it is reasonable for the individual miner to ask for more benefits to launch the BWH attack, and the pool manager has to share an extra part from the revenue to the attacking miner. Maybe we can suppose an positively correlated equation between the extra payment from the pool manager and the loyalty rate of attacking miners, the higher extra rewards, the higher loyalty.

8 Conclusion and future work

In this paper, we introduced the time dimension on the BWH attack, based on this, we proved that the BWH attack is not a profitable attack calculated by the unit time revenue. After that, we explored under which situations the BWH attacker can benefit itself, and we found that those situations are unpractical in reality, by picking the real instant data from the Bitcoin system and simulating, to eventually prove that the BWH attack is non-profitable, that is also why we hardly see it happens in the real internet.

There are also some limitations, we ignored many subsidiary factors such as the increase trend of the total computation power in the Bitcoin system, and the cost of detection if under the attack, the management fee of mining pools, and also the discussion in Section 7. We will take them into account in the future work.

References

[1] Wu Di ; Liu Xiang dong ; Yan Xiang-bin ; Peng Rui ; Li Gang. Equilibrium analysis of bitcoin block withholding attack: A generalized model. *Reliability engineering and system safety, 2019-05, Vol.185, p.318-328*, 2019.

[2] Eyal Ittay. The miner's dilemma. *Symposium on Security and Privacy, Vol.2015-, p.89-103*, 2015.

[3] Chen Zhihuai ; Li Bo ; Shan Xiaohan ; Sun Xiaoming ; Zhang Jialin. Discouraging pool block withholding attacks in bitcoin. *Journal of combinatorial optimization, 2021-07-29, Vol.43 (2), p.444-459*, 2021.

[4] Kentaro Fujita ; Masahiro Sasabe ; Shoji Kasahara. Mining pool selection under block withholding attack. *Applied sciences, 2021-01-01, Vol.11 (4), p.1617*, 2021.

[5] Seungjoo Lee, Suhyeon ; Kim. Countering block withholding attack efficiently. In *IEEE Conference on Computer Communications Workshops*, pages p.330–335. IEEE, 2019.

[6] Ping ; Liang Ying-Chang ; Kim Dong In Liu Ziyao ; Luong Nguyen Cong ; Wang Wenbo ; Niyato Dusit ; Wang. A survey on blockchain: A game theoretical perspective. *access, 2019, Vol.7, p.47615-47643*, 2019.

[7] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[8] Altman Eitan ; Menasché Daniel ; Reiffers Alexandre ; Datar Mandar ; Dhamal Swapnil ; Touati Corinne ; El-Azouzi Rachid. Blockchain competition between miners: a game theoretic perspective. *Frontiers in Blockchain, 2019, Vol.2*, 2019.

[9] Kim Seonggeun ; Hahn Sang-Geun. Mining pool manipulation in blockchain network over evolutionary block withholding attack. *IEEE access, 2019-10-03, Vol.7, p.1-1*, 2019.

[10] Cheng Yukun ; Xu Zhiqi ; Yao Shuangliang. The evolutionary equilibrium of block withholding attack. *Journal of systems science and information, 2021-07-17, Vol.9 (3), p.266-279*, 2021.

[11] Chen Yourong ; Chen Hao ; Han Meng ; Liu Banteng ; Chen Qiuxia ; Ren Tiaojuan. A novel computing power allocation algorithm for blockchain system in multiple mining pools under withholding attack. *IEEE access, 2020, Vol.8, p.155630-155644*, 2020.

[12] Feng Shuya ; He Jia ; Cheng Maggie X. Security analysis of block withholding attacks in blockchain. In *ICC 2021 - IEEE International Conference on Communications*, pages p.1–6, 2021-06.

[13] Yu Lianyang ; Yu Jiangshan ; Zolotavkin Yevhen. Game theoretic analysis of reputation approach on block withholding attack. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 2020–12–19, Vol.12570, p.149–166. Cham: Springer International Publishing, 2020.