On the Strategy and Behavior of Bitcoin Mining with N-attackers

ABSTRACT

Selfish mining is a well-known mining attack strategy discovered by Eyal and Sirer in 2014. After that, the attackers' strategy space has been extended by many works. These works only analyze the strategy and behavior of one single attacker. The extension of the strategy space is based on the assumption that there is only one attacker in the blockchain network. However, a proof of work blockchain is likely to have several attackers. The attackers can be independent of other attackers instead of sharing information and attacking the blockchain as a whole. During this problem, we are the team who for the first time analyze the miners' behavior in a proof of work blockchain with several attackers by establishing a new model. Based on our model, we extend the attackers' strategy space by proposing a new strategy set publish-n. Meanwhile, we revisit other attacking strategies such as selfish mining and stubborn mining in our model to explore whether these strategies work or not when there are several attackers. We compare the performance of different strategies through relative stale block rate of the attackers. In a proof of work blockchain model with two attackers, strategy publish-n can beat selfish mining by up to 26.3%.

KEYWORDS

Bitcoin, Mining, Selfish mining, N-attackers

1 INTRODUCTION

Traditional payment on the Internet is based on trusted third parties. The weakness of the trust based model makes completely non-reversible transactions impossible [2] and arises public's interest in decentralized cryptocurrencies based on cryptographic proof. These cryptocurrencies, as represented by Bitcoin, apply the blockchain technology which is a distributed database used to store and maintain a list of records[10]. Although a series of consensus protocol such as proof of stake (POS) and practical Byzantine fault tolerance (PBFT) is also applied to some of these cryptocurrencies, Proof of work powered blockchains take about 90 percent of the market. In a proof of work blockchain, a miner with α fraction of the whole hashpower should only gain α fraction of the total block reward. However, many studies indicate that an attacker can take some strategies to gain extra revenue. Among these strategies, the most well-known one is selfish mining represented by Eyal and Sirer in 2014. Many other strategies such as stubborn mining are the extensions of selfish mining. We can call these strategies selfish mining style strategies.

In Bitcoin, selfish mining style attacks have not shown up yet due to the stable environment of Bitcoin. A statistics from *blockchain.info* indicates that in the past year, the difficulty to find a new block has increased by four times. Up to now, the increase of block reward (in USD) and the increase of difficulty to find a new block are still proportionate.

A proof of work blockchain will have several attackers once the block reward drops to 12.5 Bitcoin per block in the future or the price of Bitcoins drops because of other factors, the crisis may show up and the likelihood for a miner or a mining pool to take tricky strategies increases. Once one attacker shows up, the other miners can either stick to the Bitcoin protocol and lose part of their share of revenue or become another attacker and steal the honest miners'revenue to make up for his loss. The second option is more appealing to a miner. Thus it is necessary to build a new model for a proof of work blockchain and analyze the attackers' behavior and strategy.

We establish a new model of a proof of work blockchain with several attackers and explore the attackers' behaviors and their mining strategies. Existing works about mining attacks [1][4][6][8] put their emphasis on the development of one single attacker's strategy space. As far as we know, the miners' behaviors and strategies in a proof of work powered blockchain with several attackers have not been studied in detail so far. What new action will be made and whether the attacking strategies for a single attacker still work has not been analyzed yet.

Contribution 1: Establishing a new model of a proof of work blockchain. Our model allows the existence of several attackers. The attackers do not share information, and they will have an impact on each other by publishing mew blocks. Their decision-making process is an independent work, but their state transition depends on other miners. A proof of work blockchain model with several attackers is first discussed in our work. Thus new mining behaviors and new mining strategies will be introduced.

Contribution 2: Presenting a new strategy set publish-n. We extend the strategy space of mining attack and propose a strategy named publish-n. Our simulation result turns out that publish-n strategy performs better than other strategies when there are several attackers and the attackers' hashpower is low. This strategy set allows the attacker earn more profits and it even benefits the honest miner sometimes.

Contribution 3: Revisiting of existing strategies. We revisit selfish mining proposed by Eyal and Sirer[1] and stubborn mining proposed by Nayak[8]. Stubborn mining may not be a good option in a blockchain with several attackers while selfish mining still works in most of the situation. Our simulation result even shows that a selfish mining attacker with the hashpower which is not enough to earn extra revenue in a blockchain with n attackers is likely to gain revenue more than his share in a blockchain with n+1 attackers.

The remainder of our paper is organized as follows: We begin by introducing the basic concepts and the attackers' strategy in a proof of work blockchain in Section 2. In Section 3, we introduce

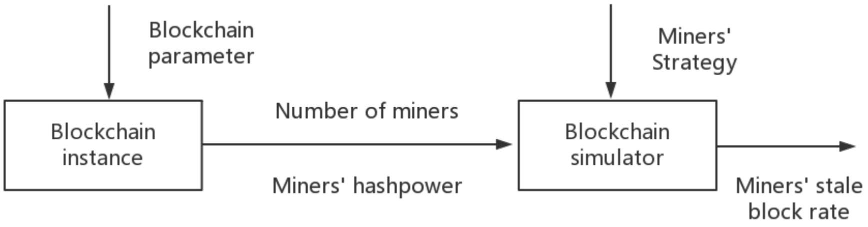


Figure 1: Our blockchain model with two phases

our model and present the attackers' potential state space, action space. In Section 4, we discuss miners' strategy space. In Section 5, we compare the strategies in the strategy space. In Section 6 we conclude our paper.

2 PRELIMINARIES

2.1 Behavior of Attackers and Honest miners

For an honest miner Alice, her action is irrelevant to her state. She obeys the relevant protocols in a proof of work blockchain system so that she reveals a block immediately after she finds it. She always accepts the most extended chain and mines on top. When a fork exists, she works on the chain she received first.

For an attacker Bob, his decision depends on the state and his strategy. Bob aims to waste his opponents' hashpower and gain extra revenue. The most well-known method is to reveal his blocks and publish it according to his state and strategy.

2.2 Mining Attacks

The proof of work consensus protocol of Bitcoin is based on an idealized assumption that the majority of the hashpower is honest. Since Eyal and Sirer defined the behavior of selfish mining in 2014, the reliability of the proof of work consensus protocol has been broken. Selfish mining allows a mining pool to obtain a revenue more massive than its ratio of mining power[1]. An attacker with more than 33 percent hashpower can gain an extra revenue. The threshold can even be lower if the attacker influences the honest miner. Selfish mining wastes the hashpower of the honest miner. Note that, selfish mining is an irrational strategy. The attacker's revenue will also drop in a short term until the difficulty of mining decreased. Several works [4][8] have analyzed that selfish mining strategy is suboptimal.

After Eyal and Sirer's work, many works have analyzed mining attack. Some works such as [4] [8] can be seen as the extension of selfish mining, and some works including [6][7] describe a network-level attack, eclipse attack. Meanwhile, [8] also combines selfish mining and eclipse attack. Among these works,[8] systematically explores the strategy space of the attacker. A new mining strategy stubborn mining is first proposed in it. The key to their new stubborn mining strategies is that the attacker should not give up so easily. In other words, the difference between stubborn mining and selfish mining is when to give up the private chain and adopt a longer chain from opponents.

2.3 Current Model of Proof of Work Blockchain

On modeling and simulation side, Eyal and Sirer [1] simulate selfish mining strategy. After their work, many works [4][5][8][9] built their model to simulate the proof of work blockchain with one attacker. Most of these works [4][8][9] analyzed selfish mining by using Markov Decision Processes. The discrete state space and action space for the player makes it fit for modeling mining behavior.

2.4 Stale Blocks

The security of blockchain is thoroughly studied in the recent year [1][11][12][13].It is related to its stale block rates. Stale blocks result from chain forks that are not included in the most extended chain. Thus the miner of a stale block will not earn block reward. The stale block rate directly represents the portion of wasted hashpower of a miner. Under most situation, the stale blocks are caused by occasional conflict ,and the stale block rate is quite low. According to Gervais [9], the stale block rate of Bitcoin is 0.41 percent and another work [3] shows that the probability is under 1.7 percent. Both of their works suggest that when all miners are honest, the possibility that stale blocks show up is low. When mining attacks especially selfish mining exists, the stale block rate will increase significantly.

Gervais [9] also use the stale block rate of the miner as a parameter to measure whether a proof of work blockchain model is safe or not. He discusses the cost of attacking behavior of selfish mining and double spending in a blockchain model with different stale block rate. His work connects stale blocks and security of a blockchain.

3 SYSTEM MODEL

In this section, we introduce our system model shown in Figure1 which can simulate the behavior of different miners and construct an environment where multiple selfish miners may occur.

Table 1: Table of notation

α	Computation power of the honest miner
β_i	Computation power of the $i^{th} attacker$
γ_h	Fraction of honest miner's computation power that will remain mining on honest miner's chain when honest miner and attackers are having a competition
γ_i	Fraction of attackers' computation power
SM	Strategy selfish mining
S_n	Strategy stubborn-n
P_n	Strategy publish-n

3.1 Our model

Our model has two phases, a blockchain instance and a proof of work blockchain simulator. A blockchain instance can be any cryptocurrencies based on proof of work blockchains such as Bitcoin or Ethereum. The output of the blockchain instance is the number of miners or mining pools and their corresponding fraction of hashpower. It will be used as the input of the blockchain simulator.In the blockchain simulator, each attacker's behavior is based on his state and action. The output of the simulator is the attackers' relative stale block rate. The notations in our model are mentioned in Table1.

3.2 Parameters

Our model has three main parameters:

Hashpower of the honest miner α: α is the fraction of the total hashpower controlled by the honest miner. This portion of miner follows the protocol of the proof of work blockchain. For instance, the honest miners of Bitcoin follow the Bitcoin protocol. To make it brief, we consider this portion of miners as an entirety and refer to it as Alice.

Hashpower of the i^{th} attacker β_i: In the basic models mentioned above, with one attacker, one value beta is enough to describe the hashpower of the attacker. While in our model, since we make the assumption that several attackers can exist simultaneously and they are independent to each other, the values of the attackers' hashpower should be an n-dimension set and $beta_i$ stands for the hashpower of the i^th attacker. We refer to these attackers as Bob_i. For any Bob_i and Bob_j, they are independent of each other, which means they do not share their state information. For Bob_i, the only method to affect Bob_j's state is to publish a new block on the main chain thus for Bob_j the behavior of Bob_i and Alice shows no difference. Note that $\sum_i \beta_i + \alpha = 1$.

The propagation ability of the honest miner γ_h: γ_h indicates whether the honest miner can be easily affected or not. A large value of γ_h means that the attackers can have little impact on the honest miners' choice.

The propagation ability of the attacker γ_i: With a large γ_i, attacker Bob_i can easily have an impact on the honest miners' choice.

Remained hashpower of the honest miner γ_h: $\gamma_h = 1 - \sum_i \gamma_i$;

3.3 Decision Process

An attacker needs to decide what action he should take and when to take an action.

Each attacker faces a single-player decision problem: M = (S, A, P, R) where S is state space, A is action space or decision space, P is the probability and R is the revenue of each action or decision. For Bob_i, when Bob_i or other miners find a block, Bob_i should make the action ,and the transition of its state will occur. For every state in Bob_i's state space:

$$P_a(S_1, S_2) = P(S_{t+1} = S_2 | S_t = S_1 \; and \; A_t = a) \tag{1}$$

For Alice, the honest miner, the action space is smaller. As an honest miner, Alice always follows the default protocol. She will publish the block as soon as she finds it and she will follow the longest published chain and work on the top of it.

3.3.1 State s_i: In our model, each attacker maintains a private state and the action of the attacker is based on his state. As a result, the following information should be included in the state:

- Whether there is a fork in the main chain: If several miners publish their chain at the same time and these chains have the same length, the fork will occur ,and under this situation, these miners are competing with each other. The competition will end if a miner publishes a new block after one of these chains or another attacker publishes a longer chain.
- Whether the attacker is involved in this competition: If the attacker is involved, he will mine on his chain. Otherwise, the action is up to the attacker's strategy.
- The attacker's lead: We define the lead of Bob_i as:

$$lead = len(Bob_i's \; chain) - len(Alice's \; chain) \tag{2}$$

The information above can be included in a 3-tuples T = (lead, f1, f2) in which f1 = 1 means the competition exists and f2 = 1 means the attacker is involved in the competition. Note that the state in which f1 = 0 and f2 = 1 is impossible.

To simplify the expression in our work, we define the state of each attacker $s_i = lead \; of \; the \; attaker$. At the same time, we denote the attacker's state in previous step as $prev_1$. With s_i and $prev_i$, the information in the 3-tuples can be inferred.

3.3.2 Action A_i: Bob_i can make the following actions: Hold, Match, Override, Adopt, Stubborn and Publish. Except for the last two actions, the other actions are mentioned in many pieces of research. Thus, we will only briefly introduce these five actions and put our emphasis on the attackers' behavior with this action space and state space in the environment with several attackers. We initially propose the action Stubborn and Publish.

Hold: Bob_i holds his private chain and keeps working on it until the state transition occurs.

Match: Bob_i releases all of his chain to generate a fork in the main chain. Under this situation, competition occurs.

Override: Bob_i publishes all or part of his chain and assures that his newly released chain is the longest chain.

Adopt: Bob_i gives up on his private chain and mines at the top of the main chain.

Publish: Bob_i publish the head of his blocks when his private chain achieves a certain length.

3.3.3 State transition. The state transition only occurs when a new block is found or published. In most cases, Bob_i has β_i possibility of mining the next block and Alice has alpha possibility of mining the next block. However, in some cases where competition occurs, due to the participants' propagation ability, Alice's hashpower will be split into different parts. We define the situation where Bob_i gets extra help from part of Alice's hashpower as redistribution of hashpower. Note that, once the competition is over, the separated hashpower of Alice will gather to the longest chain and mine at the top of this chain together.

From the i^{th} attacker Bob_i's perspective, the probability of state transition seems reasonable. However, the probability estimated by the attacker may not be the real state transition probability in the model with several attackers. For instance, when Bob_i's state is $s_i = 1$. From his perspective, when he applies the action hold, the probability to the state 0 is $1 - \beta_i$. But other attackers may take the same action as Bob_i and keep mining on their chain and their action may cause Bob_i's overestimating to the probability of state transition to 0. As a result, Bob_i may be misled and make the wrong choice between Adopt and Hold when the state is 1. Unfortunately, the gap between the real probability and the estimated probability of Bob_i cannot be eliminated since Bob_i has no idea of other miners' strategy and whether they are honest or not.

3.4 Revenue

We build connection between revenue and stale block rate to evaluate the performance of mining strategies.

Once a block is accepted by the chain, its finder will receive his block reward. The number of a miner's accepted blocks can directly show the revenue he gains. And an expectation of the revenue can be calculated by the miner where r_{tot} is the total revenue gained by a miner and r_{a_i} is the revenue gained in every action A_i:

$$r_{tot} = \mathcal{E}[\lim_{n \to +\infty} \sum_{i=1}^{n} r_{a_i}] \tag{3}$$

This number cannot indicate the efficiency of the miner. The attacking strategies are not always rational. The attackers' aim is not to increase their revenue but to increase their share of revenue. A simple comparison of the revenue gained by the attackers will not indicate whether a strategy works or not since when a mining attack exists, the victims and the attackers will both face a situation that they waste a portion of hashpower. Thus, instead of miners' revenue, miners' efficiency indicates whether a strategy works or not. In our model, we compare the miners' efficiency through their portion of wasted hashpower.

The portion of a miner's wasted hashpower can be measured by his stale block rate :

$$s_i = \frac{St_i}{St_i + Ac_i} \tag{4}$$

where St_i is the abandoned stale blocks and Ac_i is the block accepted by the main chain of the i^{th} attacker. The portion of the whole system's wasted hashpower can be measured by

$$T = \frac{St_h + \sum_i St_i}{Ac_h + St_h + \sum_i (St_i + Ac_i)} \tag{5}$$

where St_hstands for the honest miner's stale block and Ac_h is honest miner's accepted block. Then we define the relative stale block rate for the i^{th} attacker:

$$R_i = \frac{s_i}{T} \tag{6}$$

The value of R_i shows the relative efficiency of the i^{th} attacker and with $R_i < 1$, the i^{th} miner waste a less portion of hashpower than others and his aim of increasing the portion of his blocks in tha main chain can be achieved.

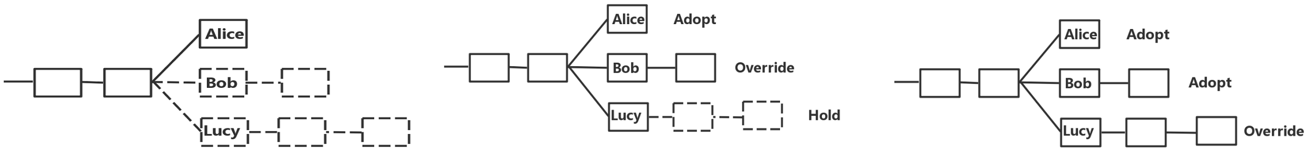
3.5 Mining behavior

With n attackers, the miners will face new situations. Thus, in this section, we discuss the miners' behavior when facing these situations.

For simplicity, we define the time between $block_i$ is mined and $block_{i+1}$ is mined as one round. An interesting fact in a blockchain with several attackers is that the attackers'state keeps changing in one round. As a result, the attacker's action varies.

First, recall the process of state transition in the model with only one attacker. For an attacker, he follows the selfish mining strategy and makes one action in one round. Once the action is made, the probability of his state transition in this round is ensured. In the blockchain with several attackers, the decision making process of the attackers seems like an auction and the state transition for the attacker Bob_i will be confirmed only if no new blocks are published in this round and his action will not change anymore. Note that, in one round, action Match and Override result in publishing of new blocks ,but only action override changes the length of blockchain.

To clarify this problem, we present a basic instance: Suppose there are three miners Alice, Bob and Lucy. Alice is an honest miner while Bob and Lucy are two selfish miners. For Bob and Lucy, they do not know each other in advance so that they have no access to each other's state. Assume that $S_{Bob} = 2$ and $S_{Lucy} = 3$ and their actions are both hold at this moment. When Alice reveals a new block, for Lucy, the state changes to 2 and the action is hold ,and for Bob, the action is override which changes his state to 0. Clearly, in this round, Lucy will continue to publish his chain and override the main chain again. At this moment, Lucy's state is 0

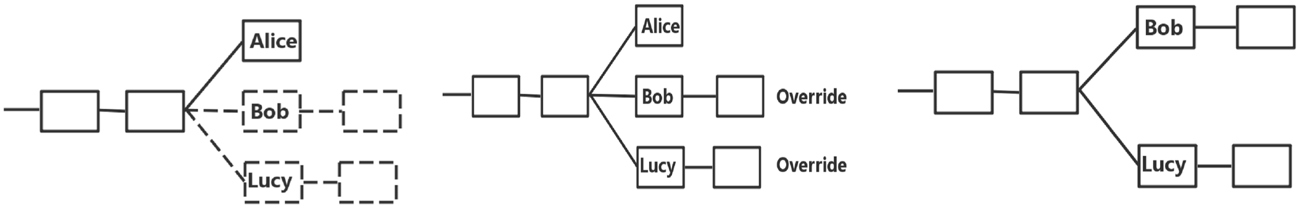


Step 1: Alice publishes her block

Step 2: Bob takes the action Override and Lucy takes action hold

Step 3: Lucy takes action Override

Figure 2: Two attackers (Bob and Lucy) with strategy selfish mining and an honest miner Alice's action in one round. The dash line represents for the unpublished blocks.



Step 1: Alice publishes her block

Step 2: Bob and Lucy takes action Override

Step 3: An unexpected competition between Bob and Lucy occurs

Figure 3: The generation of an unexpected competition when there are two attackers with strategy selfish mining an honest miner.

Algorithm 1 Attackers' behavior in one round

```
while new blocks are published do
    for $i = 0$ to $n$ do
        $Update\ attackers'\ state$
        $Update\ attackers'\ action$
    end for
    if $Action\ Override\ is\ made$ then
        $LenOfChain = LenOfChain + 1$
    end if
end while
```

with action hold and Bob's state is 0 with action Adopt. After Bob's state converts to 0 and his action converts from Adopt to Hold, in this round, no blocks will be published anymore and the state of all miners is finally be ensured. Figure2 illustrates this process in detail. In addition, we use Algorithm1 to indicate the attackers' mining behavior in one round.

Because of the variation of the attackers' action in one round, the blockchain network will arise some results which are beyond the attackers' expectation.

One of the results is called unexpected competition. In a proof of work blockchain with only one attacker, the competition occurs when the honest miner publishes a block and the attacker takes the action Match and releases one block to catch up the honest miners' chain. If the attacker's action is Override or Adopt, the competition will not show up since either the honest miner or the attacker gives up and accepts the opponent's chain. In our blockchain model with several attackers, unexpected competition shows up. In Figure3, the honest miner Alice publishes her newly found block, and two attackers Bob and Lucy hold their private chain of the length two respectively so that both Bob and Lucy publish two blocks to override Alice's chain. In this round, neither Bob nor Lucy means to start a competition, but a competition shows up.

3.6 Choice of β and γ

3.6.1 Value of β_i. We discuss the value of beta based on the real case: The hashpower of the mining pools in Bitcoin. Since selfish mining is a risky behavior, we assume that the miner cannot take the risk of being caught. Based on this assumption, the miner will be less likely to mine jointly if they are selfish. Table2 indicates the mining pool's hashpower of Bitcoin. The largest pool ever shown up in the past 3 years (2014-2017) takes 40% hashpower of the whole network. Nowadays,the largest mining pool of Bitcoin only occupies 21.9% hashpower of the whole network. If all the attackers attack the blockchain individually, the hashpower of a single attacker is less than 0.4. In our simulation, we decrease the upper boundary to 0.33 which is threshold to gain extra revenue even if all other miners are honest.

3.6.2 The value of γ_i and γ_h. The value of γ_i and γ_h is the greatest uncertainty in our model. A set of values is to be confirmed instead of one single value. The model will be too complicated if we determine γ_i respectively. Fortunately, three characteristics of mining behavior help us to simplify the model.

Table 2: Mining pool's hashpower of Bitcoin

Hashpower	Scale of the mining pool
40%	The largest mining pool of Bitcoin over the past 3 years. (2014-2017)
21.9%	The largest mining pool today. (2017.7)
12%	The second-largest mining pool today. (2017.7)

- The starter of one round is always the honest miner or the attacker with the strategy set publish-n: once an attacker applies selfish mining or stubborn-n strategy, he will hold his blocks until someone publishes a new block. The strategy he adopts does not allow him to publish a block on his initiative. Instead, he can use the action Match to start a competition in this round or use action override to lengthen his chain and finish one round.

- When the honest miner's block is still involved in the competition at the end of one round, it means that no attacker takes the action override. Once an attacker makes the action override, the honest miner has to adopt the attacker's chain since she has no unpublished blocks to match the length of the attackers' chain.

- For an attacker with the state n $n >= 2$, the priority level of action Override is higher than the action Match. It means that he will always take action override when his state changes from n to 1 instead of holding his blocks until his state changes to 0 and then taking the action Match.

Based on these three facts, the process of determining γ_i can be divided into two steps:

- Determine the portion of Alice's (honest miner's) remained hash power γ_h. If Alice is not involved in the competition, γ_h is 0.

- If the competition is an unexpected competition which means that it is caused by the action Override of several attackers, the hashpower of the honest miner will be evenly split between these attackers. Otherwise, the competitors' propagation is proportional to their hashpower.

In fact, γ_h is still in a wide range. For the best case, the propagation delay does not exist ,and the value of γ_h is 1. When propagation delay is taken into consideration, based on Bitcoin protocol, the propagation of a block takes three rounds of interaction and the first two rounds are optional. Due to several tricky methods such as Inv block attack and Eclipse attack, the information propagation of Alice's newly discovered block can be delayed by all attackers. For the worst case, all the honest miners are eclipsed so that $\gamma_h = 0$. Thus, in our paper, with a more complex and chaotic environment, the range of γ_h will not be restricted. Meanwhile, to simplify the simulation, γ_i, the attackers' propagation ability will be proportional to their hashpower.

4 MINING STRATEGY

Generally speaking, the mining strategy is about when to take the action adopt or when to take action publish. In this section, we explore existing mining strategies and propose our new mining strategies. These strategies built up the strategy space in our model. We introduce the behavior of these mining strategies through pseudo code and display the properties of these strategies through some simulation result.

4.1 Revisiting of existing strategies

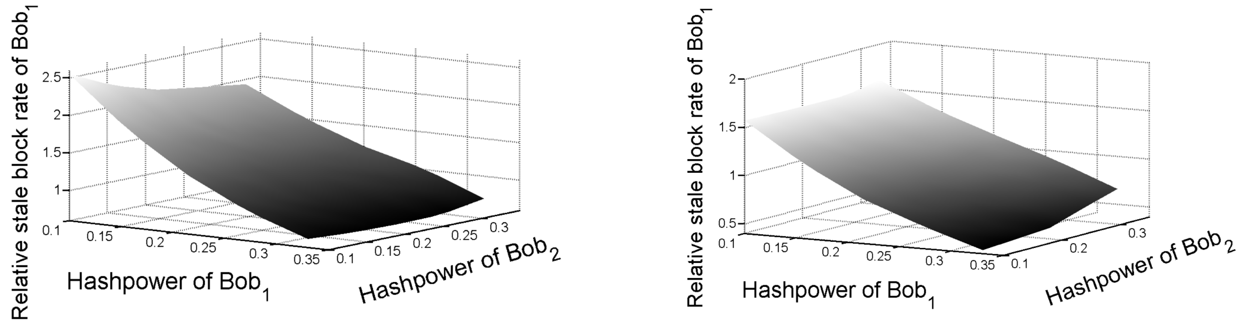
4.1.1 Strategy Selfish Mining. First, consider the case in which there is only one attacker and the other miners are all honest. The behavior of selfish mining strategy is illustrated in Algorithm2.

Many existing works indicate that when the value of γ_h is 1, the threshold of hashpower for the attacker to gain extra revenue is 1/3 while the value of γ_h drops to 0.5, the threshold drops to 0.25. As an attacker whose hashpower is less than 1/3, if there is no evidence that another attacker exists, he must consider carefully about whether to launch a selfish mining attack according to the value of γ_h.

Figure4 shows the relative stale block rate of Bob_1 when the number of attackers is 2 and the value of γ_h is 1 and 0.5 respectively. When $\gamma_h = 1$, we focus on a specific value of Bob_1's hashpower —- 0.33 which is the threshold for Bob_1 to gain extra revenue when there is only one attacker. As we can observe from the simulation result, for Bob, the threshold is no longer 1/3. Instead, the threshold for Bob to gain extra revenue is determined by the hashpower of Lucy. When $\gamma_h = 0.5$ and the hashpower of Lucy is relatively small (typically less than Bob), the threshold of Bob is less than 0.25 ,and it can even drop to 0.20. With the increase of Lucy's hashpower, the threshold for Bob also increases. As suggested in the simulation result, when Lucy's hashpower is higher than about 0.3, the threshold for Bob will larger than 0.25.

The threshold is also determined by the hashpower of Lucy. Even if the hashpower of Bob reach the threshold with which he can earn extra revenue when he is the only attacker, he cannot necessarily gain additional revenue.

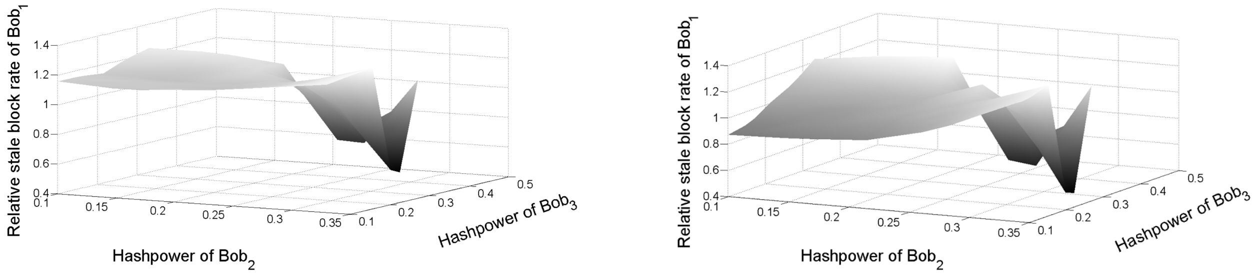
In a proof of work blockchain with several attackers, the environment becomes more complicated and there is no longer a certain value of threshold which ensures the attacker to gain extra revenue. When the attacker is tending to launch an attack with strategy selfish mining, he should not only consider the value of γ_h but also should consider the number of his opponents and his opponents' hashpower.



γ_h = 1.

γ_h = 0.5.

Figure 4: The relative stale block rate of attacker Bob_1



γ_h **= 1**

γ_h **= 0.5**

Figure 5: Relative stale block rate of an attacker Bob_i with 20 % hashpower

Result 1: The threshold for an attacker to gain extra revenue drops since the hashpower of his opponents are more separated. The threshold is related to the value of γ_h and the hashpower of another attacker.

Result 2: Bob_2 has a positive impact on Bob_1 when Bob_1's hashpower is low (Less than 0.2). When Bob_1's hashpower is high, they start to compete with each other and Bob_2 has a negative impact on Bob_1.

Then, we start to add the number of attackers. Note that when the number of attackers Bob is two and the attackers' hashpower is 0.2, he cannot gain extra revenue when γ_h is 0.5 or 1. Figure5 indicates the simulation result with more than two attackers, we set the hashpower of Bob_1 as a constant 0.2. The scope of the other two attackers' hashpower is 0.1 and 0.33. Bob_1 still has the chance to gain extra revenue while under most of the circumstance, with 20 % of hashpower, it is unwise for Bob to launch a selfish mining attack.

4.1.2 A strategy set: Stubborn-n. In most of the circumstance, when an attacker's private chain falls behind the honest miner's chain, because of the hashpower differential between the attacker and the honest miner, the attacker usually takes action adopt and adopts the honest miner's chain. When taking the action adopt, the effort of the attacker is totally wasted. Sometimes, not giving up the private chain so easily can earn unexpected revenue.

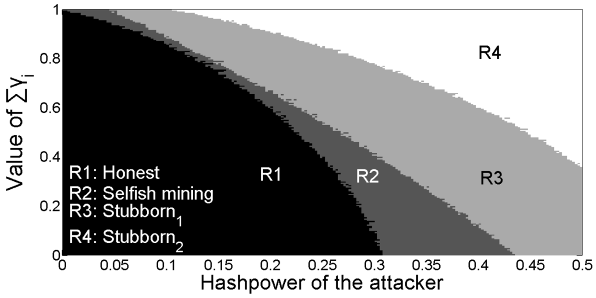


Figure 6: Dominant strategy for different value of β and γ_h when there is one attacker.

Stubborn-n is a strategy set and n represents the persistent degree of the attacker. If an attacker Bob takes the strategy Stubborn-j, j new states from -1 to -j are added to his state space. When a new block is found by his opponents, he gives up at state -j instead of state 0. According to this description, strategy selfish mining is a special instance of stubborn-n with the value of n =0. To avoid garble, when strategy Stubborn-n is mentioned about, the default value of n is greater than 0. The behavior of attacker with strategy stubborn-n is illustrated in Algorithm3.

Algorithm 2 Selfish Mining

```
LenPrivateChain = 0
PrivateChain = PublicChain
while Mining do
    if MyPoolFound then
        prev = state
        state = state + 1
        LenPrivateChain = LenPrivateChain + 1
        if prev == 0 and PrivateChain == 2 then
            publish this block
            state = 0
            LenPrivateChain = 0
        else
            Action hold
        end if
    else
        prev = state
        state = state - 1
        if prev == 0 then
            Action Adopt
            private chain == public chain
            LenPrivateChain = 0
            state = 0
        else if prev == 1 then
            Action Match
            state = 0
        else if prev == 2 then
            Action Override
            state = 0
            LenPrivateChain = 0
        else
            Publish the first unpublished block
        end if
    end if
end while
```

Consider the case in which there is only one attacker ,and the other miners are all honest. Since our strategy space has been enlarged to $\{S_1, ..., S_n, SM\}$, we test the efficiency of different attacking strategies and find out which one is optimal under a large parameter space.

Figure6 is the simulation result when there is only one attacker. The regions in the result indicate that a certain strategy outperforms others in a certain parameter space. In most of the circumstance, strategy selfish mining is not the best option and when the hashpower of the attacker grows, the value of n increases.

Result 3: Strategy stubborn-n has a lower relative stale block rate than selfish mining in the parameter space where hashpower of the attacker is high when there is only one attacker. The performance of stubborn-n indicates that compared with selfish mining it waste more hashpower of the honest miner.

Algorithm 3 Stubborn-n

```
LenPrivateChain = 0
PrivateChain = PublicChain
while Mining do
    if MyPoolFound then
        prev = state
        state = state + 1
        LenPrivateChain = LenPrivateChain + 1
        if prev == 0 and PrivateChain == 2 then
            publish this block
            state = 0
            LenPrivateChain = 0
        else
            Action hold
        end if
    else
        prev = state
        state = state - 1
        if prev == -n then
            Action Adopt
            private chain == public chain
            LenPrivateChain = 0
            state = 0
        else if prev > -n and prev <= 0 then
            Action hold
        else if prev == 1 then
            Action Match
            state = 0
        else if prev == 2 then
            Action Override
            LenPrivateChain = 0
            state = 0
        else
            Publish the first unpublished block
        end if
    end if
end while
```

4.2 A new strategy set: Publish-n

During the attack, the attacker may face an embarrassing situation: He holds a long private chain and it turns out that he still falls behind the main chain. Under this situation, he may face a choice: either to take the action adopt and give up the efforts he made in a long period of time or choose the strategy S_n.

He has another option: Applying the strategy set publish-n, denoted by P_n. This strategy is originally proposed by us. The value of n can be seen as a cordon the attacker set for his state. When his state reaches n, he will either publish the first block of his private chain or take the action override depending on whether he finds the next block or not. This strategy helps the attacker to shorten his private chain quickly so that his state will never exceed *n*. Algorithm4 indicates the behavior of strategy P_n.

Actually, P_n can be seen as a combination of selfish mining and honest mining, when the attacker reaches state n, he acts like an

Algorithm 4 Publish-n

```
LenPrivateChain = 0
PrivateChain = PublicChain
while Mining do
    if MyPoolFound then
        prev = state
        state = state + 1
        LenPrivateChain = LenPrivateChain + 1
        if prev == 0 and PrivateChain == 2 then
            publish this block
            state = 0
            LenPrivateChain = 0
        else
            if prev < n then
                Action hold
            else
                Publish the first unpublished block
            end if
        end if
    else
        prev = state
        state = state - 1
        if prev == 0 then
            Action Adopt
            private chain == public chain
            LenPrivateChain = 0
            state = 0
        else if prev == 1 then
            Action Match
            state = 0
        else if prev == 2 then
            Action Override
            LenPrivateChain = 0
            state = 0
        else
            if prev <n then
                Publish the first unpublished block
            else
                Action Override
                LenPrivateChain = LenPrivateChain -2
                state = state - 1
            end if
        end if
    end if
end while
```

honest miner if he finds the next block while he acts like a selfish miner at state 2. Note that, the behavior of P_1 is equivalent to the honest miner and P_2 is similar to a selfish miner. When taking about P_n, our default value of n is $n > 2$. Meanwhile, for a P_n miner with hashpower of β_i, the probability to reach state n is:

$$P_{s\to n} = {\beta_i}^n \tag{7}$$

$$\lim_{n\to+\infty} P_{s\to n} = \lim_{n\to+\infty} {\beta_i}^n = 0 \tag{8}$$

Thus, when n is sufficiently large, the behavior of P_n can be equivalent to Selfish mining.

From Algorithm4, we can notice that an attacker with strategy publish-n will publish his block initiative when he reaches state n. This characteristic of publish-n determines that it wastes less hashpower of the honest miner than the selfish miner if there is only one attacker in a proof of work blockchain.

5 PERFORMANCE OF DIFFERENT MINING STRATEGIES

In this section, the attackers will take strategy P_n, S_n and selfish mining at the same time and their performance will be compared. We use numeric simulations to evaluate the stale block rate of the miners. We simulate 100 paths of the state machine and for each path and iterate for 100000 times. In our simulation, the hashpower of the attackers will be the same. They will launch an attack independently while they can have an impact on the honest miner and the other attackers. The most well-known mining attack strategy —- selfish mining will be used as a standard of comparison. Other mining attack strategy will be compared with selfish mining in our blockchain model with several attackers.

5.1 Stubborn-n against selfish mining

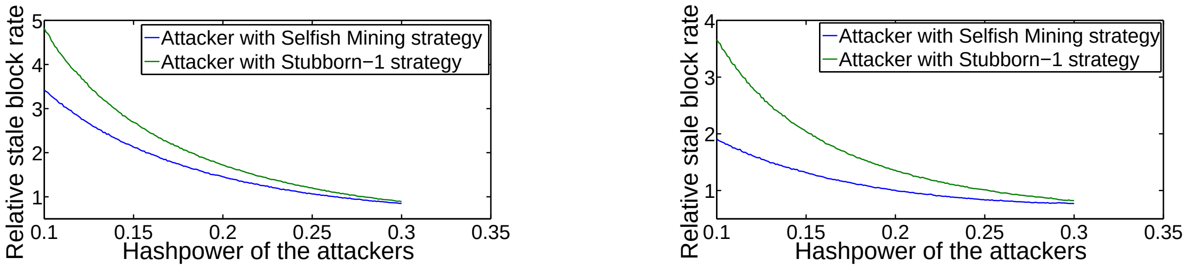
To test the performance of stubborn mining, we simulate stubborn mining in our blockchain model where one honest miner and one selfish miner exist. In our simulation, both of the attackers hashpower will not exceed 33 percent so that the honest miner is still the majority. Among the strategy set S_n, we choose S_1 which has the lowest persistent degree to compare with selfish mining.

Figure7 illustrates the simulation result under the situation where γ_h is 1 and 0.5. Selfish mining outperforms S_1 from the beginning to the end. The relative stale block rate of selfish mining is always lower than stubborn-1 which indicates that selfish mining is a more efficient strategy when there are more than one attackers in the blockchain model. When the hashpower of both the attackers grows, stubborn-1 narrows the gap.

Another fact which can be observed from the simulation result is that with the decrease of the value of γ_h, the gap between stubborn-1 and selfish mining is increasing. It indicates that, when the honest miner can be easily influenced, strategy selfish mining receives more support from the honest miner.

Result 4: When $\gamma_h = 1$, the relative stale block rate of an attacker with strategy selfish mining is 40% lower than the attacker with strategy stubborn-1 while the attackers' hashpower is 0.1. When their hashpower increases to 0.3, the relative stale block rate of selfish mining is only 4.3% lower than stubborn-1.

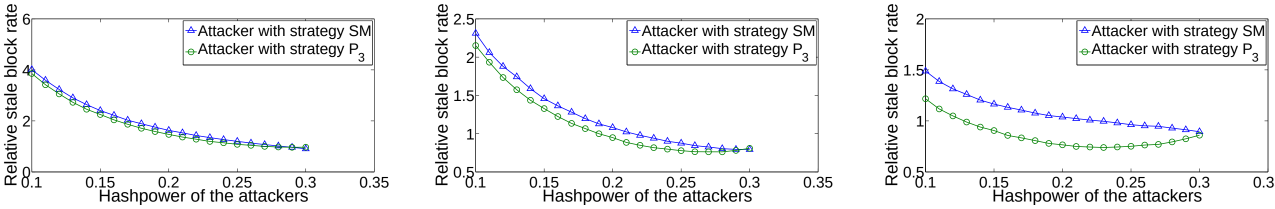
Generally speaking, in a blockchain with several attackers, the hashpower of the attackers is more separated. Under this situation, Stubborn-n is suboptimal compared with selfish mining. Stubborn-n is fit for the situation where the hashpower of the attacker is high.



$\gamma_h = 1$

$\gamma_h = 0.5$

Figure 7: Comparison between an attacker with strategy Stubborn-1 and another attacker with strategy Selfish mining



$\gamma_h = 1$

$\gamma_h = 0.5$

$\gamma_h = 0.2$

Figure 8: Comparison between an attacker with strategy publish-3 and another attacker with strategy Selfish mining

Result 5: When $\gamma_h = 0.5$, the relative stale block rate of selfish mining is 89% lower than stubborn-1 while the attackers' hashpower is 0.1. When their hashpower increases to 0.3, the relative stale block rate of selfish mining is 12.1% lower than stubborn-1.

The decrease of γ_h benefits attackers with strategy selfish mining instead of stubborn-1. In addition, with the value of n increases, the attacker with strategy stubborn-n is more persistent on his private chain so that he will get less support from the honest miner. We draw the conclusion that selfish mining outperforms stubborn-n when several attackers launch attacks at the same time.

5.2 Publish-n against selfish mining

Strategy publish-n does not fit for the blockchain model with only one attacker. In a proof of work blockchain with several attackers, an attacker should not only consider wasting his opponents' computation power but also consider earning the honest miner's support. The failure of strategy stubborn-1 gives a full illustration of this point.

In the simulation we have one honest miner, one attacker who takes the selfish mining strategy and another attacker who takes the p_n strategy. Among the strategy set p_n, we select p_3 since the difference between selfish mining and p_3 is more significant than any other strategies in the strategy set p_n.

In Figure8, the relative stale block rate of an attacker with strategy p_3 is lower than the attacker with strategy selfish mining when the hashpower of both attackers are low. When the hashpower increases, the performance of selfish mining narrows the gap and eventually it outperforms p_3. Another fact which can be observed from the simulation result is that when the value of γ_h is lower, p_3 performs better. This phenomenon indicates that strategy p_3 can gain more support from the honest miner.

Result 6: With $\gamma_h = 1$, the efficiency of publish-1 is 0.69% better than selfish mining when the hashpower of attackers is 0.1 and the efficiency is 2.25% worse than selfish mining when the hashpower of attackers is 0.3.

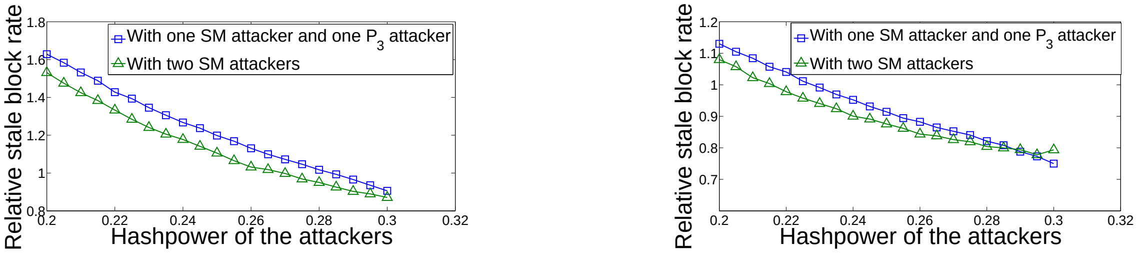
Result 7: With $\gamma_h = 0.2$, the efficiency of publish-1 is 26.3% better than selfish mining when the hashpower of attackers is 0.1 and the efficiency is 3.78% better than selfish mining when the hashpower of attackers is 0.3.

When the hashpower of the attackers is low, strategy publish-n has lower relative stale block rate than selfish mining. With the increasing of the attackers' hashpower, selfish mining eventually outperforms publish-n. With the value of γ_h dropping, the honest miners are more likely to accept the chain published by the attackers and gap between the two different strategies grows larger.

Figure9 compares the relative stale block rate of the selfish miner in the blockchain model with two selfish miners and the selfish miner in the blockchain model with one selfish miner and one p_3 miner. The simulation result indicates that the selfish miner in the model with one selfish miner and one p_3 miner always earn less revenue.

To find the lost revenue, we compared the relative stale block rate of the honest miner in the two situations mentioned above.

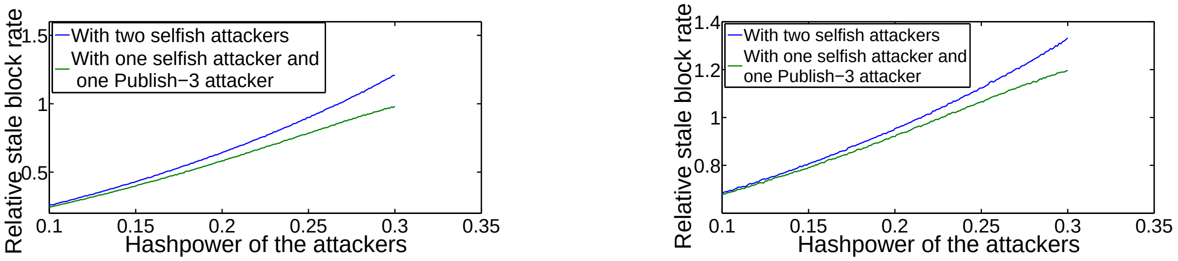
Figure10 displays a great decrease of the honest miner's relative stale block rate. Under the circumstance $\gamma_h = 1$, the honest



$\gamma_h = 1$

$\gamma_h = 0.5$

Figure 9: Comparison between one selfish miner in a model with two selfish miners and in a model with one selfish miner and one p_n miner



$\gamma_h = 1$

$\gamma_h = 0.2$

Figure 10: Honest miner's relative stale block rate

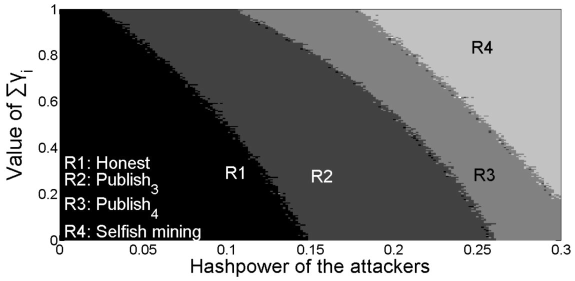


Figure 11: Dominant strategy for different value of β and γ_h with 3 attackers

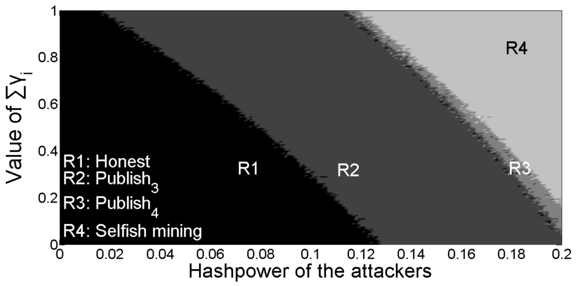


Figure 12: Dominant strategy for different value of β and γ_h with 5 attackers

miner can even gain extra revenue. The poor selfish mining attacker becomes the victim of the strategy of P_n

Result 8: The strategy P_n decreases the revenue of selfish mining attacker. This part of revenue not only benefit the P_n miner but also benefits the honest miner if the attackers' hashpower is low.

In the discussion above, the number of attackers is limited to two. The situation in which more attackers launch the attack should also be considered. The increase of attacker will result in a complicate mining circumstance and the decrease of the hashpower of the honest miner. Thus, in this section, we will not assure that the honest is the majority. The hashpower of the honest miner is in a wide range from 0 to 1.

First, we consider the case in which 3 attackers with strategy P_i, P_j and selfish mining respectively. We find the fact that when the value of n is greater than 5, there is no significant difference between the mining result of P_n and selfish mining. Thus, we set the value of i to 3 and j to 4.

Figure11 is the simulation result. Each region represents that a certain mining strategy performs the best with the parameter space of the region. Strategy P_3 has the lowest relative stale block rate among the three mining attack strategies when the hashpower of the attackers are low while selfish mining outperforms other strategies when the hashpower of the attackers are high.

Then strategy s_1 and s_2 are added to the model so that the number of attackers increases to 5. Although the simulation result in Figure7 has proved that strategy s_n do not perform well when there are several attackers, they still have impacts on other attackers.

Figure12 is the simulation result. As expected, there is no region for strategy s_1 and s_2. The greatest difference between the Figure12 and Figure11 is that the region for strategy p_4 almost disappear. The edge between region p_4 , region p_3 and region SM is blurry. This result indicates that the efficiency of p_3, p_4 and SM is too close when the number of attackers is 5.

6 CONCLUSION AND FUTURE WORK

6.1 Detection of mining attack

Our conclusion about mining attack is that: mining attack is easy to be detected but the attacker is difficult to be caught. The detection of mining attack results from the variety of the stale block rate of miners. Fluctuations in the value of total stale block rate can be detected easily. But owning the information of the stale block rate of each miner or mining pool is not enough to find who is the attacker especially when multiple attackers are launching attacks to a proof of work blockchain.

6.2 Mining attack is risky

One reason is that due to the discovery of strategies to earn extra revenue in mining, the Bitcoin community deploys monitors to monitor the behaviors of miners.

Another explanation is based on our simulation result. For a miner with low computation power, typically less than 20%, he can barely gain extra revenue even if there are three attackers in the blockchain. That means, under most circumstance, he cannot earn extra revenue compared with mining honestly. Since he knows nothing about other miners' strategy space, he cannot cooperate with other attackers either. For an attacker with a large amount of computation power, typically larger than 30%, he indeed has the power to launch an attack and gain extra revenue compared with mining honestly. Other miners will soon be aware of the fact that someone has launched an attack according to the raising stale block rate. According to Result 8, when other miners take strategy publish-n the efficiency of the attacker will drop significantly. He may find an embarrassing fact that no one in the blockchain network earns more than before, including himself. A huge amount of computation power has been wasted.

6.3 P_n receives more support than S_n

Stubborn-n has the lowest relative stale block rate compared with other mining strategies when there is only one attacker. When the number of attackers increases, strategy stubborn-n soon lose its advantage. We draw the conclusion that strategy stubborn-n pays too much focus on wasting his opponents' computation power and when his opponent is the honest miner, this strategy always works.

When competing with the honest miner and other attackers, another aspect should be noticed: getting the support from the honest miner. In a blockchain model with several attackers, forks exist more frequently. Being the first one to publish the block helps gain the support from the honest miner. This is the reason why publish-n strategy success in the competition of several attackers when the computation power of the attacker is low. But strategy publish-n also has a side effect: The attacker wastes less computation power of his opponents. When the computation power of the attacker raises, this side effect's influence becomes more significant.

7 FUTURE WORK

We show that in a proof of work blockchain, several attackers may show up. Through the model, we analyze the attackers' potential behaviors and adding publish-n to strategy space. Our work leaves the following challenge:

- The miners' behavior in a proof of work blockchain with several attackers can be explored.
- With the existence of several attackers, the stale block rate or relative stale block rate is not enough to distinguish the attacker and the honest miner. How to detect the attackers remains a problem to be solved.
- Our simulation only shows the performance of different strategies. Actually, an attacker can change his attacking strategy on his own initiative. Determining when to shift from one strategy to another remain to be discussed.

REFERENCES

[1] Eyal I, Sirer E G. Majority is not enough: Bitcoin mining is vulnerable[C]//International conference on financial cryptography and data security. Springer, Berlin, Heidelberg, 2014: 436-454.

[2] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. 2008.

[3] Decker C, Wattenhofer R. Information propagation in the bitcoin network[C]//Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on. IEEE, 2013: 1-10.

[4] Sapirshtein A, Sompolinsky Y, Zohar A. Optimal selfish mining strategies in bitcoin[C]//International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2016: 515-532.

[5] Eyal I. The miner's dilemma[C]//Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 2015: 89-103.

[6] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin's peer-to-peer network.Usenix, 2015.

[7] A.Gervais,H.Ritzdorf,G.O.Karame and S.Capkun. Tampering with the Delivery of Blocks and Transactions in Bitcoin,Acm Sigsac Conference on Computer and Communications Security, 2015.

[8] K.Nayak,S.Kumar,A.Miller and E.Shi.Stubborn Mining:Generalizing Selfish Mining and Combining with an Eclipse Attack,IEEE European Symposium on Security and Privacy,2016.

[9] A. Gervais,Ghassan O. Karame and K.Wust. On the Security and Performance of Proof of Work Blockchains, Acm Sigsac Conference on Computer and Communications Security,2016.

[10] Economist Staff (31 October 2015). "Blockchains: The great chain of being sure about things". The Economist. Retrieved 18 June 2016.

[11] Y.Sompolinsky and A.Zohar. Secure high-rate transaction processing in bitcoin. In Financial Cryptography and Data Security, 2015.

[12] Ghassan O. Karame, E.Androulaki, and S.Capkun.Double-spending fast payments in bitcoin. Conference on Computer and communications security, 2012.

[13] M.Rosenfeld.Analysis of hashrate-based double spending.arXiv preprint arXiv:1402.2009,2014.

[14] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S.Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pages 919-927. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[15] Miller A, Litton J, Pachulski A, et al. Discovering bitcoinâĂŹs public topology and influential nodes[J]. et al., 2015.

[16] Bag S, Ruj S, Sakurai K. Bitcoin block withholding attack: Analysis and mitigation[J]. IEEE Transactions on Information Forensics and Security, 2017.