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

Hanqing Liu
Dept. of ECE Shanghai Jiao Tong University
Shanghai, P.R. China
hqliu233@sjtu.edu.cn

Na Ruan[*]
Dept. of ECE Shanghai Jiao Tong University
Shanghai, P.R. China
naruan@cs.sjtu.edu.cn

Rongtian Du
Dept. of ECE Shanghai Jiao Tong University
Shanghai, P.R. China
evil_devil@sjtu.edu.cn

Weijia Jia
Dept. of ECE Shanghai Jiao Tong University
Shanghai, P.R. China
jiawj@sjtu.edu.cn

ABSTRACT

Selfish mining is a well-known mining attack strategy discovered by Eyal and Sirer in 2014. After that, the attackers' strategy has been further discussed by many other works, which analyze the strategy and behavior of a single attacker. The extension of the strategy research is greatly restricted by the assumption that there is only one attacker in the blockchain network, since, in many cases, a proof of work blockchain has multiple attackers. The attackers can be independent of others instead of sharing information and attacking the blockchain as a whole. In this paper, we will establish a new model to analyze the miners' behavior in a proof of work blockchain with multiple attackers. Based on our model, we extend the attackers' strategy by proposing a new strategy set publish-n. Meanwhile, we will also review other attacking strategies such as selfish mining and stubborn mining in our model to explore whether these strategies work or not when there are multiple attackers. The performances of different strategies are compared using 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%.

CCS CONCEPTS

• **Security and privacy → Economics of security and privacy**; *Distributed systems security*;

KEYWORDS

Bitcoin, Mining, Selfish mining, N-attackers

ACM Reference Format:
Hanqing Liu, Na Ruan, Rongtian Du, and Weijia Jia. 2018. On the Strategy and Behavior of Bitcoin Mining with N-attackers. In *ASIA CCS '18: 2018 ACM Asia Conference on Computer and Communications Security, June 4–8, 2018, Incheon, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3196494.3196512

1 INTRODUCTION

Traditional payment on the Internet is based on trusted third parties, making completely irreversible transactions impossible[11] to achieve. This has arisen public's interest in decentralized cryptocurrencies based on cryptographic proof, as represented by Bitcoin. These cryptocurrencies use blockchain, a distributed database used to store and maintain a list of records[16], as its underlying technology. 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 (PoW) based blockchains account for 90 percent of the market. In a proof of work blockchain, a miner or a mining pool 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 gain extra revenue by taking some strategies, including the strategy called selfish mining represented by Eyal and Sirer in 2014. Many other strategies such as stubborn mining are the extensions of selfish mining. Basically, We call these strategies selfish mining style strategies.

In Bitcoin, selfish mining style attacks have not yet stood out due to the stable environment of Bitcoin. Statistics from *blockchain.info* indicate that in the past few years, the difficulty to find a new block has increased by four times.

A proof of work blockchain might have multiple attackers once the block reward drops to half of the current reward in the future or the price of Bitcoins drops due to realistic factors. A crisis may show up which results in an increasing likelihood for a miner or a mining pool to take tricky strategies. Once one attacker exists, the other miners can either stick to the Bitcoin protocol and lose part of their share of revenue, or become attackers ae well and steal the honest miners' revenue to compensate for his loss, which is more appealing to a miner compared with the former one. Therefore, 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 based on a proof of work blockchain with multiple attackers to explore the attackers' behaviors and their mining strategies. Existing works about mining attacks [4, 7, 12, 14] put their emphasis on the development of one single

[*] Corresponding author

ASIA CCS '18, June 4–8, 2018, Incheon, Republic of Korea
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5576-6/18/06…$15.00
https://doi.org/10.1145/3196494.3196512

attacker's strategy space. As far as we know, the miners' behaviors and strategies in a proof of work based blockchain with multiple attackers have not yet been studied in detail. What new action will be made and whether the attacking strategies for a single attacker still work have not been analyzed yet.

Contribution 1: Establishing a new model of a proof of work blockchain. Our model allows the existence of multiple attackers. The attackers do not share information, and they will have an impact on each other by publishing new blocks. Their decision-making process is independent, but their state transition depends on other miners. A proof of work blockchain model with multiple attackers is first discussed in this paper, which means that 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 to a strategy named publish-*n*. Our simulation result turns out that publish-*n* strategy performs better than other strategies when there are mutiple attackers if the attackers' hashpower is low. This strategy set allows the attacker to earn more profits.

Contribution 3: Reviewing existing strategies. We review selfish mining proposed by Eyal and Sirer[4] and stubborn mining proposed by Nayak[12]. Stubborn mining may not be a good choice in a blockchain with multiple 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 rest of our paper is organized as below: In section 2, we introduce the basic concepts and the attackers' strategy in a proof of work blockchain. In Section 3, we introduce our model and present the attackers' potential state space and 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 Bitcoin Mining

Bitcoin system uses a proof-of-work system to implement a distributued timestamp sever [11]. In Bitcoin, the header of a block contains the previous block header's hash value, a Merkle root of the transactions, and a nonce. The work of finding a new block is searching for the header which when hashed, the result begins with a number of zero bits. In a short word, a miner's work is to find a valid nonce based on Bitcoin protocol. When he finds the nonce, he generates a new block and broadcasts it to every other node in the Bitcoin network.

Mining pools are organized since a single miner has a huge variance in his reward, especially when the mining diffucuty increases [3, 9]. A mining pool has one manager and several miners. The manager allots work to others, and a miner uses his hashpower to generate partial proof of work (PPoWs). PPoWs can verify that a miner is using his hashpower to solve the PoW for the mining pool. A manager finds full proof of work (FPoW) among the PPoWs submitted by other miners. Only a FPoW can fetch an incentive of the block reward. The difficulty of generating a PPoW is lower than that of an FPoW.

2.2 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 she reveals a block immediately after she finds it. She always accepts the most extended chain and mines on top of the chain. 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.

An attacker can be either an individual miner or a mining pool. If two attackers shares infomation and use the same attacking strategy, they can be seen as one entity. Otherwise, they will be seen as two different entities. Due to the same principle, since an honest miner does not try to hide any infomation from others, all honest miners can be seen as an entity.

2.3 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 revenue larger than its ratio of mining power[4]. An attacker with more than 33 percent of hashpower can gain an extra revenue using the selfish mining strategy. The threshold can even be lower if the attacker influences the honest miner. In this case, selfish mining wastes the hashpower of the honest miner. Note that, selfish mining is an irrational strategy, which means that the attacker's revenue will also drop in a short term until the difficulty of mining decreased. Several works have analyzed that selfish mining strategy is suboptimal [12, 14].

After Eyal and Sirer's work, many works have analyzed mining attack. Some works can be regarded as the extension of selfish mining [12, 14], and works like E. Heilman's and A.Gervais's describe a network-level attack, eclipse attack [6, 7]. K.Nayak systematically explores the strategy space of the attacker [12]. A new mining strategy stubborn mining is first proposed by K.Nayak, the key of which is that the attacker does 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. Meanwhile, attacking strategies in the case where pools use some of its participants to infiltrate other pools are also discussed in many works [1, 3, 9, 10].

2.4 Current Model of Proof of Work Blockchain

On modeling and simulation side, Eyal and Sirer [4] simulate selfish mining strategy. After their work, many works built their model to simulate the proof of work blockchain with one attacker [3, 5, 12, 14]. Most of these works [5, 12, 14] analyzed selfish mining by using Markov Decision Processes. The discrete state space and action space for the player make it suitable to model mining behavior.

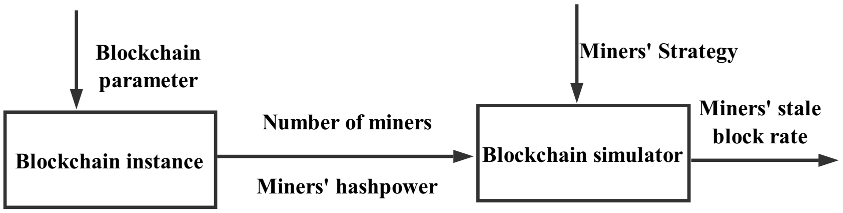


Figure 1: Our blockchain model with two phases

2.5 Stale Blocks

The security of blockchain has been thoroughly studied in recent years [4, 8, 13, 15], which is closely related to its stale block rates. Stale blocks result from chain forks that are not included in the most extended chain, which means that the miner of a stale block will not earn block reward. The stale block rate directly represents the proportion of wasted hashpower of a miner. In most situations, the stale blocks are caused by occasional conflict, in which case the stale block rate is quite low. According to Gervais [5], the stale block rate of Bitcoin is 0.41 percent, and Decker's work [2] shows that the occasional conflict probability is under 1.7 percent. Their works suggest that when all miners are honest, the possibility of the existence of stale blocks is quite low. When mining attacks, particularly selfish mining, exists, the stale block rate will increase significantly.

Gervais [5] also uses 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 like selfish mining and double spending in a blockchain model with different stale block rate. His work connects stale blocks with the 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 exist.

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 porportion of hashpower, which 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' stale block rate. The notations in our model are mentioned in Table1.

3.2 Parameters

Our model has three main parameters:

Table 1: Table of notation

α	Computation power of the honest miner
β_i	Computation power of the $i^{th} attacker$
γ_h	The radio of honest miners that choose to mine at the top of honest miners' chain
γ_i	The radio of honest miners that choose to mine at the top of the i^{th} attacker's chain
SM	Strategy selfish mining
S_n	Strategy stubborn-n
P_n	Strategy publish-n

Hashpower of the honest miner α: α is the porportion of hashpower controlled by the honest miner. This portion of miner follows the protocol of the proof of work blockchain. For example, the honest miners of Bitcoin follow the Bitcoin protocol. Simply, we consider this portion of miners as an entire and mark it as Alice.

Hashpower of the i^{th} attacker β_i: In the basic models mentioned above, with one attacker, one value β is enough to describe the hashpower of the attacker. In our model, we have made the assumption that multiple attackers can exist simultaneously and they are independent from each other, therefore the values of the attackers' hashpower should be an n-dimension set, in which β_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. Note that $\sum_i \beta_i + \alpha = 1$.

The radio of honest miners that choose to mine at the top of honest miners' chain γ_h: γ_h is the radio of honest miners that choose to mine at the top of honest miners' chain. 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 radio of honest miners that choose to mine at the top of the i^{th} attacker's chain γ_i: When the i^{th} attacker Bob_i competes with others, generating a fork, a fraction of honest miners will consider Bob_i's chain as the new main chain and work at the top of it. With a large γ_i, attacker Bob_i can easily have an impact on the honest miners' choice. Note that, $\gamma_h = 1 - \sum_i \gamma_i$.

3.3 Decision Process

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

Each attacker faces a single-player decision problem: M = (STATE, A, P, R) where STATE 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 take an action, and the transition of his state will happen. For every state in Bob_i's state space:

$$P_a(STATE_1, STATE_2) = \\ P(STATE_{t+1} = STATE_2 | STATE_t = STATE_1 \ and \ A_t = a) \tag{1}$$

For Alice, the honest miner, the action space is smaller. As an honest miner, Alice always follows the 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 $STATE_i$: 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, a fork will exist. 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, f_1, f_2) in which f_1 = 1 means the competition exists and f_2 = 1 means the attacker is involved in the competition. Note that the state in which f_1 = 0 and f_2 = 1 is impossible.

To simplify the expression in our work, we define the state of each attacker $STATE_i = lead \ of \ the \ attacker$. At the same time, we denote the attacker's state in previous step as $prev_1$. With $STATE_i$ 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, and Publish. These are basic actions and have been mentioned in many other works. Thus, we will only briefly introduce these five actions and lay our emphasis on the attackers' behavior with this action space and state space in the environment with multipe attackers.

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.

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 α possibility of mining the next block. However, in some cases where competition occurs, Alice's hashpower will be split into different parts, due to the participants' propagation ability. We define the situation where Bob_i gets extra help from part of Alice's hashpower as redistribution of hashpower. 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 multiple attackers. For example, when Bob_i's state is $STATE_i = 1$, for him, if 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. In this case, their action may lead to Bob_i's overestimation of the probability of state transition to 0. As a result, Bob_i may be misled and makes 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 a 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, which means that 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:

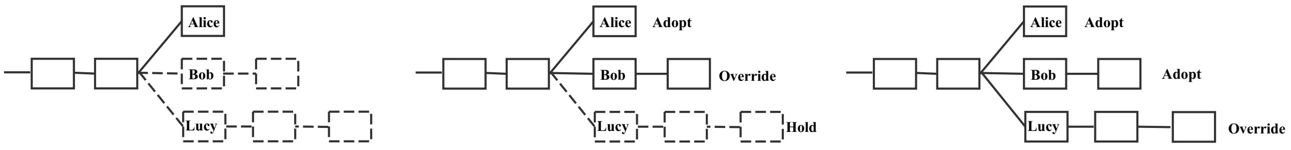
$$r_{tot} = \mathcal{E}[\lim_{n \to +\infty} \sum_{i=1}^{n} r_{a_i}] \tag{3}$$

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

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

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

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

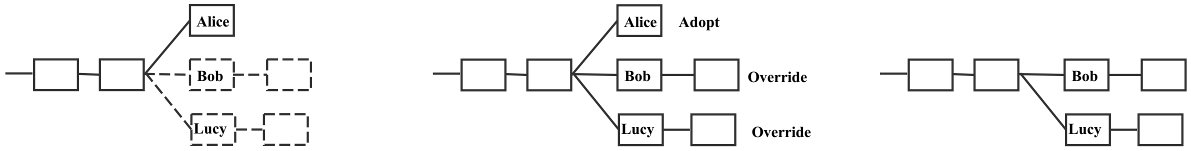


Step 1: Alice publishes her block

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: Both Bob and Lucy take 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.

system's wasted hashpower can be measured by

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

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:

$$Rsbr_i = \frac{sbr_i}{T} \quad (6)$$

The value of $Rsbr_i$ shows the relative efficiency of the i^{th} attacker and with $Rsbr_i < 1$, the i^{th} miner waste a less portion of hashpower than others and his aim of increasing the portion of his blocks in the 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 the mining of $block_i$ and $block_{i+1}$ as one round. An interesting fact in a blockchain with multiple attackers is that the attackers' state keeps changing in one round. As a result, the attacker's action varies in one round.

First, we recall the process of state transition in the model with only one attacker. For an attacker, if he uses selfish mining strategy, he 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 multiple attackers, the decision-making process of the attackers seems like an auction, which means the state transition for the attacker Bob_i will be confirmed only if no new blocks are published in this round. In this case, his action will not change anymore. In one round, action Match and Override result in publishing of new blocks, but only action override changes the length of blockchain.

To make it clearer, 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 $STATE_{Bob}$ = 2 and $STATE_{Lucy}$ = 3 and their actions are both hold at this moment. When Alice finds and publishes 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 time, Lucy's state is 0 with action hold and Bob's state is 0 with action Adopt. After Bob's state converts to 0, in this round, no blocks will be published anymore and the state of all miners is finally fixed. 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 have some results 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. The attacker then takes the action Match and releases one block to catch up the honest

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)

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
```

miners' chain. If the attacker's action is Override or Adopt, the competition will not exist since either the honest miner or the attacker gives up and accepts the opponent's chain. In our blockchain model with multiple attackers, an unexpected competition occurs. 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 simplify the model.

- The starter of one round is always the honest miner or the attacker who takes the action Publish: Once an attacker adopts strategies like selfish mining, 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 he 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.

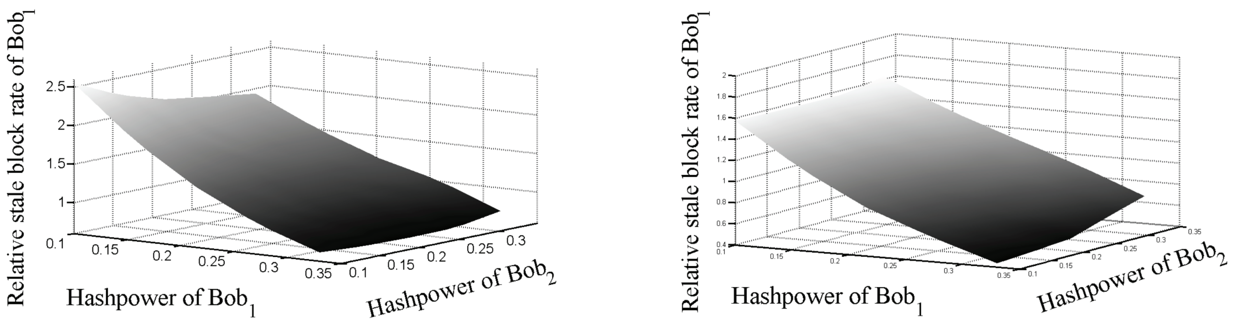
Based on these three facts, the process of determining γ_i can be divided into two steps:

- 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.



$\gamma_h = 1$.

$\gamma_h = 0.5$.

Figure 4: The relative stale block rate of attacker Bob_1

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 when 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 whether to launch a selfish mining attack according to the value of γ_h.

Algorithm 2 Selfish Mining

```
Initialization
while Mining do
    if MyPoolFound then
        if prev == 0 and PrivateChain == 2 then
            Publish the block
        else
            Action hold
        end if
    else
        if prev == 0 then
            Action Adopt
        else if prev == 1 then
            Action Match
        else if prev == 2 then
            Action Override
        else
            Publish the first unpublished block
        end if
    end if
end while
```

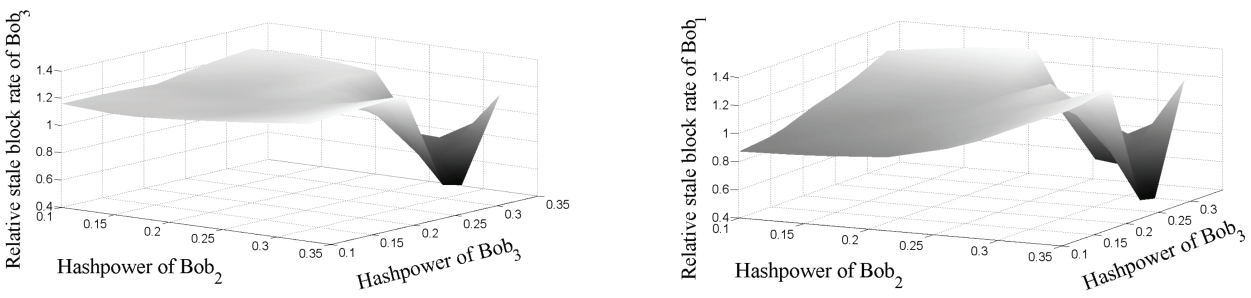
Figure4 shows the relative stale block rate of Bob_1 when the number of attackers is 2 and the value of γ_h is 1 and 0.5 respectively. When $\gamma_h = 1$, we focus on a specific value of Bob_1's hashpower —-0.33, which is the threshold for Bob_1 to gain extra revenue when there is only one attacker. As we can observe from the simulation result, for Bob, the threshold is no longer 1/3. Instead, the threshold for Bob_1 to gain extra revenue is determined by the hashpower of Bob_2. When $\gamma_h = 0.5$ and the hashpower of Bob_2 is relatively small (typically less than Bob_1), the threshold of Bob_1 is less than 0.25, and it can even drop to 0.20. With the increment of Bob_2's hashpower, the threshold for Bob_1 also increases. As suggested in the simulation result, when Bob_2's hashpower is higher than 0.3, the threshold for Bob_1 will be larger than 0.25.

Algorithm 3 Stubborn-n

```
Initialization
while Mining do
    if MyPoolFound then
        if prev == 0 and PrivateChain == 2 then
            Publish the block
        else
            Action hold
        end if
    else
        if prev == -n then
            Action Adopt
        else if prev > -n and prev <= 0 then
            Action hold
        else if prev == 1 then
            Action Match
        else if prev == 2 then
            Action Override
        else
            Publish the first unpublished block
        end if
    end if
end while
```

The threshold is also determined by the hashpower of Bob_2. Even if the hashpower of Bob_1 reaches 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 multiple 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 tends to launch an attack with strategy selfish



γ_h **= 1**

γ_h **= 0.5**

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

mining, he should not only consider the value of γ_h but also the amount of his opponents and his opponents' hashpower.

Result 1: The threshold for an attacker to gain extra revenue drops since the hashpower of his opponents is 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. Recall that in Figure4, when the number of attackers is two and the attackers' hashpower is 0.2, Bob_1 cannot gain extra revenue when γ_h is 0.5 or 1. Figure5 shows the simulation result with more than two attackers, where we set the hashpower of Bob_1 as a constant 0.2. The other two attackers' hashpower is ranging from 0.1 to 0.33. Bob_1 still has the chance to gain extra revenue, though under most of the circumstance, with 20% of hashpower, it is unwise for Bob_1 to launch a selfish mining attack.

*4.1.2 A strategy set: Stubborn-*n. Under most circumstances, when an attacker's private chain falls behind the honest miner's chain, the attacker usually takes action adopt and adopts the honest miner's chain due to the hashpower difference between the attacker and the honest miner. 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.

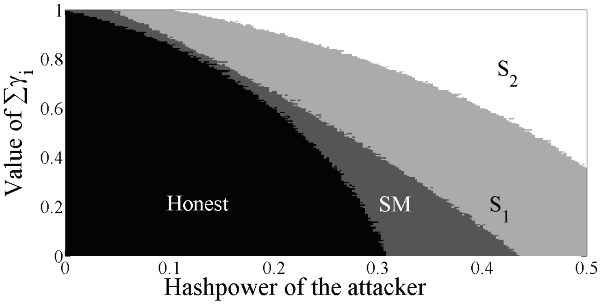


Figure 6: Dominant strategy for different value of β and γ_h when there is one attacker.

S_n is a strategy set, in which n represents the persistent degree of the attacker. If an attacker Bob takes the strategy S_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 S_n with the value of n = 0. To avoid ambiguity, the default value of n is greater than 0 when strategy S_n is mentioned in this paper. The behavior of attacker with strategy S_n is illustrated in Algorithm3.

Consider the case where 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.

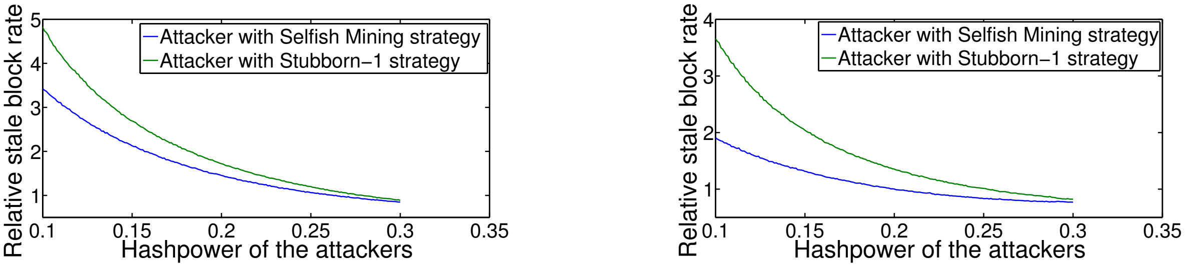
Figure6 is the simulation result when there is only one attacker. When a strategy has a lowest relative stale block rate compared with other strategies in the strategy space, we can make the conclusion that it outperforms other strategies. The regions in the result indicate which strategy outperforms 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 S_n has a lower relative stale block rate than selfish mining in the parameter space where hashpower of the attacker is high in the case of one attacker. The performance of S_n indicates that, compared with selfish mining it wastes more hashpower of the honest miner.

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.

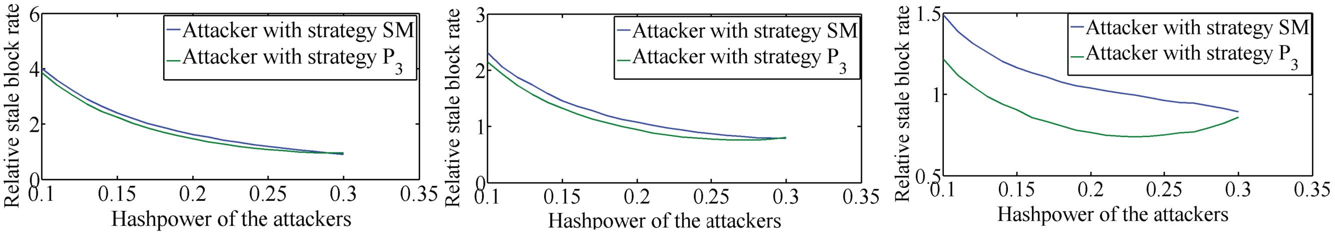
He has another option: Apply the strategy set P_n, denoted by P_n. This strategy is originally proposed by us in this paper. The value of n can be regarded as an alarm the attacker set for his state. When his state reaches n, he will either publish the first block of his private chain or take the action 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.



$\gamma_h = 1$

$\gamma_h = 0.5$

Figure 7: Comparison between an attacker with strategy S_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 P_3 and another attacker with strategy Selfish mining

Different from Algorithm2 and Algorithm3, Algorithm4 contains more detials including how the parameters change when a new block is found.

P_n can be considered as a combination of selfish mining and honest mining, when the attacker reaches state n, he acts like an honest miner if he finds the next block while he acts like a selfish miner if honest miners publish a new block first. Note that, the behavior of P_1 is equivalent to the honest miner and P_2 is similar to a selfish miner. When talking 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 notice that an attacker with strategy P_n will publish his block voluntarily when he reaches state n. This feature of P_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 The simplest case with two attackers

First, we test the performance of different mining strategies in the simplest case which has two attackers with different attacking strategy and one honest miner. In this caes, the attackers will not only compete with the honest miner, but also compete with each other. Their mining performance is evaluated by their relative stale block rate.

5.1.1 *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 when 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 S_1, which means that selfish mining is a more efficient strategy when

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
```

there are more than one attacker in the blockchain model. When the hashpower of both attackers grows, S_1 narrows the gap.

Another fact observed from the simulation result is that with the decrement of the value of γ_h, the gap between S_1 and selfish mining increases. This indicates that strategy selfish mining receives more support from the honest miner, if the honest miner can be easily influenced.

Result 4: When γ_h = 1, the relative stale block rate of an attacker with strategy selfish mining is 40% lower than the attacker with strategy S_1 with the attackers' hashpower set as 0.1. When their hashpower increases to 0.3, the relative stale block rate of selfish mining is only 4.3% lower than S_1.

Generally speaking, in a blockchain with multiple attackers, the hashpower of the attackers is more separated. Under this situation, S_n is suboptimal compared with selfish mining. S_n fits the situation where the hashpower of the attacker is higher.

Result 5: When γ_h = 0.5, the relative stale block rate of selfish mining is 89% lower than S_1 with the attackers' hashpower set as 0.1. When their hashpower increases to 0.3, the relative stale block rate of selfish mining is 12.1% lower than S_1.

The decrement of γ_h benefits attackers with strategy selfish mining instead of S_1. In addition, the attacker with strategy S_n is more persistent on his private chain as the value of *n* increases, therefore he will get less support from the honest miner. We finally come to the conclusion that selfish mining outperforms S_n when multiple attackers launch attacks at the same time.

5.1.2 *Publish-*n *against selfish mining.*

Strategy P_n does not fit the blockchain model with only one attacker. In a proof of work blockchain with multiple attackers, an attacker should not only consider wasting his opponents' computation power but also earning the honest miner's support. The failure of strategy S_1 gives a full illustration of this point.

In this 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.

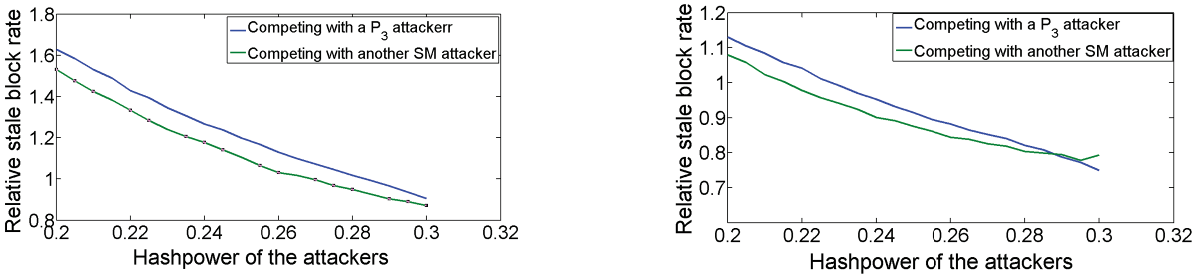
In Figure8, the relative stale block rate of an attacker with strategy P_3 is lower than the attacker with strategy selfish mining when the hashpower of both attackers are low. When the hashpower increases, the performance of selfish mining narrows the gap and eventually it outperforms P_3. Another fact 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 γ_h = 1, the efficiency of P_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 γ_h = 0.2, the efficiency of P_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 P_n has lower relative stale block rate than selfish mining. With the increment of the attackers' hashpower, selfish mining eventually outperforms P_n. When the value of γ_h drops, the honest miners are more likely to accept the chain published by the attackers and gap between the two different strategies grows larger.

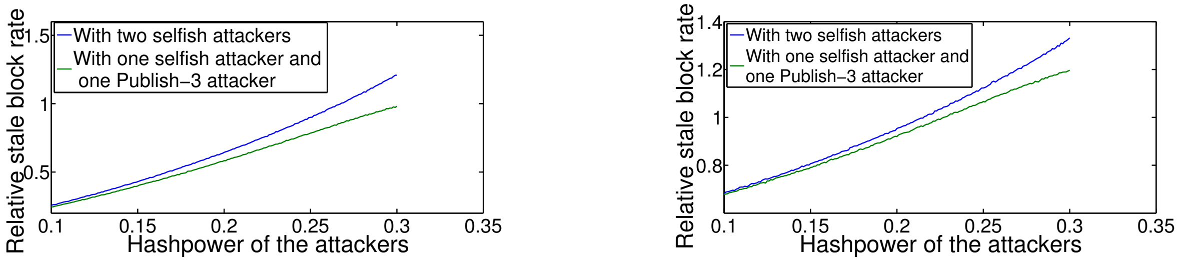
Figure9 compares the relative stale block rate of the selfish miner in the blockchain model with two selfish miners and the blockchain model with one selfish miner and one P_3 miner. The simulation result indicates that the selfish miner in the model with one selfish miner and one P_3 miner always earn less revenue.



$\gamma_h = 1$

$\gamma_h = 0.5$

Figure 9: Comparison between one selfish miner competing with another selfish miner or with a P_n miner.



$\gamma_h = 1$

$\gamma_h = 0.5$

Figure 10: Honest miner's relative stale block rate

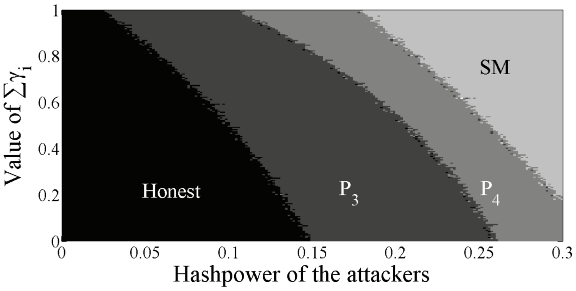


Figure 11: Dominant strategy for different value of β and γ_h with 3 attackers

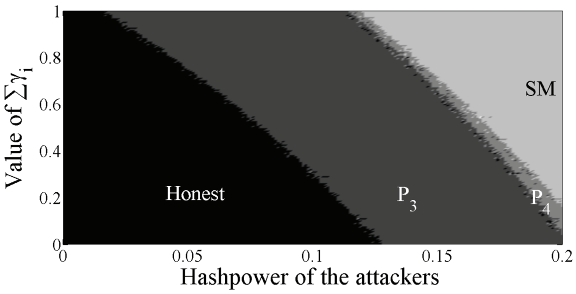


Figure 12: Dominant strategy for different value of β and γ_h with 5 attackers

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 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.

5.2 The case with more attackers

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 increment of the number of attackers will lead to a complicated mining circumstance and the decrement of the hashpower of the honest miner. Thus, in this section, we will not assume that the honest is the majority. The hashpower of the honest miner is in a wider range from 0 to 1.

First, we consider the case that 3 attackers adopt strategy P_i, P_j and selfish mining respectively. The fact is 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 a certain mining strategy that has the best performance when given 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 is low, while selfish mining outperforms other strategies when the hashpower of the attackers is high.

Then strategy S_1 and S_2 are added and the number of attackers is 5. Although the simulation result in Figure7 has proved that strategy S_n do not perform well when there are several attackers, they are still involved to observe their impact on other strategies.

Figure12 is the simulation result. As expected, there is no region for strategy S_1 and S_2. The greatest difference between Figure12 and Figure11 is that the region for strategy P_4 almost disappears. Actually, in this simulation, the efficiency of P_3, P_4 and SM is very 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, yet owning the information of the stale block rate of each miner or mining pool is not enough to find out 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 the Bitcoin community deploys monitors to monitor the behaviors of miners due to the discovery of strategies to earn extra revenue in mining.

Another explanation is based on our simulation result. For a miner with a low computation power, typically less than 20%, he can barely gain extra revenue even if there are three attackers in the blockchain. This means that, under most circumstances, he cannot earn extra revenue compared with honest mining strategy. 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 honest mining strategy. Other miners will soon be aware of the fact that someone has launched an attack according to the rising stale block rate. According to Result 8, when other miners take strategy P_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

S_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 S_n soon loses its advantage. We draw the conclusion that strategy S_n pays too much focus on wasting his opponents' computation power. While being stubborn, it is difficult for an attacker to get support from the honest miners.

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 multiple 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 P_n strategy succeeds in the competition of multiple attackers when the computation power of the attacker is low. But strategy P_n also has a side effect: The attacker wastes less computation power of his opponents. When the computation power of the attacker rises, this side effect becomes even more significant.

ACKNOWLEDGMENTS

This work is supported by Chinese National Research Fund (NSFC) No. 61702330, Chinese National Research Fund (NSFC) Key Project No. 61532013 and National China 973 Project No. 2015CB352401.

REFERENCES

[1] Samiran Bag, Sushmita Ruj, and Kouichi Sakurai. 2017. Bitcoin block withholding attack: Analysis and mitigation. *IEEE Transactions on Information Forensics and Security* 12, 8 (2017), 1967–1978.

[2] Christian Decker and Roger Wattenhofer. 2013. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on.* IEEE, 1–10.

[3] Ittay Eyal. 2015. The miner's dilemma. In *Security and Privacy (SP), 2015 IEEE Symposium on.* IEEE, 89–103.

[4] Ittay Eyal and Emin GÃijn Sirer. 2013. Majority Is Not Enough: Bitcoin Mining Is Vulnerable. 8437 (2013), 436–454.

[5] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* ACM, 3–16.

[6] Arthur Gervais, Hubert Ritzdorf, Ghassan O Karame, and Srdjan Capkun. 2015. Tampering with the delivery of blocks and transactions in bitcoin. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security.* ACM, 692–705.

[7] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse Attacks on Bitcoin's Peer-to-Peer Network.. In *USENIX Security Symposium.* 129–144.

[8] Ghassan O Karame, Elli Androulaki, and Srdjan Capkun. 2012. Double-spending fast payments in bitcoin. In *Proceedings of the 2012 ACM conference on Computer and communications security.* ACM, 906–917.

[9] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. 2017. Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.* ACM, 195–209.

[10] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S Rosenschein. 2015. Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 919–927.

[11] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *Consulted* (2008).

[12] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. 2016. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on.* IEEE, 305–320.

[13] Meni Rosenfeld. 2014. Analysis of hashrate-based double spending. *arXiv preprint arXiv:1402.2009* (2014).

[14] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security.* Springer, 515–532.

[15] Yonatan Sompolinsky and Aviv Zohar. 2015. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security.* Springer, 507–527.

[16] Economist Staff. 2016. Blockchains: The great chain of being sure about things. *The Economist. Retrieved* 18 (2016).