BSM-Ether: Bribery Selfish Mining in Blockchain-based Healthcare Systems

Yilei Wang[a,b], Zhaojie Wang[a], Minghao Zhao[c,d], Xueyang Han[e], Huiyu Zhou[f], Xiaoying Wang*[g], Arthur Sandor Voundi Koe*[h]

[a]*School of Computer Science, Qufu Normal University, China*
[b]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, China*
[c]*School of Software, Tsinghua University, China*
[d]*School of Computing, National University of Singapore, Singapore*
[e]*Beijing Ant Cloud Financial Information Service Co., Ltd., China*
[f]*School of Computing and Mathematical Sciences, University of Leicester, United Kingdom*
[g]*Third Affiliated Hospital of Sun Yat-sen University, China*
[h]*Institute of Artificial Intelligence and Blockchain, Guangzhou University, China*

Abstract

Blockchain-enabled distributed networks present a new infrastructure for building reliable and privacy-preserving healthcare utilities. Among them, Ethereum networks have gained specialized attention for their high usability and security. In this paper, we raise the awareness of Ethereum's vulnerability due to selfish mining, in which ill-disposed miners are feasible to receive more rewards than honest ones. To demonstrate this, we compose a new bribery selfish mining scheme, the BSM-Ether, targeted to Ethereum. The BSM-Ether attack can be easily conducted and has higher rewards than other existing malicious attacks. Besides, we present a formal and rigorous analysis of multiple kinds of rewards from the BSM-Ether attack. Simulation experiments show the high effectiveness of BSM-Ether – the attacker can get a higher revenue with few efforts. To tackle this variability, we present some implementation level proposals and suggestions for building healthcare systems on top of Ethereum to minimize the negative effect of the security deficiency of underlying systems.

Email addresses: wang_yilei2019@qfnu.edu.cn (Yilei Wang), wangzhaojie82@gmail.com (Zhaojie Wang), mh-zhao17@mails.tsinghua.edu.cn (Minghao Zhao), warm3snow@gmail.com (Xueyang Han), Hz143@leicester.ac.uk (Huiyu Zhou), xiaoying.wxy@outlook.com (Xiaoying Wang*), 2517482859@qq.com (Arthur Sandor Voundi Koe*)

Keywords: Blockchain, Bribery selfish mining, Ethereum, Healthcare system

1. Introduction

Ethereum is an open-sourced platform of public blockchains embedded with smart contracts, where Ethereum Virtual Machines are utilized to deal with the contracts [1]. Smart contract is Turing complete, which allows us to put the data on chain in healthcare. For example, the medical records and sheets can be processed by smart contracts, which realize data sharing among patients, doctors and hospitals. Meanwhile, the decentralized features, including consensus mechanism and anonymity guarantee data consistent, tamper proof and privacy [2, 3, 4, 5, 6]. In fact, however, the security of Ethereum networks may not be as good as expected in the real world, which actually may devastate the functionality and security of the healthcare systems. Selfish mining is such an attack, which leverages the vulnerabilities in a consensus mechanism [7, 8, 9, 10]. In Bitcoin, such attacks are widely studied [11, 12, 13, 14, 15, 16, 17, 18]. Recently, the attacks of selfish mining against Ethereum have received large attentions due to the widely Ethereum-based applications. Existing works indicate that Ethereum is more fragile against selfish mining attacks than those in Bitcoins [19, 20, 21]. That is, selfish mining in Ethereum has a larger impact than that in Bitcoins. Although there are lots of methods against such attacks [22, 23, 24], attackers, especially intelligent ones, update their attacking actions to breach the securities for Ethereum-based applications [25, 26, 27, 28, 29]. Additionally, the attacker may threaten the Ethereum mobile client by choosing a suitable attacking algorithm [30, 31].

1.1. Related Works

Eyal and Sirer propose the notion of selfish mining (SM1), where the attack can improve the rewards (e.g. double spending) only by controlling 0.25 mining power of the whole system. This threshold is far less than the traditional one 0.51. Sapirshtein et al. propose ϵ-optimal selfish mining, which

reduces the threshold to 0.2321 [8]. On the other hand, selfish mining can also be implemented in Ethereum. Ritz and Zugenmaier quantify the rewards, in Ethereum, considering the uncle blocks with Monte Carlo. They reduce the threshold from 0.25 to 0.185($\pm$0.012) [20]. The following works prove that the threshold for selfish mining in Ethereum can be further lower to 0.163 due to the uncle blocks [19]. Yang et al. quantify the impact of selfish mining on imperfect blockchain networks [32]. They construct Markov models for selfish mining in Bitcoin and Ethereum respectively. Recently, machine learning has become hot topic [33, 34] and has been applied to reducing the threshold in selfish mining [12, 35]. In [35], they propose intelligent bribery selfish mining (IPBSM) by combining selfish mining and bribery attacks. They optimize the rewards and reduce the threshold by utilizing machine learning. Garay applies bribery attacks in smart contracts for the sake of fairness in business [36]. Bribery can be also utilized in rational smart contract [37] to resist collusion in cloud computing [38, 39, 40, 41]. quantifies the bribery attack and compare with other attacks, which improve the rewards for the attacker. The series of security problems are mainly caused by the vulnerability of the consensus mechanism. Then Saad et al. [42] perform a comprehensive analysis of the PoW consensus mechanism. They show that execution of the consensus is asynchronous, which is the main reason for potential attacks. Gervais et al. [43] introduce a novel quantitative framework to analyse the security and performance of PoW blockchains and devise optimal adversarial strategies for double-spending and selfish mining.

1.2. Motivation and Contribution

Previous works indicate that the threshold of selfish mining in Ethereum is lower than that of SM1 [19]. In Ethereum, it is easier for attackers to launch selfish mining attacks. On the other hand, compared with SM1 and ϵ-optimal, the threshold of IPBSM [35] is 0.075 with higher rewards. That is, bribery selfish mining attacks have strong damages. Currently, most applications are implemented in Ethereum due to the Turing completeness. However, as mentioned above, the works stress on selfish mining attacks on Bitcoin instead of Ethereum.

Consequently, the attacks such as selfish mining against Ethereum are neglected. Therefore, it is urgent to focus on the attacks against the Ethereum to improve the security level for the Ethereum-based applications. In this paper, we propose a novel BSM-Ether attack against the Ethereum. The relative rewards are analyzed and the simulation results demonstrate that BSM-Ether performs well in terms of threshold and rewards. Our contributions are threefold.

1. We propose a novel selfish mining: BSM-Ether by combining selfish mining and bribery attacks in the Ethereum. Further, we present BSM-Ether as a state machine by describing the components in MDP.
2. We analyze the regular rewards, uncle rewards and nephew rewards by deriving necessary equations. In addition, we also consider the impact of bribes on the relative rewards for the attacker and the rational miners.
3. The simulation results indicate that the threshold of BSM-Ether is only 0.067, which is lower than other attacks. Meanwhile, we also simulate the impact of BSM-Ether on the uncle rewards and nephew rewards. We also present some suggestions for the blockchain-based healthcare system against BSM-Ether.

1.3. Roadmap of This Paper

Section 2 presents the process of BSM-Ether, which is then formally described by MDP. More specifically, we define the sets of states and actions, transition matrixes, and the rewards. In Section 3, a simplified MDP is presented, based on the rewards discussed in details. The total rewards consists of regular rewards, uncle rewards, nephew rewards and bribes. Afterwards, we analyze these rewards respectively and list the detailed deductions in Section 4. Simulations are illustrated in Section 5 with respect to the relative rewards, uncle rewards and nephew rewards. We compare BSM-Ether with SM1, IPB-SM and honest mining, the results of which indicate that BSM-Ether achieves higher relative rewards with a lower threshold. That is, BSM-Ether is easily implemented in Ethereum with more damages.

2. Bribery Selfish Mining in Ethereum

In this section, we first explain parameters in the BSM-Ether, facilitating to better describe the actions and rewards for the parties in the system.

2.1. The Terminologies of BSM-Ether

Following [7, 19, 35], the terminologies and parameters are given below.

1. **Participants.** The participants in BSM-Ether consist of the attacker and a group of rational miners (the rational miners for short), who are profit-oriented without attacking. The rational miners behave honestly most of the time, but when certain actions can increase their rewards, they will deviate from honest behaviors to maximize their rewards.
2. **Mining power**. We standardize the computational power of the network to 1. Similar to [7, 19], the attacker's power is α ($0 \leq \alpha \leq 0.45$), and the rational miners' power is $1 - \alpha$. We assume that the group of rational miners behave individually without collusion. Otherwise, the group of rational miners may launch 51% attack.
3. **Round.** Participants mine blocks according to their mining powers. Each time a participant mines a block and appends it to the chain (regardless of public or private chain), we call it a round. In general, only one party (either the attacker or one rational miner) takes one action in each round, such as appending a block to the public or private chain. However, there are exceptions. For example, when a rational miner appends one new block to the public chain, the attacker may immediately publish the oldest block of the private chain to generate a fork. In this situation, we call the whole process one round, where one rational miner first takes action and then the attacker takes action. For brevity, we call the round unilateral round when only one participant takes action in one round and bilateral round when both one rational miner and the attacker take actions alternatively.
4. **Forks.** When two blocks are simultaneously appended to the same block in the previous round, a *natural* fork will occur. Although many natural

forks in Ethereum, we do not focus on them. Therefore, we neglect the natural forks in this paper. In fact, we only consider the forks generated by the attacks. If we say that there is a fork, generally speaking, this is from the perspective of rational miners. Because the attacker always follows the private chain, there is no fork towards the view of the attacker.

5. **Lead racing.** When there is a fork, there will be two situations: the length of the private chain is 0, and the length of the private chain is not 0, called "0-lead" racing and "$\overline{0}$-lead" racing respectively (ref. Fig. 1).
6. **Branch.** Towards the view of the rational miners, who are not aware of the existence of the private chain, each fork consists of two branches (ref. Fig. 1) [1]: $B_{attacker}$ and $B_{rational}$. The latest block in the former branch is mined by the attacker and the latter is mined by the rational miners. Generally when there is a fork, the rational miners choose to mine on the attacker's branch with probability γ and the public chain with probability $1-\gamma$.
7. **Bribes.** The rational miners may mine on the attacker's branch when the attacker bribes them with probability β since the rational miners may obtain additional bribes other than the rewards from mining blocks. Note the attacker may bribe the rational miners by increasing the mining rewards in branch $B_{attacker}$ or by bribing them off-line (e.g. the attacker gives the one, physical currencies if they follow the branch $B_{attacker}$).
8. **State.** For the rational miners and the attacker, the views of the blockchain structure are distinct due to the private chain. That is, the rational miners are not aware of the private chain. For instance, as for "$\overline{0}$-lead" racing, toward the rational miners' view, there is a fork with two equal-length branches $B_{attacker}$ and $B_{rational}$. While towards the attacker's view, there is a private chain with positive length. We call such views the states of the block. Further, the states for the rational miners are simple, which include two states: fork and no fork. The states for the attacker are a little

[1] Note that we only consider the scenario, where only two branches in a fork.

bit complex, which should reflect the length of the private chain. Recall that the natural forks are beyond the consideration in this paper. There is no forks towards the view of the attacker.

9. **Main chain.** In this paper, the public chain denotes the public views for both the rational miners and the attacker. Note the public chain may include more than one branch. The main chain denotes the public chain with the longest branch.

10. **The difference.** The length of the private chain is counted from the first private block while the length of the public chain is counted from the parallel block with the private chain (if there are any). Otherwise, the length of the public chain is considered to 0. Let l_a be the length of the attacker's private chain, l_h the length of public chain, $\Delta = l_a - l_h$ denotes the difference between the private and public chains. For example, the length of the private chain is 0 denotes that there is no private chain. The length of the main chain is 0 denotes that, in the public chain, no blocks parallels with the private chain.

11. **Uncle and nephew blocks.** The rewards after mining a regular block is normalized to 1. All miners can cite the uncle blocks in the public chain and obtain the corresponding rewards. However, the blocks in the private chain (private blocks) fail to be cited as uncle blocks since the private blocks are invisible for rational miners. On the other hand, these private blocks will be published later and become uncle blocks or regular blocks in the public chain. However, it's too late for these blocks to be cited by other blocks. That is, the private blocks can cite other uncle blocks but fail to be cited as uncle blocks. In Ethereum, the rewards also consist of uncle and nephew rewards defined in Equation (1), where $k_u(l)$ and $k_n(l)$ denote the uncle and nephew reward respectively. l denotes the length

difference between the uncle block and the nephew block.

$$k_u(l) = \begin{cases} (8-l)/8, & 1 \leqslant l \leqslant 6; \\ 0, & otherwise. \end{cases}$$
$$k_n(l) = \begin{cases} 1/32, & 1 \leqslant l \leqslant 6; \\ 0, & otherwise. \end{cases} \quad (1)$$

12. **Rewards and revenue.** Before section 3, we do not distinguish between reward and revenue, they are both general concepts, representing the profits of participants. After section 3 we distinguish between them, where revenue only refers to the profits of participants in each round, and reward refers to the sum of all revenue.

3. The MDP of BSM-Ether

The attacker and the rational miners alternatively take actions, hoping to get rewards by mining new blocks. Recall that the main target of the attacker to achieve additional rewards disproportional to the mining power. Markov Decision Process (MDP) is a mathematical model of sequential decision, which is used to simulate the random strategies and rewards of multiple agents [44, 45]. Specifically, the agents first perceive the current state of the system, and implement actions according to their strategies, thereby changing the state of the environment. As mentioned above, MDP is carried out round by round, where actions consist of unilateral actions and bilateral actions. In each round, each agent will get corresponding revenues through state transition. After multiple rounds of interaction in the MDP process, the sum of the revenues in each round is called the reward. In general, MDP can be used to model reinforcement learning problems in machine learning. By using dynamic programming, random sampling, policy search and other methods [46, 47], MDP can solve agent strategies that maximize rewards. The function of MDP gears to the target of our BSM-Ether attack. Therefore, we rehearsal our BSM-Ether towards the view of MDP.

Generally, MDP contains a set of interactive objects, namely agents and environment. In this paper, the agents are the attacker and the rational miners and the environment is the blockchain of Ethereum. MDP consists of 5 ingredients: the states set $S = \{s_0, s_1, s_2, ...\}$, the action set $A = \{a_1, a_2, ...\}$, the strategies $p(s_{t+1}|s_t, a_t)$, the revenue $R_t = R(s_{t+1}|s_t, a_t)$ and the rewards $G = \sum_{t=0}^{N} R_t$. In the following, we present these ingredients.

3.1. The States in MDP

Recall that in SM1 and other similar attacks towards Bitcoin. The states, denoting as $< l_a, l_h >$, only include length of the public and the private chain. However, in Ethereum, we also consider the forks due to the citations between uncle and nephew blocks. Therefore, the states set $S = \{s_0, s_1, s_2, ...\}$ are denoted as $s_t =< (l_a)_t, (l_h)_t, forkable_t >$. $forkable_t \in \{nofork, fork, fork_b\}$ is an indicator denoting, in the t^{th} round, whether there is a fork, and if there is a fork, whether the attacker bribes the rational miners on the fork. That is, $nofork$ denotes there is no fork, $fork$ denotes a fork without bribery and $fork_b$ denotes a fork with bribery. For example, the sate $< 2, 0, fork_b >$ of the t^{th} round denotes that the length of the private and the public chain is 2, 0 respectively. Further, there is a fork towards the view of the rational miners and the attacker bribes the rational miners to follow the $B_{attacker}$. For simplicity, let $s_0 =< 0, 0, nofork >$ be the initial states. In addition, let the maximize Δ be 5 since the probability α^6 of leading 6 blocks is negligible, especially when α is small.

Fig. 1 presents the structure of states $< 1, 1, fork >$ and $< 2, 0, fork_b >$. The circles denote the blocks mined by the rational miners and the rectangles denotes the blocks mined and published by the attacker. The rectangles with dotted lines denote the blocks of private chain, which are mined by the attacker without publishing in the public chain. The value b in rectangles denotes the bribes for the rational miners. Further, we present the forks and branches.

The simplified states and MDP model. The format of state $< l_a, l_h, forkable >$ is not concise, especially when drawing the flow of the MDP model. So, in this

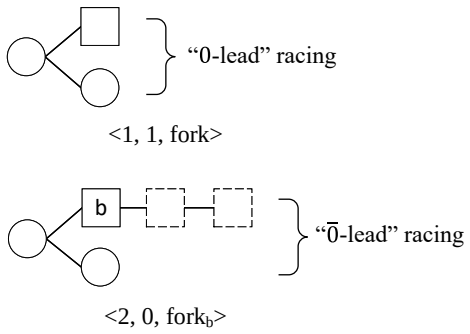


Figure 1: The examples of "0-lead" racing and "$\bar{0}$-lead" racing.

paper, we denote the states by following the common format in selfish mining (e.g. the difference of the private and public chain). Meanwhile, we should also reflect the forks and bribes in the simplified format of the states. Let $\Delta^{[']}_{[b]}$ is the general format for the simplified states, where Δ is defined in section 2.1, [] denotes an option. More specifically, $'$ and b denote the value of $forkble$ is $fork$ and $fork_b$ respectively. There are altogether 16 simplified stated in BSM-Ether according to the parameters in section 2.1, which are listed in Table 1. For example, $2'$ denotes the state $< 2, 0, fork >$ where $\Delta = 2$ and there is a fork towards the view of the rational miners. $3'_b$ denotes the state $< 3, 0, fork_b >$ where $\Delta = 3$ and there is a fork towards the view of the rational miners. Further, the attacker bribes the rational miners once they choose the branch $B_{attacker}$. As mentioned above, 0 is the initial state and 5, $5'$, $5'_b$ are terminal states.

3.2. The Actions in MDP

The participants involved in BSM-Ether choose their actions according to the states. More specifically, the actions of the rational miners depend on whether there is a fork in the main chain. While the action of the attacker depends on the length of the private chain. Note that we do not specifically describe the citations between the uncle and nephew blocks. Instead, we use citations (if any) as their default actions. In the following, we describe the actions in more details.

Table 1: The simplified format of the states.

state	**simplified**	**state**	**simplified**
<0,0,$nofork$>	0	<3,0,$fork$>	$3'$
<1,1,$fork$>	$0'$	<3,0,$fork_b$>	$3'_b$
<1,1,$fork_b$>	$0'_b$	<4,0,$nofork$>	4
<1,0,$nofork$>	1	<4,0,$fork$>	$4'$
<2,0,$nofork$>	2	<4,0,$fork_b$>	$4'_b$
<2,0,$fork$>	$2'$	<5,0,$nofork$>	5
<2,0,$fork_b$>	$2'_b$	<5,0,$fork$>	$5'$
<3,0,$nofork$>	3	<5,0,$fork_b$>	$5'_b$

The actions for the rational miners. As mentioned above, the rational miners behave honestly most of the time. Recall that the states induced by "0-lead" racing and "$\overline{0}$-lead" racing are identical for the rational miners. That is, regardless of "0-lead" racing and "$\overline{0}$-lead" racing, they only see that there are two branches $B_{attacker}$ and $B_{rational}$ on the public chain. Then we present the actions for the rational miners if one of them mines a new block with probability $1-\alpha$.

1. ***rational*$_{nor}$** (no forks): The rational miners append the new block to the main chain when there is no forks.
2. ***rational*$_{pri}$** (forks): There are two subcases according to the existence of bribes.
   (a) The rational miners choose to append the new block to $B_{attacker}$ with probability γ when the attacker does not bribe them with probability $1-\beta$.
   (b) The rational miners choose to append the new block to $B_{attacker}$ when the attacker bribes them with probability β.
3. ***rational*$_{pub}$** (forks): The rational miners choose to mine on $B_{rational}$ with probability $1-\gamma$ without bribery. Note that the rational miners will not mine on $B_{rational}$ when the attacker bribes them.

The actions for the attacker. Recall that the actions of the attacker depend on the length of the private chain. More specifically, they depend on the difference of the private and the public chain. In the sequel, we describe the actions according to different Δ in more details. Note that we do not consider the case where $\Delta < 0$. The reasons is: once a private chain is established, the attacker will follow that private chain. If there is no private chain, the attacker will choose to create a private chain on the main chain. So there will be no case $\Delta < 0$.

1. ***Wait*** ($\Delta \geq 0$ and the attacker appends a new block with probability α): The attacker appends the new block to the private chain without publicly publishing and continue to mine. However, there is an exception when $\Delta = 0$ with a fork ("0-lead" racing), the attacker will immediately publish the block instead of setting up a private chain since otherwise, (s)he may lose the leading position later.
2. ***Match*** ($\Delta \geq 0$, $\Delta \neq 2$ and one rational miner appends a new block with probability $1 - \alpha$): Note that we do not list the case of $\Delta = 0$ since the attacker does not take any actions. The rational miner takes the action of $\boldsymbol{rational_{nor}}$. The attacker publishes the oldest block of the private chain to form a fork (towards the view of the rational miners). Note that the generated fork is "0-lead" racing and "$\overline{0}$-lead" when $\Delta = 1$ and $\Delta > 1$ respectively.

   As mentioned above, ***Match*** is kind of bilateral action, where the attacker immediately publishes the oldest block of the private chain. However, in this paper, the attacker may bribe rational miners. Therefore, there are two actions according to whether the attacker is bribing or not.

   (a) $\boldsymbol{Match_b}$: The attacker puts the bribes on the branch $B_{attacker}$ with probability β. In the next round, a rational miner receives the bribes once (s)he appends a new block to the branch $B_{attacker}$. Recall that the bribes may be paid off-line as mentioned above.

   (b) $\boldsymbol{Match_{nb}}$: The attacker does not puts the bribes on the branch $B_{attacker}$ with probability $1 - \beta$.

3. ***Override*** ($\Delta = 2$ and one rational miner appends a new block with probability $1 - \alpha$): The attacker publishes all two private blocks in private chain and the previous private chain becomes the main chain. The reason why the ***Match*** strategy is not adopted in this case, is because only releasing the oldest block of the private chain is not enough for the attacker to maintain the leading advantage. In other words, if the attacker only publishes the oldest block of the private chain at this time, then the private chain is very likely to be overtaken (because the overall mining power of the rational miner is higher than that of the attacker), so that all blocks in the private chain will be invalid. In order to avoid this situation, the attacker adopts the ***Override*** strategy.

3.3. The Strategies in MDP

The strategy $p(s_{t+1}|s_t, a_t)$ denotes the decision on the available actions when the state is s_t at the t^{th} round. Here the strategy for each round is the distribution on all available actions. That is, the sum of the probabilities on each available action is 1. For example, at the t^{th} round, the attacker has two available actions $Wait$ and $Override$, which may be taken with probability α and $1-\alpha$ respectively. Note that the probability depends on the parameters of α, β and γ (refer to section 2.1). Figure 2, 3, 4 illustrate the strategies with respect to different states.

In the following, we will explain the strategies through a few examples in Figure 2, 3, 4.

State 1. In the t^{th} round, the length of the private and public chain is 1 and 0 respectively. In addition, there is no fork towards the view of the rational miners. There will be two situations next:

1. The attacker mines a block with probability α. This is a unilateral round since only the attacker takes actions in the current round. That is, the attacker takes $Wait$ and appends the new block to the private chain without publishing. Thus the state of the $(t+1)^{th}$ round becomes $< 2, 0, nofork >$.

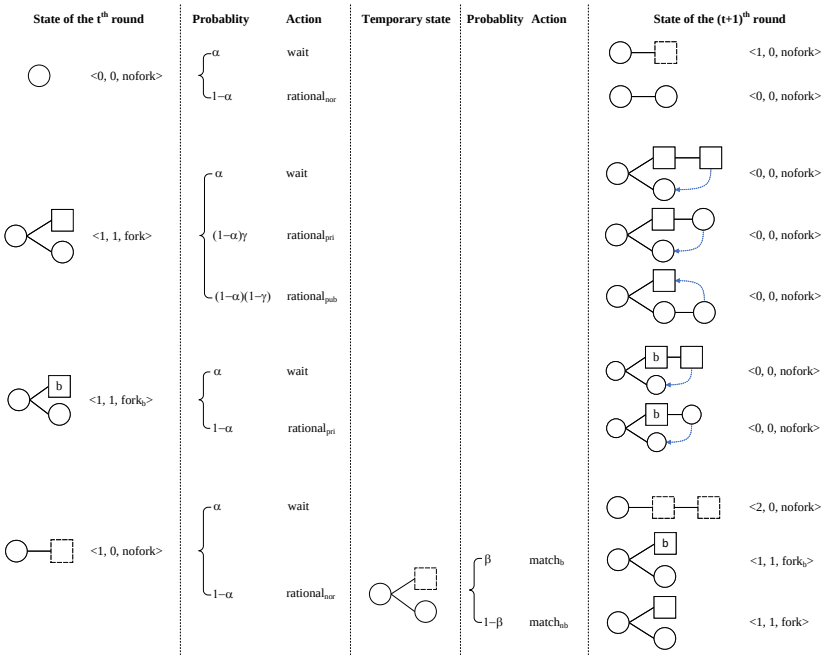


Figure 2: The state transition of state 0, $0'$, $0'_b$, 1. (The dotted arrows denote the citations of the uncle and nephew blocks. Temporary states are listed in bilateral rounds, where both the rational miners and the attacker take actions in one round.)

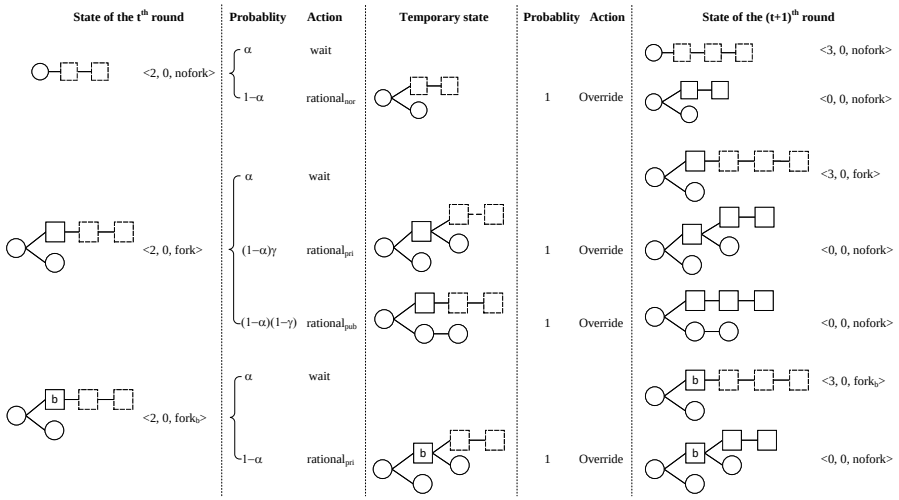


Figure 3: The state transition of state 2, $2'$, $2'_b$.

2. One rational miner mines a block with probability $1-\alpha$. This is a bilateral round since both a rational miner and the attacker take actions in the current round. More specifically, the rational miner first takes $rational_{nor}$ by appending and publishing the new block in the public chain. The state becomes the temporary state, where the length of both the public chain and the private chain is 1. In the sequel, the attacker immediately takes action $Match_b$ and $Match_{nb}$ with probability β and $1-\beta$ respectively.
   (a) For the former action, the attacker publishes the block in the private chain and try to bribe the rational miners when they mine new blocks in the next round, leading to the state $< 1, 1, fork_b >$. Note that we inherit the length of the public and private chain in the temporary state. Further, there is a fork with bribe towards the view of the rational miners.
   (b) For the latter action. the attacker only publishes the block in the private chain without bribing, leading to the state $< 1, 1, fork >$.

State $0'$. As mentioned in section 2.1, in the t^{th} round, there is a "0-lead"

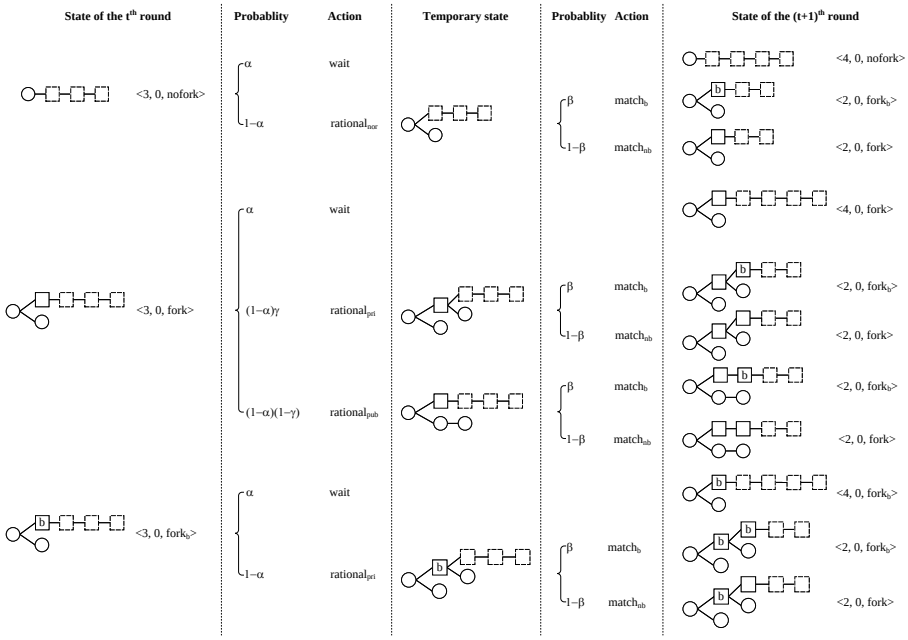


Figure 4: The state transition of state 3, $3'$, $3'_b$.

racing. There will be three situations next:

1. The attacker mines a block with probability α. Then the attacker takes the action *wait*. That is, the attacker publishes and appends the new block to the branch $B_{attacker}$, leading to the state $< 0, 0, nofork >$.
2. One rational miner mines a block with probability $1 - \alpha$. Then there are two sub-cases:
   (a) The rational miner chooses to follow the branch $B_{attacker}$ with probability γ. The rational miner takes action $rational_{pri}$, leading to the state $< 0, 0, nofork >$.
   (b) The rational miner chooses to follow the branch $B_{rational}$ with probability $1 - \gamma$. The rational miner takes action $rational_{pub}$, leading to the state $< 0, 0, nofork >$.

For these three situations, the attacker and the rational miners will cite the block

in the branch $B_{rational}$ or $B_{attacker}$, obtaining the uncle and nephew rewards.

State $2'$. As mentioned in section 2.1, in the t^{th} round, there is a "$\bar{0}$-lead" racing. There will be three situations next:

1. The attacker mines a block with probability α. Then the attacker takes action $wait$, appending the new block to private chain, leading to the state $< 3, 0, fork >$.
2. One rational miner mines a block with probability $1 - \alpha$. Then there are two sub-cases:
   (a) The rational miner chooses to follow the branch $B_{attacker}$ with probability γ.
   (b) The rational miner chooses to follow the branch $B_{rational}$ with probability $1 - \gamma$.

   Then the attacker immediately takes action $Override$, publishing all blocks in the private chain. leading to the state $< 0, 0, nofork >$. Note that there is no citations between the uncle and nephew blocks since the nephew blocks are older than uncle blocks. Thus, the nephew blocks can not cite the uncle blocks when the latter blocks are published.

State $2'_b$. The distinction of state $< 2, 0, fork_b >$ from state $< 2, 0, fork >$ is the bribe on the fork. That is, the rational miners will definitely choose branch $B_{attacker}$ if possible. There will be only two situations next:

1. The attacker mines a block with probability α. The attacker takes action $wait$, leading to the state $< 3, 0, fork_b >$.
2. One rational miner mines a block with probability $1 - \alpha$ and chooses to follow the branch $B_{attacker}$, leading to state $< 0, 0, nofork >$. Note that there is no citation between uncle and nephew blocks.

State $3'$. In the t^{th} round, there is a "$\bar{0}$-lead" racing with a private chain. There will be three situations next:

1. The attacker mines a block with probability α. The attacker takes action $wait$. This action will append the newly mined block to the private chain, leading to the state $< 4, 0, fork >$.

2. One rational miner mines a block with probability $1-\alpha$. Then it's turn for the attacker to take actions. That is, it's a bilateral round. There are two sub-cases.

   (a) If the rational miner chooses to follow the branch $B_{attacker}$ with probability γ, leading to a temporary state. There are still two sub cases.

      i. The attacker takes action $Match_b$ with probability β, leading to the state $< 2, 0, fork_b >$.

      ii. The attacker takes action $Match_{nb}$ with probability $1-\beta$, , leading to the state $< 2, 0, fork >$.

   (b) If the rational miner chooses to follow the branch $B_{rational}$ with probability $1-\gamma$, leading to a temporary state. There are still two sub cases.

      i. The attacker takes action $Match_b$ with probability β, leading to the state $< 2, 0, fork_b >$.

      ii. The attacker takes action $Match_{nb}$ with probability $1-\beta$, leading to the state $< 2, 0, fork >$.

   Note that the sub-cases 2)a)i) and 2)b)i) are identical. Similarly, the sub-cases 2)a)ii) and 2)b)ii) are identical.

With the simplified format, we present the Markov decision process of BSM-Ether in Figure 5. Note that the pentagon denotes the initial state $s_0 = 0$ and the triangle denotes the terminal states 5, $5'$ and $5'_b$. The values on the arrows denote the probability from s_t to s_{t+1}. For example, the probability from state 1 to 2 is α.

The revenue R_t and rewards G: The definitions and evaluations for these two ingredients are much complex. So, we leave the details in Section 4.

4. Analysis of Revenue and Rewards

In this section, we analyze the revenue and rewards for the attacker and rational miners respectively. Note that the revenue, including regular revenue, uncle and nephew revenues, derive from each state transition. In addition, we

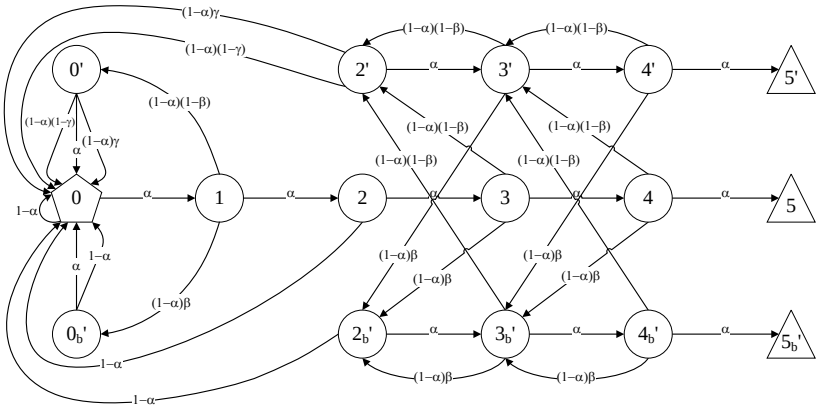


Figure 5: The Markov decision process of BSM-Ether.

should also consider the bribes since it's part of the revenue for the attacker. The rewards are the sum of these revenues. Table 2 presents the revenue for each state transition in MDP.

4.1. The Probability for Each State in BSM-Ether

In fact, the total revenue is an expected one of each state. We should first evaluate the probability $P_{\Delta^{[']}_{[b]}}$ for each state $\Delta^{[']}_{[b]}$. Then the average expectations are calculated for the attacker and the rational miners. In this section, we assume that the probability of the initial state s_0 is fixed. The relationship of each state is deduced through Fig. 5. For example, the state 0 can be reached by state 0 with probability $1-\alpha$, state $0'$ with probabilities α, $(1-\alpha)\gamma$, $(1-\alpha)(1-\gamma)$, state $2'$ with probabilities $(1-\alpha)\gamma$, $(1-\alpha)(1-\gamma)$, state 2 with probability $1-\alpha$ and state $2'_b$ with probability $1-\alpha$. Therefore, the probability of state 0 can be deduced in Equation (2). The derivations of other states are similar to Equation (2).

$$\begin{aligned} P_0 = {} & (1-\alpha)P_0 + P_{0'_b} + P_{0'} + (1-\alpha)P_{2'} + (1-\alpha)P_2 \\ & + (1-\alpha)P_{2'_b} \end{aligned} \tag{2}$$

Table 2: The revenue for each state transition in MDP.

Number	s_t	Probability	s_{t+1}	regular	uncle	nephew	bribe
a-1	0	α	0	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
a-2	0	$1-\alpha$	1	$< 0, 1 >$	$< 0, 0 >$	$< 0, 0 >$	0
b-1	$0'$	α	0	$< 2, 0 >$	$< 0, k_u(1) >$	$< k_n(1), 0 >$	0
b-2	$0'$	$(1-\alpha)\gamma$	0	$< 1, 1 >$	$< 0, k_u(1) >$	$< 0, k_n(1) >$	0
b-3	$0'$	$(1-\alpha)(1-\gamma)$	0	$< 0, 2 >$	$< k_u(1), 0 >$	$< 0, k_n(1) >$	0
c-1	$0'_b$	α	0	$< 2, 0 >$	$< 0, k_u(1) >$	$< k_n(1), 0 >$	$-b$
c-2	$0'_b$	$1-\alpha$	0	$< 1, 1 >$	$< 0, k_u(1) >$	$< 0, k_n(1) >$	$-b$
d-1	1	α	α	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
d-2	1	$(1-\alpha)\beta$	$0'_b$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
d-3	1	$(1-\alpha)(1-\beta)$	$0'$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
e-1	2	α	3	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
e-2	2	$1-\alpha$	0	$< 2, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
f-1	$2'$	α	$3'$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
f-2	$2'$	$(1-\alpha)\gamma$	0	$< 3, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
f-3	$2'$	$(1-\alpha)(1-\gamma)$	0	$< 3, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
g-1	$2'_b$	α	$3'_b$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
g-2	$2'_b$	$1-\alpha$	0	$< 3, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
h-1	3	α	4	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
h-2	3	$(1-\alpha)\beta$	$2'_b$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
h-3	3	$(1-\alpha)(1-\beta)$	$2'$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
i-1	$3'$	α	$4'$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
i-2	$3'$	$(1-\alpha)\beta$	$2'_b$	$< 1, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
i-3	$3'$	$(1-\alpha)(1-\beta)$	$2'$	$< 1, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
j-1	$3'_b$	α	$4'_b$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
j-2	$3'_b$	$(1-\alpha)\beta$	$2'_b$	$< 1, 0 >$	$< 0, 0 >$	$< 0, 0 >$	$-b$
j-3	$3'_b$	$(1-\alpha)(1-\beta)$	$2'$	$< 1, 0 >$	$< 0, 0 >$	$< 0, 0 >$	$-b$
k	4	α	5	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
l	$4'$	α	$5'$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0
m	$4'_b$	α	$5'_b$	$< 0, 0 >$	$< 0, 0 >$	$< 0, 0 >$	0

$$P_{0'} = (1-\alpha)(1-\beta)P_1 \tag{3}$$

$$P_{0'_b} = (1-\alpha)\beta P_1 \tag{4}$$

$$P_1 = \alpha P_0 \tag{5}$$

$$P_2 = \alpha P_1 \tag{6}$$

$$P_{2'} = (1-\alpha)(1-\beta)(P_3 + P_{3'} + P_{3'_b}) \tag{7}$$

$$P_{2'_b} = (1-\alpha)\beta(P_3 + P_{3'} + P_{3'_b}) \tag{8}$$

$$P_3 = \alpha P_2 \tag{9}$$

$$P_{3'} = (1-\alpha)(1-\beta)(P_4 + P_{4'} + P_{4'_b}) + \alpha P_{2'} \tag{10}$$

$$P_{3'_b} = (1-\alpha)\beta(P_4 + P_{4'} + P_{4'_b}) + \alpha P_{2'_b} \tag{11}$$

$$P_4 = \alpha P_3 \qquad\qquad P_5 = \alpha P_4 \tag{12}$$

$$P_{4'} = \alpha P_{3'} \qquad\qquad P_{5'} = \alpha P_{4'} \tag{13}$$

$$P_{4'_b} = \alpha P_{3'_b} \qquad\qquad P_{5'_b} = \alpha P_{4'_b} \tag{14}$$

It's obvious that,

$$\begin{aligned} &P_1 = \alpha P_0, P_2 = \alpha^2 P_0, P_3 = \alpha^3 P_0, P_4 = \alpha^4 P_0, P_5 = \alpha^5 P_0 \\ &P_{0'} = \alpha(1-\alpha)(1-\beta)P_0, \\ &P_{0'_b} = \alpha(1-\alpha)\beta P_0. \end{aligned} \tag{15}$$

From Equation (7) and (8), we have,

$$\frac{P_{2'}}{P_{2'_b}} = \frac{1-\beta}{\beta} \Rightarrow P_{2'_b} = \frac{\beta}{1-\beta} P_{2'}. \tag{16}$$

Then, substituting Equation (16) into Equation (2), we get,

$$\begin{aligned}
P_0 &= (1-\alpha)P_0 + \alpha(1-\alpha)\beta P_0 + \alpha(1-\alpha)(1-\beta)P_0 \\
&+ (1-\alpha)(\alpha^2 P_0 + P_{2'} + \frac{\beta}{1-\beta}P_{2'}) \\
&\Rightarrow \alpha P_0 - \alpha(1-\alpha)P_0 - (1-\alpha)\alpha^2 P_0 = \frac{P_{2'}}{1-\beta} \\
&\Rightarrow \alpha P_0 - \alpha(1-\alpha)P_0(1+\alpha) = \frac{P_{2'}}{1-\beta} \\
&\Rightarrow \alpha P_0 - \alpha(1-\alpha^2)P_0 = \frac{P_{2'}}{1-\beta} \\
&\Rightarrow P_{2'} = (1-\beta)[\alpha - \alpha(1-\alpha^2)]P_0 \\
&\Rightarrow P_{2'} = \alpha^3(1-\beta)P_0.
\end{aligned} \tag{17}$$

Similarly,

$$P_{2'_b} = \alpha^3\beta P_0. \tag{18}$$

From Equations (10), (12),(13), and (14), we have,

$$\begin{aligned}
P_{3'} &= (1-\alpha)(1-\beta)(P_4 + P_{4'} + P_{4'_b}) + \alpha P_{2'} \\
&= (1-\alpha)(1-\beta)\alpha(P_3 + P_{3'} + P_{3'_b}) + \alpha P_{2'} \\
&= 2\alpha P_{2'} \\
&= 2\alpha^4(1-\beta)P_0.
\end{aligned} \tag{19}$$

From Equations (11), (12),(13), and (14), we have,

$$\begin{aligned}
P_{3'_b} &= \alpha(1-\alpha)\beta(P_3 + P_{3'} + P_{3'_b}) + \alpha P_{2'_b} \\
&= 2\alpha P_{2'_b} \\
&= 2\alpha^4\beta P_0.
\end{aligned} \tag{20}$$

Similarly, we have,

$$\begin{aligned}
P_{4'} &= 2\alpha^5(1-\beta)P_0, P_{4'_b} = 2\alpha^5\beta P_0, \\
P_{5'} &= 2\alpha^6(1-\beta)P_0, P_{5'_b} = 2\alpha^6\beta P_0.
\end{aligned} \tag{21}$$

4.2. The Expected Revenue

In this section, combining the probability of each state and the revenue listed in Table 2, we analyze the regular revenue (r_s^x), uncle revenue (r_u^x) and nephew

revenue (r_n^x) for x, where x denotes the attacker a or rational miners h. Further, *bribes* will also be included in the total revenue. Expected revenue is to estimate the various possible results of future revenues, and then use the probability of their occurrence to do a weighted average of these estimated revenues. In this subsection, the expected revenue is a weighted average of future revenues with various possible events.

4.2.1. Regular Revenue r_s^x

The regular revenues are calculated from the states transition according to Table 2. In each transition, regular revenues are generated when some blocks are released, shown in Table 2, where the tuple $< r_a, r_h >$ indicates the revenues of the attacker and rational miners respectively.

It can be seen from Table 2 that the attacker gains 3 regular revenues in events f-2,f-3, g-2, 2 regular rewards in events b-1, c-1, e-2 and 1 in events b-2, c-2, i-2 i-3, j-2, j-3 respectively. The attacker's regular revenue r_s^a is shown in Equation (22).

$$\begin{aligned} r_s^a =& 2P_{0'}\alpha + P_{0'}(1-\alpha)\gamma + 2P_{0'_b}\alpha + P_{0'_b}(1-\alpha) + \\ & 2P_2(1-\alpha) + 3P_{2'}(1-\alpha)\gamma + 3P_{2'}(1-\alpha)(1-\gamma) + \\ & 3P_{2'_b}(1-\alpha) + P_{3'}(1-\alpha)\beta + \\ & P_{3'}(1-\alpha)(1-\beta) + P_{3'_b}(1-\alpha)\beta + P_{3'_b}(1-\alpha)(1-\beta) \\ =& P_{0'}[2\alpha + (1-\alpha)\gamma] + P_{0'_b}(1+\alpha) + 2P_2(1-\alpha) + \\ & 3P_{2'}(1-\alpha) + 3P_{2'_b}(1-\alpha) + P_{3'}(1-\alpha) + P_{3'_b}(1-\alpha) \end{aligned} \tag{22}$$

Similarly, we could find the rational miners only get 2 revenues in event b-3, and get 1 revenue in events a-2, b-2, and c-2. We then derive the rational miners' regular revenue r_s^h is shown in Equation (23).

$$\begin{aligned} r_s^h &= P_0(1-\alpha) + P_{0'}(1-\alpha)\gamma + 2P_{0'}(1-\alpha)(1-\gamma) + P_{0'_b}(1-\alpha) \\ &= P_0(1-\alpha) + P_{0'}(1-\alpha)(1-\gamma) + P_{0'_b}(1-\alpha) \end{aligned} \tag{23}$$

4.2.2. Uncle Revenue r_u^x

Note that the attacker only gets uncle revenue when he loses the "0-lead" racing (i.e., event b-3 in Table 2) with 1 referenced distance. Hence, the attacker's uncle revenue r_u^a is shown in Equation (24).

$$r_u^a = P_{0'}(1-\alpha)(1-\gamma)k_u(1). \tag{24}$$

The rational miners receive uncle revenues from events b-1, b-2, c-1, c-2. The uncle revenue of rational miners r_u^h is shown in Equation (25).

$$\begin{aligned} r_u^h &= P_{0'}\alpha k_u(1) + P_{0'}(1-\alpha)\gamma k_u(1) + P_{0'_b}\alpha k_u(1) + P_{0'_b}(1-\alpha)k_u(1) \\ &= P_{0'}[\alpha + (1-\alpha)\gamma]k_u(1) + P_{0'_b}k_u(1) \end{aligned} \tag{25}$$

4.2.3. Nephew Revenue r_n^x

The attacker collects the nephew rewards in events b-1, c-1. The nephew revenue of the attacker r_n^a is shown in Equation (26).

$$r_n^a = P_{0'}\alpha k_n(1) + P_{0'_b}\alpha k_n(1) \tag{26}$$

The rational miners collect nephew revenues in the events b-2, b-3 and c-2 in Table 2. The nephew revenue of rational miners r_n^h is shown in Equation (27).

$$\begin{aligned} r_n^h &= P_{0'}(1-\alpha)\gamma k_n(1) + P_{0'}(1-\alpha)(1-\gamma)k_n(1) + P_{0'_b}(1-\alpha)k_n(1) \\ &= P_{0'}(1-\alpha)k_n(1) + P_{0'_b}(1-\alpha)k_n(1) \end{aligned} \tag{27}$$

4.2.4. The Cost of Bribes

Similar to [35], the attacker implements bribery attack in the private chain. That is, the attacker should pay *bribes* to rational miners if they choose to mine after the attacker's block in events c-1, c-2, j-2, j-3. The cost of the bribes of the attacker is shown in Equation (28).

$$\begin{aligned} r_b^a &= P_{0'_b}\alpha b + P_{0'_b}(1-\alpha)b + P_{3'_b}(1-\alpha)\beta b + P_{3'_b}(1-\alpha)(1-\beta)b \\ &= P_{0'_b}b + P_{3'_b}(1-\alpha)b \end{aligned} \tag{28}$$

In sum, the attacker's reward R^a_{total} for each state, listed in Equation (29), is the sum of the corresponding revenues: regular revenues, uncle revenues, nephew revenues and bribes.

$$r^a_{total} = r^a_s + r^a_u + r^a_n - r^a_b \tag{29}$$

Similarly, the reward of rational miners R^h_{total} is shown in Equation (30).

$$r^h_{total} = r^h_s + r^h_u + r^h_n + r^a_b \tag{30}$$

Generally, the indicator of relative reward is utilized to evaluate the attacks in selfish mining. Similar to [8, 19, 35], the relative rewards for the attacker and rational miners are defined in Equation (31) respectively. In the following sections, we highlight the relative rewards for the attacker and rational miners.

$$\begin{cases} R^a_{relative} = R^a_{total}/(R^a_{total} + R^h_{total}), \\ R^h_{relative} = R^h_{total}/(R^a_{total} + R^h_{total}). \end{cases} \tag{31}$$

5. Simulation Results

In this section, we analyze the effect of the parameters (e.g. α, β and γ) on relative rewards by the simulation. The simulation is implemented by the Python 3.9 programming language. Concretely, we model BSM-Ether as a Markov decision process. And the Markov decision process is solved by MDP-toolbox [48], which provides functions for the resolution of stochastic dynamic programming problems. In the simulation process, we use policy iteration to find the optimal strategy and calculate the rewards of the attacker. Further, we compare BSM-Ether with selfish mining (SM1), IPBSM and honest mining. In addition, we discuss the effect of BSM-Ether on uncle and nephew rewards respectively.

5.1. The Threshold of Relative Rewards

We simulate the relative rewards of BSM-Ether by setting $\gamma = 0.75$ in Ethereum. Note that IPBSM is not compared in Fig. 6 since it is only ap-

plied to Bitcoin. Recall that γ is set to be 0.5 in SM1. While in this section $\gamma = 0.75$, which means that rational miners prefer to accept the *bribes*. In the sequel, γ is larger than 0.5. In Fig. 6, the relative reward of the attacker is larger than that of SM1 and honest mining when $\alpha > 0.067$. As a control, the threshold for SM1 is 0.165. That is, BSM-Ether can achieve higher relative rewards with less power.

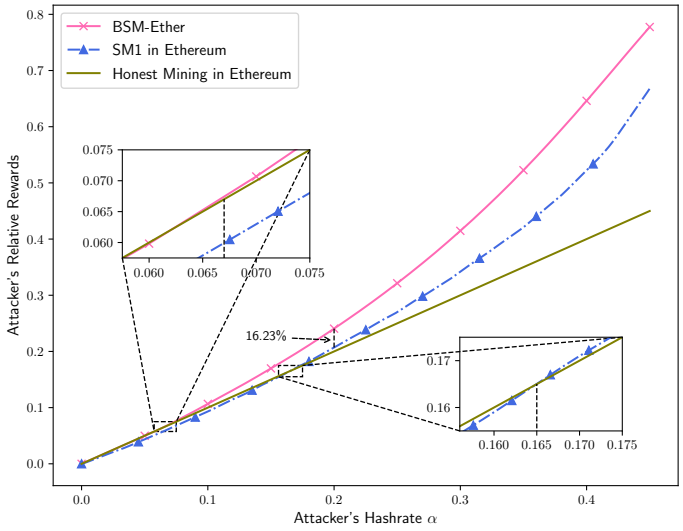


Figure 6: The attacker's relative rewards of BSM-Ether, Selfish Mining (SM1) and honest mining in Ethereum when α changes from 0 to 0.45 and $\gamma = 0.75$.

In addition, we compare the relative rewards of BSM-Ether in Ethereum with IPBSM and SM1 in Bitcoin (refer to Fig. 7). BSM-Ether, compared with IPBSM, increase the relative rewards by 15.34% when $\alpha = 0.2$ and $\gamma = 0.75$. These additional rewards are incurred by uncle and nephew rewards in Ethereum. Further, uncle and nephew rewards can compensate for, to some extent, the cost for sponsoring selfish mining. Therefore, the simulations prove again that Ethereum is more vulnerable to the family of selfish mining attacks.

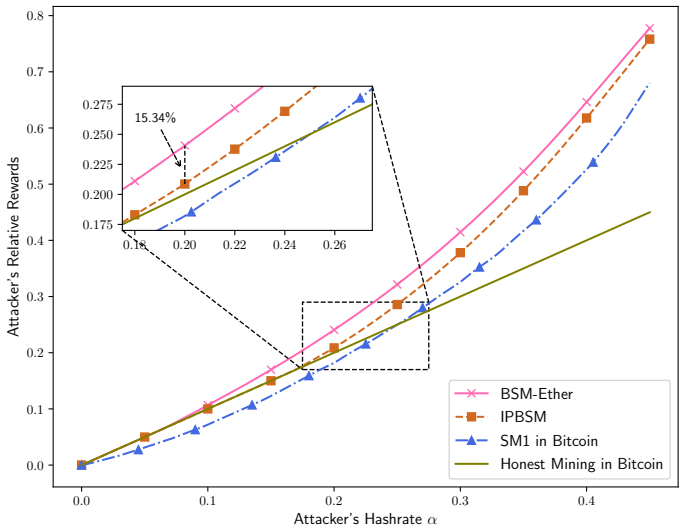
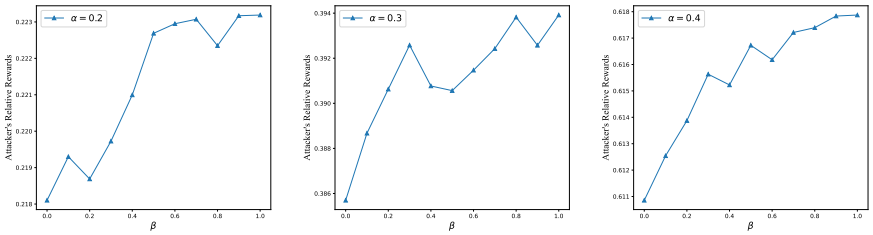


Figure 7: The attacker's relative rewards of BSM-Ether, IPBSM, SM1 and Honest mining in Bitcoin when α changes from 0 to 0.45 and $\gamma = 0.75$.

5.2. The Effect of β and γ on Rewards

In this subsection, we investigate the effect of β and γ on the relative rewards of BSM-Ether in Fig. 8 and Fig. 9 respectively. Overall, the relative rewards are positively correlated with β and γ. In other words, the attacker can collect more relative rewards by increasing the value of the two.



(a) The rewards with $\alpha = 0.2$.(b) The rewards with $\alpha = 0.3$.(c) The rewards with $\alpha = 0.4$.

Figure 8: The effect of β on attacker's relative rewards.

We first analyze the impact of β on the attacker's relative rewards. Recall

that the attacker puts bribes on his branch with probability β when taking action *Match*. The bribes will lure rational miners to append their new blocks to the attacker's branch. Although the bribes need to be paid as the attacking cost, the attacker could collect more rewards through this method, as demonstrated in Fig. 8. It is worth noting that increasing β does not necessarily increase the attacker's rewards. For example, when β changes from 0.3 to 0.5 in Fig. 8(b), the relative rewards decrease instead. But from the overall trend, increasing β is conducive to expanding rewards. Such as, when the attacker separately sets $\alpha = 0.2$, $\alpha = 0.3$, and $\alpha = 0.4$, letting $\beta = 1$ can always get the highest rewards.

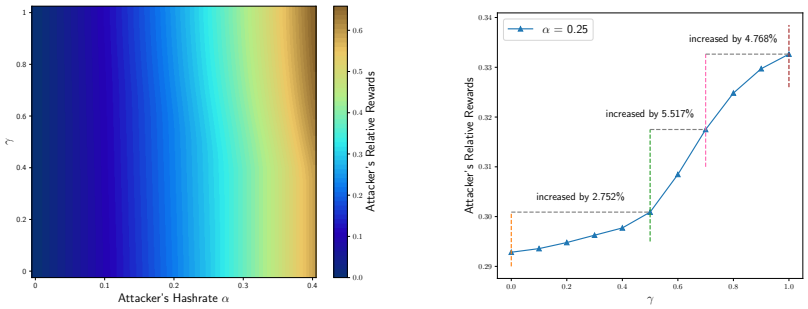
Further, we present the influence of α and γ on the relative rewards in Fig. 9(a). The relative rewards improve slowly when α is small ($0 \leqslant \alpha \leqslant 0.15$) since forks are rare when the attacker has small mining power. On the other hand, the relative rewards improve significantly when α is large ($0.25 \leqslant \alpha \leqslant 0.4$). The reason lies in that more forks are generated when the attacker has more power. In the sequel, the attacker can improve their rewards through bribing (resulting in a large γ). However, the rational miners fail to improve their rewards due to the invalidated blocks.

We then illustrate the impact of γ on the relative rewards in Fig. 9(b), where a median of $\alpha = 0.25$ is chosen. It's obvious that 2.752%, where γ changes from 0 to 0.5 and the attacker does not bribes the rational miners. While, the relative rewards of the attacker improve by 5.517% (4.768%), where γ changes from 0.5 to 0.7 (0.7 to 1). It means that bribery attack can further improve the attacker's relative revenue.

5.3. The Uncle and Nephew Rewards

The distinction of the threshold between IPBSM and BSM-Ether derives from the uncle and nephew rewards in Ethereum. Therefore, we further, in this subsection, analyze the uncle and nephew rewards for the attacker and rational miners in BSM-Ether.

Fig. 10(a) demonstrates the impact of α, γ on uncle rewards of the attacker, where the ratio of uncle rewards is inversely proportional to α, γ. For example,



(a) The attacker's relative rewards change with respect to α and γ.

(b) The impact of γ on attacker's relative rewards with $\alpha = 0.25$.

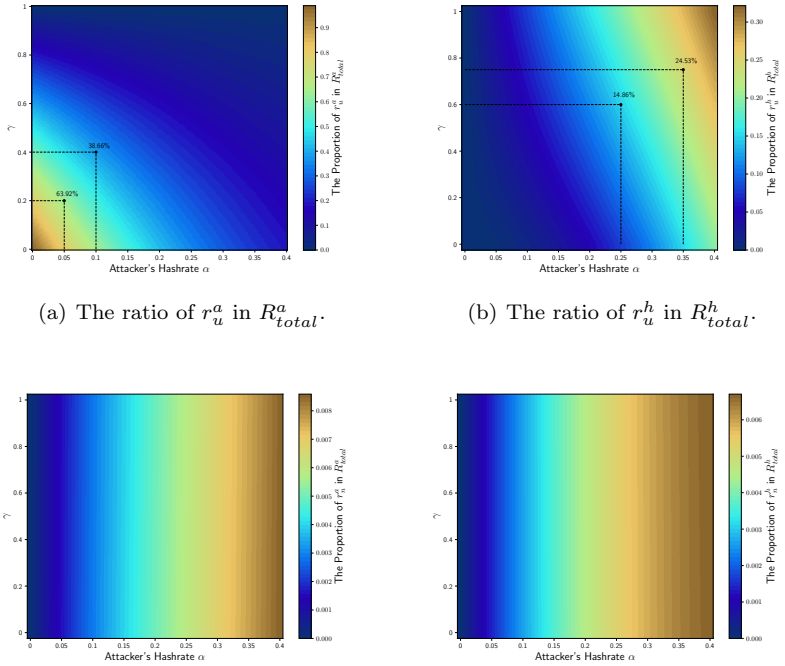
Figure 9: The effect of γ on attacker's relative rewards.

the ratio of uncle rewards of the attacker is 38.66% with $\alpha = 0.1$, $\gamma = 0.4$. While the ratio reaches 63.92% with $\alpha = 0.05$, $\gamma = 0.2$. Fig. 10(b) demonstrates the impact of α, γ on uncle rewards of rational miners, where the ratio of uncle rewards is proportional to α, γ. For example, the ratio of uncle rewards of the rational miners is 14.86% with $\alpha = 0.25$, $\gamma = 0.6$. While the ratio reaches 24.53% with $\alpha = 0.35$, $\gamma = 0.75$. Recall that the attacker generates more forks with larger α and rational miners chose to mine after the attacker's chain, which leading to a higher γ. When there is a fork, more blocks belonging to rational miners become uncle blocks during BSM-Ether. Therefore, the ratio of uncle blocks increases with α.

Contrary to the uncle rewards, the ratio of nephew rewards only relies on α (refer to Fig. 10(c) and 10(d)). The reason lies in that the nephew rewards are immune to the forks. That is, the nephew rewards remain unchanged no matter which branch is chosen in the fork. Meanwhile, the higher of α, the more blocks the attacker release in the private chain. Consequently, the attacker collects more nephew rewards.

5.4. Anti-measures against BSM-Ether

The above theoretical analysis and simulation results show that, compared with the selfish mining attack under Bitcoin, the BSM-Ether attack threshold



(a) The ratio of r_u^a in R_{total}^a.

(b) The ratio of r_u^h in R_{total}^h.

(c) The ratio of r_n^a in R_{total}^a.

(d) The ratio of r_n^h in R_{total}^h.

Figure 10: The ratio of uncle and nephew rewards for the attacker and the rational miners.

under Ethereum is lower and the revenue are higher. Therefore, the BSM-Ether attack is more harmful to the healthcare system. To reduce the loss caused by the BSM-Ether attack to the system, we give suggestions for the security of the healthcare system from several aspects such as node configuration, consensus mechanism and monitoring mechanism.

Suggestion 1 [node configuration]: In the blockchain-based healthcare system, when configuring nodes, each node is required to submit a considerable amount of deposit to the system (the deposit should be much larger than the expected mining income). Once there is a violation (such as establishing a private chain), the deposit will be confiscated. This can reduce the motivation of attackers to launch BSM-Ether attacks.

Suggestion 2 [consensus mechanism]: A new node type validator is introduced when building the blockchain-based healthcare system. The validators

verify and vote for each new block, where the block with more than 2/3 of the votes is marked as a valid block. We recommend that only the chain containing the vaild block is considered as the public chain. This can prevent an attacker from hiding a large number of blocks in the private chain.

Suggestion 3 [monitoring mechanism]: Add a monitoring mechanism in the blockchain-based healthcare system to monitor the proportions of various rewards in the system. If the proportion of rewards changes abnormally, such as a sudden increase in the proportion of the uncle rewards, it is also very likely that a BSM-Ether attack will occur, and corresponding countermeasures can be taken.

6. Conclusion and Future Work

In this paper, we propose BSM-Ether, based on IPBSM, by considering the uncle and nephew rewards in Ethereum. More specifically, we describe the process of BSM-Ether and then formally define it towards the view of MDP by presenting details the components of state set, action set and states transition etc. We further go into the rewards due to the existence of uncle and nephew rewards in BSM-Ether. Finally, we discuss the impact of the parameters on the relative rewards by comparing BSM-Ether with SM1, IPBSM and honest mining. The simulation results indicate that BSM-Ether has lower threshold for attacking. Meanwhile, the attacker can obtain higher relative rewards than other attacks. In addition, we find that the bribes can effectively improve γ and then lead to vulnerability of Etherem against BSM-Ether. In the future, we will stress on the optimization of bribes and focus on the scenario of multiple attackers.

Acknowledgments

This study is supported by the Foundation of National Natural Science Foundation of China (Grant Number: 62072273, 72111530206, 61962009, 61873117, 61832012, 61771231, 61771289); Natural Science Foundation of Shandong Province

(ZR2019MF062); Shandong University Science and Technology Program Project (J18A326); Guangxi Key Laboratory of Cryptography and Information Security (No: GCIS202112); The Major Basic Research Project of Natural Science Foundation of Shandong Province of China (ZR2018ZC0438); Major Scientific and Technological Special Project of Guizhou Province(20183001), Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2019BDKFJJ009), Talent project of Guizhou Big Data Academy. Guizhou Provincial Key Laboratory of Public Big Data. ([2018]01).

References

[1] The ethereum whitepaper, https://ethereum.org/en/whitepaper/, february 9, 2021.

[2] C. Ge, W. Susilo, J. Baek, Z. Liu, L. Fang, Revocable attribute-based encryption with data integrity in clouds, IEEE Transactions on Dependable and Secure Computing (2021) 1–1doi:10.1109/TDSC.2021.3065999.

[3] C. Ge, W. Susilo, J. Baek, Z. Liu, L. Fang, A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds, IEEE Transactions on Dependable and Secure Computing (2021) 1–1doi:10.1109/TDSC.2021.3076580.

[4] X. Lin, J. Li, J. Wu, H. Liang, W. Yang, Making knowledge tradable in edge-ai enabled iot: A consortium blockchain-based efficient and incentive approach, IEEE Transactions on Industrial Informatics 15 (12) (2019) 6367–6378. doi:10.1109/TII.2019.2917307.

[5] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, H. Zhou, Psspr: A source location privacy protection scheme based on sector phantom routing in wsns, International Journal of Intelligent Systems.Doi:10.1002/int.22666.

[6] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, Y.-a. Tan, Secure multi-party computation: theory, practice and applications, Information Sciences 476 (2019) 357–372.

[7] I. Eyal, E. G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: Proceedings of the International conference on financial cryptography and data security (FC), Springer, 2014, pp. 436–454.

[8] A. Sapirshtein, Y. Sompolinsky, A. Zohar, Optimal selfish mining strategies in bitcoin, in: J. Grossklags, B. Preneel (Eds.), Proceedings of the International conference on financial cryptography and data security (FC), Springer, 2017, pp. 515–532.

[9] Y. Wang, G. Yang, A. Bracciali, H. F. Leung, X. Yu, Incentive compatible and anti-compounding of wealth in proof-of-stake, Information Sciences 530 (2020) 85–94.

[10] D. Zhang, J. Le, X. Lei, T. Xiang, X. Liao, Exploring the redaction mechanisms of mutable blockchains: A comprehensive survey, International Journal of Intelligent Systems 36 (9) (2021) 5051–5084.

[11] T. Li, Y. Chen, Y. Wang, Y. Wang, M. Zhao, H. Zhu, Y. Tian, X. Yu, Y. Yang, Rational protocols and attacks in blockchain system, Security and Communication Networks 2020 (2020) 11, doi: 10.1155/2020/8839047.

[12] Y. Wang, G. Yang, T. Li, L. Zhang, Y. Wang, L. Ke, Y. Dou, S. Li, X. Yu, Optimal mixed block withholding attacks based on reinforcement learning, International Journal of Intelligent Systems 35 (12) (2020) 2032–2048, doi: 10.1002/int.22282.

[13] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, A. Hobor, On power splitting games in distributed computation: The case of bitcoin pooled mining, in: Proceedings of the 2015 IEEE 28th Computer Security Foundations Symposium (CSF), IEEE, 2015, pp. 397–411.

[14] I. Eyal, The miner's dilemma, in: Proceedings of the IEEE Symposium on Security and Privacy (S&P), IEEE, 2015, pp. 89–103.

[15] Y. Kwon, D. Kim, Y. Son, E. Vasserman, Y. Kim, 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 (CCS), ACM, 2017, pp. 195–209.

[16] S. Gao, Z. Li, Z. Peng, B. Xiao, Power adjusting and bribery racing: Novel mining attacks in the bitcoin system, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS), ACM, 2019, pp. 833–850.

[17] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, Y. Yang, Is semi-selfish mining available without being detected?, International Journal of Intelligent Systemsdoi:10.1002/int.22656.

[18] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, X. Yu, Semi-selfish mining based on hidden markov decision process, International Journal of Intelligent Systems 36 (7) (2021) 3596–3612. doi:10.1002/int.22428.

[19] C. Feng, J. Niu, Selfish mining in ethereum, in: Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2019, pp. 1306–1316.

[20] F. Ritz, A. Zugenmaier, The impact of uncle rewards on selfish mining in ethereum, in: Proceedings of the 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2018, pp. 50–57.

[21] C. Grunspan, R. Pérez-Marco, Selfish mining in ethereum, in: Proceedings of the Mathematical Research for Blockchain Economy (MARBLE), Springer, 2020, pp. 65–90.

[22] Y. Wang, Z. Wang, G. Yang, S. Ai, X. Xiang, C. Chen, M. Zhao, On-chain is not enough: Ensuring pre-data on the chain credibility for blockchain-based source-tracing systems, Digital Communications and Networksdoi:10.1016/j.dcan.2021.10.002.

[23] Y. Wang, A. Bracciali, T. Li, F. Li, X. Cui, M. Zhao, Randomness invalidates criminal smart contracts, Information Sciences 447 (2019) 291–301.

[24] Z. Wang, Q. Lv, Z. Lu, Y. Wang, S. Yue, Forkdec: Accurate detection for selfish mining attacks, Security and Communication Networks 2021. doi:10.1155/2021/5959698.

[25] S. Solat, M. Potop-Butucaru, Zeroblock: Preventing selfish mining in bitcoin, ArXiv abs/1605.02435.

[26] V. Chicarino, C. Albuquerque, E. Jesus, A. Rocha, On the detection of selfish mining and stalker attacks in blockchain networks, Annals of Telecommunications 75 (2020) 1–10, doi: 10.1007/s12243-019-00746-2.

[27] M. Saad, L. Njilla, C. Kamhoua, A. Mohaisen, Countering selfish mining in blockchains, in: Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), IEEE, 2019, pp. 360–364.

[28] M. Zhaofeng, W. Xiaochang, D. K. Jain, H. Khan, G. Hongmin, W. Zhen, A blockchain-based trusted data management scheme in edge computing, IEEE Transactions on Industrial Informatics 16 (3) (2020) 2013–2021. doi:10.1109/TII.2019.2933482.

[29] S. Qi, Y. Lu, Y. Zheng, Y. Li, X. Chen, Cpds: Enabling compressed and private data sharing for industrial internet of things over blockchain, IEEE Transactions on Industrial Informatics 17 (4) (2021) 2376–2387. doi:10.1109/TII.2020.2998166.

[30] N. Zhang, J. Xue, Y. Ma, R. Zhang, T. Liang, Y.-a. Tan, Hybrid sequence-based android malware detection using natural language processing, International Journal of Intelligent Systems 36 (10) (2021) 5770–5784.

[31] Q. Gu, S. Chen, S. Jiang, N. Xiong, Improved strength pareto evolutionary algorithm based on reference direction and coordinated selection strategy, International Journal of Intelligent Systems 36 (9) (2021) 4693–4722.

[32] R. Yang, X. Chang, J. Mišić, V. B. Mišić, Assessing blockchain selfish mining in an imperfect network: Honest and selfish miner views, Computers & Security 97 (2020) 101956, doi: 10.1016/j.cose.2020.101956.

[33] N. N. Son, V. K. Cao, H. Anh, Hysteresis compensation and adaptive control based evolutionary neural networks for piezoelectric actuator, International Journal of Intelligent Systems 36 (10) (2021) 5472–5492.

[34] X. Chen, F. Zhou, F. Zhang, M. Bonsangue, Modeling microscopic and macroscopic information diffusion for rumor detection, International Journal of Intelligent Systems 36 (10) (2021) 5449–5471.

[35] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, S. Li, Ipbsm: An optimal bribery selfish mining in the presence of intelligent and pure attackers, International Journal of Intelligent Systems 35 (11) (2020) 1735–1748, doi: 10.1002/int.22270.

[36] J. Garay, A. Kiayias, N. Leonardos, The bitcoin backbone protocol: Analysis and applications, in: Proceedings of the Annual international conference on the theory and applications of cryptographic techniques (EUROCRYPT), Springer, 2015, pp. 281–310.

[37] C. Dong, Y. Wang, A. Aldweesh, P. McCorry, A. van Moorsel, Betrayal, distrust, and rationality: Smart counter-collusion contracts for verifiable cloud computing, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (ACM), ACM, 2017, pp. 211–227.

[38] H. Sun, N. Ruan, C. Su, How to model the bribery attack: A practical quantification method in blockchain, in: Proceedings of the European Symposium on Research in Computer Security (ESORICS), Springer, 2020, pp. 569–589.

[39] G. Chunpeng, Z. Liu, J. Xia, F. Liming, Revocable identity-based broadcast proxy re-encryption for data sharing in clouds, IEEE Transaction-

s on Dependable and Secure Computing 18 (3) (2021) 1214–1226. doi:10.1109/TDSC.2019.2899300.

[40] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, F. Liming, Secure keyword search and data sharing mechanism for cloud computing, IEEE Transactions on Dependable and Secure Computingdoi:10.1109/TDSC.2020.2963978.

[41] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, L. Fang, Revocable attribute-based encryption with data integrity in clouds, IEEE Transactions on Dependable and Secure Computingdoi:10.1109/TDSC.2021.3065999.

[42] M. Saad, A. Anwar, S. Ravi, D. Mohaisen, Revisiting nakamoto consensus in asynchronous networks: A comprehensive analysis of bitcoin safety and chainquality, in: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 988–1005.

[43] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, S. Capkun, On the security and performance of proof of work blockchains, in: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 3–16.

[44] R. Sutton, A. Barto, Reinforcement Learning:An Introduction, Reinforcement Learning:An Introduction, 1998.

[45] Gagniuc, A. Paul, Markov chains: From theory to implementation and experimentation.

[46] J. Christopher, Q-learning. machine learning, Machine Learning 3.

[47] Williams, J. Ronald, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning 8 (3-4) (1992) 229–256.

[48] Iadine, Chads, Guillaume, Chapron, Marie-Jose, Cros, Frdrick, Garcia, Rgis, Sabbadin, Mdptoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems, Ecography 37 (9) (2014) 916–920.