Stowaway Mining: a Selfish Mining Against Strategy

Linqun Wang, Zumin Wang

Dalian University, No. 10, Xuefu Street, Jinzhou District, Dalian City, Liaoning Province, Dalian, Index, China

Abstract

Selfish mining is an attack strategy in blockchain, which "increases" its revenue by accumulating private branches and selectively releasing hidden blocks. Existing work has focused on the enhancement of selfish mining and the against to selfish mining attacks by changing the original mining rules and selfish mining games, but it is difficult to effectively counter the attacks without changing the mining rules. To address this problem, a strategy called "stowaway mining" is proposed, in which an undercover miner is infiltrated into a selfish mining pool to obtain information about its private branches and mine the latest blocks based on the private branches, and the process is modeled as a finite Markov reward process. The theory demonstrates that the relative revenue of the stowaway pool is strictly greater than that of the selfish pool when the stowaway and selfish pools have the same arithmetic power. The simulation compares the relative revenue of a mining pool with an honest mining strategy and with a stowaway mining strategy, and verifies that the stowaway mining strategy can reverse the mining advantage of the selfish mining pool under both equal and lesser arithmetic power, and is an effective strategy to against selfish mining attacks with zero loss.

Keywords

blockchain; selfish mining; mining pool; pool revenue

1. Introduction

As electronic payment technology continues to advance, more and more decentralised digital currencies are appearing in the public eye. Bitcoin[1] as the pioneer of digital currencies, invented a decentralised universal currency system that for the first time used Proof of Work (POW)[2] consensus as the cornerstone of global consensus. The essence of the system is for all miners to calculate mathematical puzzles where the answers are easily verifiable and consensus is formed, and the person who solves it gets the right to keep track of the global electronic bills and receives a certain amount of bitcoins into account. This process is known as mining. However, as hardware performance increases and the amount of computing power put into the Bitcoin system increases dramatically, the difficulty of mining rises, and it can take months or even years for an individual miner to mine a block[3] . To avoid the instability of revenue, many individual miners join mining pools to form mining groups, which are managed by pool managers and work together to mine blocks, splitting their contribution.

In the Bitcoin system, if all miners mined according to Bitcoin rules, individual gains would be proportional to their computing power. However, this norm is broken by the selfish mining attack proposed by Eyal and Sirer[4], in which an attacker actively builds a private chain and selectively releases hidden blocks based on factors such as the length of the public chain and the length of their own private chain. The essence behind the attack is to trick honest miners into wasting mining arithmetic. As a result, selfish mining pools can gain a higher relative revenue than their share of arithmetic power. The literature [5-7] models selfish mining as a Markov decision process and obtains a more robust selfish mining strategy that can maximize revenue arbitrarily close to the maximum. The literature [8] solves for the selfish and honest mining probabilities by size method to determine the release of hidden blocks and increase the relative revenue of selfish mining. The literature [9] improves

ICBASE2022@3rd International Conference on Big Data & Artificial Intelligence & Software Engineering, October 21-23, 2022, Guangzhou, China

EMAIL: vipwanglinqun@163.com(linqun Wang)
ORCID:0000-0002-7169-0999 (linqun Wang);

the efficiency of selfish mining by combining it with other attacks. In the face of selfish mining attacks, literature [10] proposes adding anti-forgery timestamps to the system to counter selfish mining attacks. Literature [11] proposes a mechanism to prevent block hiding, which can make the hidden blocks not recognized by the public chain. The literature [12] proposes a new framework based on deep reinforcement learning, SquirRL, whose experiments show that when faced with a selfish mining attack, using a selfish mining strategy to counteract it is not the best choice. In [13-16], we study the "prisoner's dilemma" caused by the game between mining pools when there are multiple selfish mining pools in the system, and propose some strategies to reduce the wasted computing power caused by the game. However, these researchers do not consider how a selfish mining pool can cope with selfish mining attacks without changing the rules of the Bitcoin system, nor do they propose a way to reverse the mining lead of a selfish pool without changing the rules of the Bitcoin system, which is the focus of this paper.

Inspired by the act of sending undercover miners to[17] in the interception attack, this paper proposes a strategy called "stowaway mining", where the set of miners using the stowaway mining strategy is called the stowaway pool and the set of miners using the selfish mining strategy is called the selfish pool. In a stowaway mining strategy, once a stowaway pool discovers the presence of a selfish pool in its system, it sends undercover miners to monitor the length of its private chain and the hash value of the latest blocks on the private chain. With this information, the stowaway pool is able to know the current status of mining in the system, i.e. the length of the public chain and the length of the private chain hidden by the selfish pool. This information is then used to perform different mining behaviour. On the one hand, when the selfish pool is observed to be ahead, the stowaway pool will mine implicitly on top of its lead and force the selfish pool to release hidden blocks under certain conditions. On the other hand, when a stowaway pool is in the lead, it will execute a strategy similar to selfish mining.

The main works of this paper are: 1) Based on the selfish mining strategy, a strategy called stowaway mining is proposed, and the stowaway mining strategy is modeled as a two-dimensional Markov chain reward process. 2) It is theoretically demonstrated that the relative revenue of the stowaway mining strategy's pool is strictly greater than that of the selfish mining strategy's pool under the same arithmetic power. 3) The simulation quantifies the relative revenue of a mining pool with an honest mining strategy and a mining pool with a stowaway mining strategy, and shows that if there is a selfish mining pool in the system that affects one pool's revenue, the stowaway mining strategy can protect its revenue and reverse the mining advantage of the selfish mining pool. 4) The simulation simulates the arithmetic power threshold for the stowaway mining strategy to against the selfish mining attack, and verifies that the stowaway mining pool can also effectively against attack when its arithmetic power is smaller than that of the selfish mining pool within a certain range.

2. Stowaway mining strategies

2.1.Models and strategies

First define that there are *n* miners in the Bitcoin system and the system arithmetic share of miner *i* is m_i , then we can find $\sum_{i=1}^{n} m_i = 1$, such that *H, S, and T are taken* as the set of honest miners, selfish miners, and stowaway miners, respectively. Since honest miners strictly follow the Bitcoin protocol and do not hide block information from each other, they are considered as a whole and are called honest mining pools. Similarly, all selfish miners using a selfish mining strategy are combined together as a single agent, which is known as the selfish pool. The remaining set of miners is called the stowaway pool and uses the following stowaway strategy. Let α be the system arithmetic share of the selfish pool and β be the system arithmetic share of the stowaway pool, then we have $\alpha = \sum_{i \in S} m_i$ and $\beta = \sum_{i \in T} m_i$, and the system arithmetic share of the honest pool can be expressed as $1-\alpha-\beta$, so the arithmetic share of the selfish pool is α , the arithmetic share of the stowaway pool is β , and the arithmetic share of the honest pool is $1-\alpha-\beta$. Based on previous work at[4][5] , in this paper we also assume that the time to broadcast a block is negligible, that transaction fees are negligible, and that honest miners randomly choose blocks to mine. In other words, these mining pools derive their revenue mainly from block rewards. In addition, block generation is treated as a stochastic model that generates a new block in each time slot, and when there are equal-length chains in the system, honest miners randomly choose one for the next block to mine.

The stowaway mining strategy is now described (see Figure 1). Before discussing it in detail, the undercover behavior of undercover miners is explained. Firstly, the behavior of sending undercover miners is borrowed from the undercover[17] behavior of the interception attack, but the undercover of the interception attack only discards the mined blocks and reduces the arithmetic power of the undercovered pool, while the undercover of the stowaway strategy is to gain the mining status of the selfish mining pool. Now an example of undercover scenario, selfish mining behavior will leave many consecutive private chains and some consecutive isolated blocks in the main chain, so the stowaway pool can detect the selfish pool's storage by periodically scanning the main chain[18] , once the stowaway pool finds the selfish pool in the network it will select suitable undercover miners in its own pool, and deliver an undercover message to the miners through the mining pool protocol mining pool manager, after the undercover receives the message through If the manager chooses to execute, the pool manager will inform the undercover miner of the selfish pool found, because the pool is publicly joined, the undercover miner can apply to join the selfish pool without quitting the stowaway pool, and after entering the selfish pool as a member of the selfish pool, it will receive a mining task from the manager of the selfish pool, the mining task assigned by the manager is essentially a The mining task assigned by the manager is essentially a data structure that contains the manager-determinable *address, merkle root, nounce* and a fixed, unchanging hash value of the previous block, so the undercover agent must be able to get the hash value of the previous block from the task. Normally, this hash value corresponds to the last block of the common chain, but once a selfish mining pool has mined a block, its manager will privately keep the block and issue a new task based on it, at which point the hash value does not match the hash value of the last block of the common chain. From the perspective of the undercover miner, there is no new block released in the public chain, but the manager of the selfish pool releases a new task based on an unknown block. Then, it can be concluded that the manager of the selfish mining pool hides the block. Furthermore, the number of hidden blocks is equal to the sum of the number of unmatched hash blocks. Secondly selfish mining pools expose the hash value of the latest block in the hidden block list when they issue mining tasks, and it is based on this hash value that the stowaway mining strategy overpowers to checkmate selfish pools. Firstly the undercover behaviour takes advantage of the normal rules of the Bitcoin system, secondly the control of the undercover is entirely determined by the mining pool protocol, which is determined by the pool itself and does not attempt to change the Bitcoin system rules, so the undercover behaviour and undercover information gathering is perfectly feasible. The undercover miner takes normal mining behaviour in the selfish pool, working for the selfish pool and receiving mining rewards from the selfish pool, while passing mining information from the selfish pool back to the stowaway pool and receiving undercover rewards from the stowaway pool, a two-way profitable process. The stowaway pool will prefer to choose miners with low arithmetic power and high activity rate as undercover miners, which can ensure the timeliness of message return, and the undercover reward is much larger than the mining reward in the selfish pool, which can increase the loyalty of undercover miners. In order to further improve the accuracy of undercover messages and prevent dishonest behaviour of undercover miners, the stowaway pool will send 3 undercover miners to the selfish pool at the same time, if two or more of the returned messages are the same, the message will be considered accurate, if all 3 messages are different, the original undercover miner will be dropped and 3 miners will be chosen again.

Through undercover miners, the stowaway pools gain access to the current mining status in the system, i.e. the length of the public chain and the length of the private chain hidden by the private pools. While all mining pools mine based on the public chain information, the three pools can hold different sub-chains (also called branches) behind the public chain in the mining competition. Let l_{h} , l_{s} ,and l_{t} denote the branch lengths of the Honest, Selfish, and Stowaway pools, respectively. For ease of calculation, the branch lengths are all calculated from 0. In the mining process, the honest pool only knows the length of the honest branch (the public chain of the blockchain), the selfish pool knows the length of both the selfish and honest branches, and the stowaway pool can observe the length of the branches of all three pools. The miners in the above three pools mine blocks based on their own pools' information, and generate blocks through a round-robin competition, with each round starting from the global consensus reached by the longest public chain in the system. When the stowaway pool and the selfish pool have disclosed all their hidden blocks, or no hidden blocks, the current round of competition ends, and the next round starts from the global consensus, and so on. For the first block of each round of competition, there are three possible scenarios.

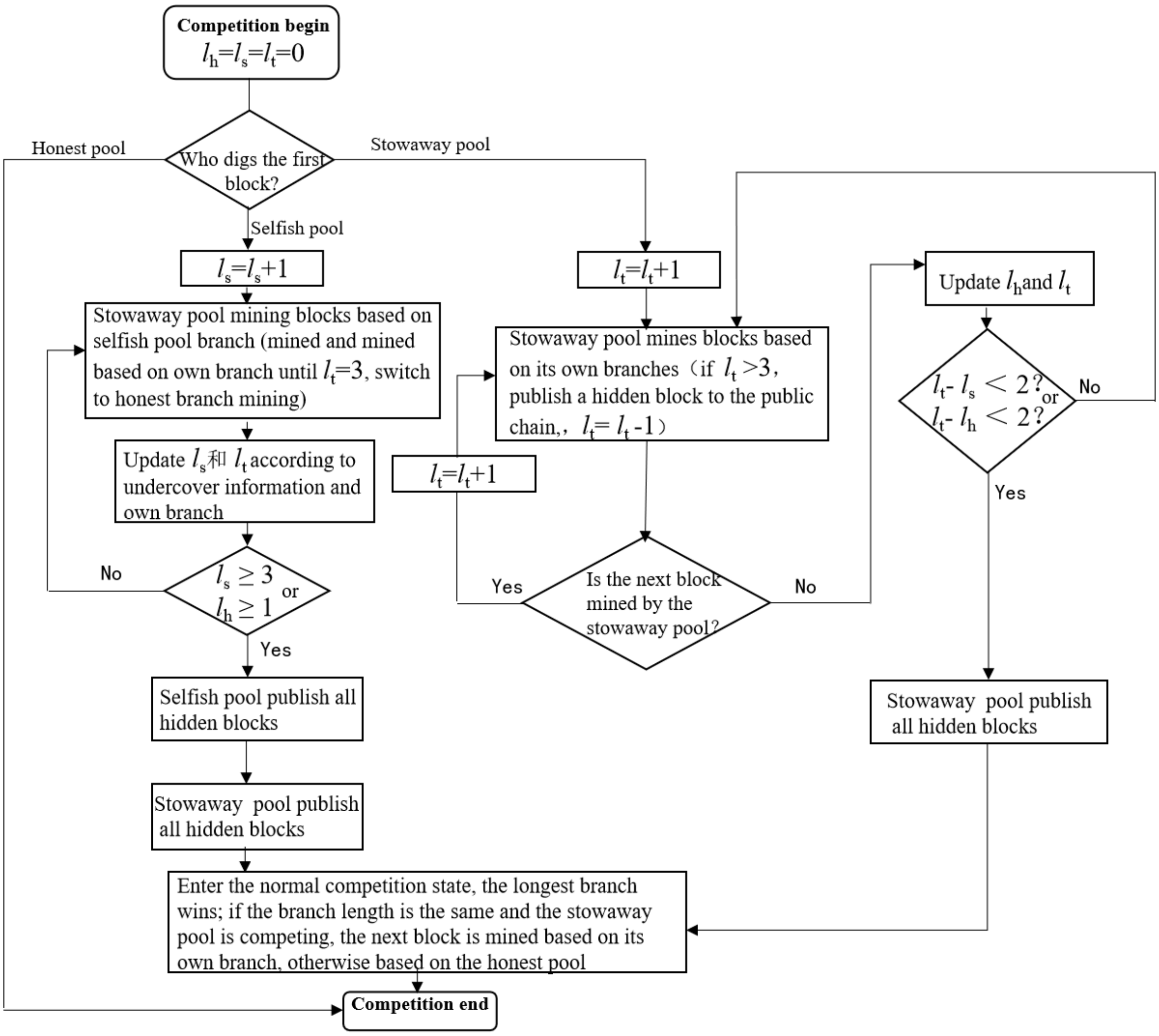

Competition begin
l_h=l_s=l_t=0
Honest pool
Who digs the first block?
Stowaway pool
Selfish pool
l_s=l_s+1
Stowaway pool mining blocks based on selfish pool branch (mined and mined based on own branch until l_t=3, switch to honest branch mining)
Update l_s和 l_t according to undercover information and own branch
No
$l_s \geq 3$ or $l_h \geq 1$
Yes
Selfish pool publish all hidden blocks
Stowaway pool publish all hidden blocks
Enter the normal competition state, the longest branch wins; if the branch length is the same and the stowaway pool is competing, the next block is mined based on its own branch, otherwise based on the honest pool
Competition end
l_t=l_t+1
Stowaway pool mines blocks based on its own branches（if l_t >3，publish a hidden block to the public chain,，l_t= l_t -1）
l_t=l_t+1
Yes
Is the next block mined by the stowaway pool?
No
Update $l_h$and l_t
l_t- l_s < 2? or l_t- l_h < 2?
No
Yes
Stowaway pool publish all hidden blocks

Figure 1 Stowaway mining strategy flow

Case 1: The honest pool is the first to mine a block with a probability of $1-\alpha-\beta$, and after mining a bitcoin block, the honest pool will quickly publish the block to the system public chain, in which case both the selfish pool and the stowaway pool acknowledge the block and the current round of competition ends to move on to the next round of competition.

Case 2: The selfish pool is the first to mine the block with a probability of α . In order to balance the complexity of the model and the actual scenario, the selfish pool adopts a selfish mining strategy, that is, when the length of l_s has not yet reached 3, the selfish attack behaves in the same way as a traditional selfish attack, but when the length of reaches 3, the selfish pool will actively release all the hidden blocks. The mining strategy of the stowaway pool is obtained through undercover information. When the stowaway pool finds out that the selfish pool has hidden blocks, it will mine based on the top end of the hidden blocks, and once the blocks are mined, it has its own stowaway branch and then moves to its own stowaway branch for mining. The stowaway pool will release its own stowaway blocks in the following two cases: first, the selfish pool releases the hidden blocks on which the stowaway branch depends, and second, when l_t =3, the stowaway pool needs to move to the public chain to mine, forcing the selfish pool to release its own hidden blocks, and then the stowaway pool follows by releasing the hidden blocks on its own stowaway branch, reducing the risk of mining while gaining stowaway revenue.

Case 3: The stowaway pool is the first to mine a block with a probability of β . When the stowaway pool mines a block first, it is in the mining lead in the system and does not release the block directly in order to maintain the mining lead, but hides it and mines on it implicitly. When the difference between the number of hidden blocks in its own branch and the number of other branches is less than 2, it immediately releases all hidden blocks. The maximum number of hidden blocks is 3. When the number exceeds 3, a block is actively released, always keeping the maximum number of hidden blocks at 3.

2.2.Markov reward processes

To analyze the relative revenue of each pool under the stowaway strategy, $s=(\mu,\nu)$ is used to represent the state in the system. μ indicates the number of blocks where the selfish pool is ahead of the honest pool, i.e. the number of blocks hidden by the selfish pool. ν denotes the number of blocks that the stowaway pool has ahead of the hidden branch of the selfish pool, the superscript of ν denotes all blocks mined by the stowaway pool (based on the hidden blocks of the stowaway pool), and the superscript * denotes an equal-length competition phase, where the winning power is determined based on the creation of the next block. For example, $(0^*,0)$ means that the selfish mining pool released only one hidden block immediately after the honest mining pool released one block, and then an equal-length competition is formed on the public chain, with the winning right determined by the generation of the next block. A two-dimensional Markov state transfer diagram (see Figure 2) was created based on the above states and transfer probabilities, where the states (0,0) indicate the same state, it's a return to global consensus.

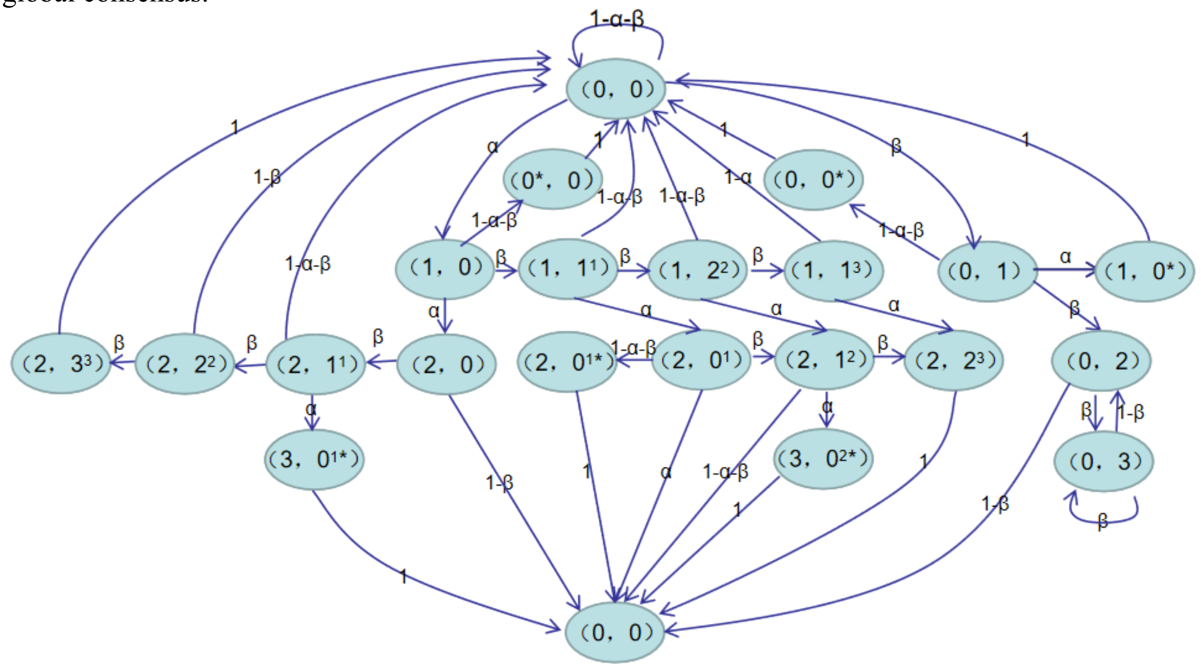


Figure 2 Markov state transfer diagram

The Markov state transfer diagram shows that one branch wins at the end of each round of competition (reversion to global consensus), and the gains of each branch at the end of each round of competition can be easily calculated based on the last state change before reversion to global consensus, so the expected gains of each mining pool will be known at the end of a round of competition. According to the model, block vesting during competition is not certain and the expected gain for each mining pool is only accounted for when a round of competition ends and reverts to global consensus. To quantify the relative revenue various parameters in state transfers are elicited, $P_{ss'}$ denotes the probability of going from state s to state s' and $R_{ss'}$ denotes the expected gain from this state transition. It contains three components, corresponding to the payoffs of the honest pool, the selfish pool and the stowaway pool, respectively. W_s denotes the steady-state probability of the state, which can be calculated from the state transfer probability and the state transfer diagram. All steady-state probabilities can be used to denote $W_{(0,0)}$.

2.3.Relative revenue

Let $M=\{H,S,T\}$ denote the set of miners, miner $i\in M$, where H is the set of honest miners, S is the set of selfish miners, T is the set of stowaway miners, $s_z=s(0,0)$ denotes that the competition starts with a global consensus, R_{z+1}^{i} denotes the gain of miner i at the end of a round of competition, and each addition of 1 to z denotes another round of competition, then the relative revenue of any miner $i\in M$ is

$$RREV_i=E\left[\frac{\sum_{k=0}^{\infty}R_{z+k+1}^{i}\mid s_z=s(0,0)}{\sum_{k=0}^{\infty}\sum_{j\in M}R_{z+k+1}^{j}\mid s_z=s(0,0)}\right] \tag{1}$$

Because it is a finite Markov reward process, the relative revenue will be stable after a certain time, close to the relative payoff expectation for each round of competition, and because the probability of state transfer for each round of competition P and the reward function R depend on α and β . Thus the Markov reward process can be reduced to a binary function on α and β . Then for each miner $i\in\{H,S,T\}$ we have:

$$ER_i(\alpha,\beta)=\sum_{s\in SW}\sum_{s'\in SW}P_{ss'}\cdot R_{ss'}^{i}\cdot W_s \tag{2}$$

Where SW denotes the set of steady states and W_s denotes the steady state probability of the state s , the relative revenue of the selfish and stowaway mining pools are

$$RRER_S(\alpha,\beta)=\frac{ER_S(\alpha,\beta)}{ER_H(\alpha,\beta)+ER_S(\alpha,\beta)+ER_T(\alpha,\beta)} \tag{3}$$

$$RRER_T(\alpha,\beta)=\frac{ER_T(\alpha,\beta)}{ER_H(\alpha,\beta)+ER_S(\alpha,\beta)+ER_T(\alpha,\beta)} \tag{4}$$

So the relative revenue difference between the stowaway pool and the selfish pool is obtained as:

$$DRRER(\alpha,\beta)=RRER_T(\alpha,\beta)-RRER_S(\alpha,\beta) \tag{5}$$

Like the previous work at[21], focus first on the case where the stowaway and selfish mining pools have the same arithmetic power. In the $\alpha=\beta$ scenario, the difference in relative revenue between the stowaway and selfish mining pools can be further reduced to a monadic function on α with the following function.

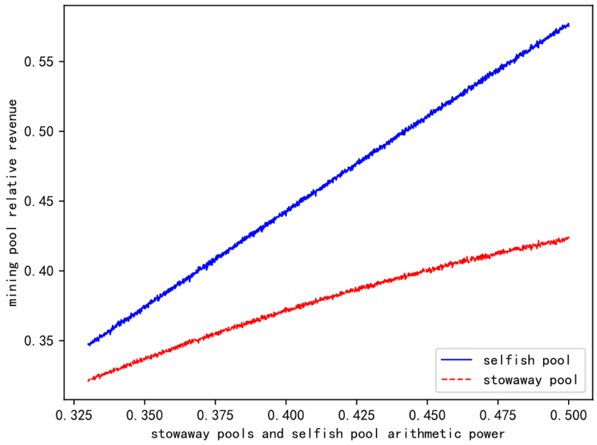
$$DRRER(\alpha,\beta)=DRRER_T(\alpha)=\frac{\alpha^2-2\alpha^3+\dfrac{2\alpha^3}{(1-\alpha)^2}-4\alpha^5}{2+4\alpha+8\alpha^3+\dfrac{2\alpha^3}{(1-\alpha)^2}-2\alpha^5} \tag{6}$$

It is easy to see that the relative revenue difference function between the stowaway pool and the selfish pool is strictly increasing for the same arithmetic power, and the relative revenue of the stowaway pool is always higher than that of the selfish pool, and the higher the arithmetic power, the greater the degree of lead.

3. Simulation analysis

The simulations in this subsection evaluate the effectiveness of the stowaway mining strategy. The honest mining pool, the selfish mining pool and the stowaway mining pool are treated as three independent agents. Competition between mining pools is modeled as a discrete-time random wandering process. At each step, any pool may generate blocks with a probability proportional to the pool's hash arithmetic power, and after the first block is generated, the other pools react according to their own strategies. The simulation ends after 500,000 steps, and the relative revenue of the process is calculated, which is defined as the percentage of blocks generated by each pool on the public chain. Selfish mining pools because of the accumulation of hidden blocks in the early stage, when the arithmetic power is relatively low, the first hidden block is always annihilated by other arithmetic power, so it leads to a decrease in its gain, and only when the arithmetic power of selfish pools is above a certain threshold, its relative revenue increases rapidly, encroaching on the relative revenue of other miners. Based on previous work[4][20] it is known that this arithmetic threshold is 33%, i.e. for a selfish pool in the system to launch a selfish mining attack, its arithmetic power must reach more than 33% of the system's arithmetic power, and so the simulation is performed above this arithmetic power range. The simulation code is published on the platform https://gitee.com/monstermr/smuggling-and-mining-fa.

First, a scenario is simulated where a selfish mining pool launches a selfish mining attack and the stowaway pool remains honestly mining without adopting a against strategy..



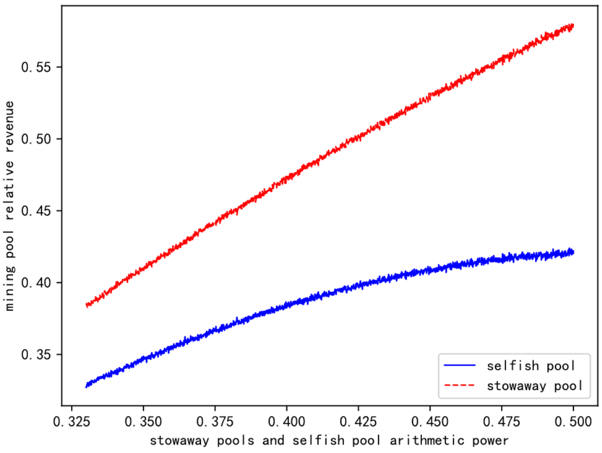


Figure 3 Non-adoption of coping strategy **Figure 4** Adoption of stowaway mining strategy

From the Figure 3, we can see that when the stowaway pool does not adopt any counter strategy under the same arithmetic power, its relative revenue is gradually stolen by the selfish pool, and as the arithmetic power of the selfish pool increases, the stowaway relative revenue rapidly decreases and is significantly lower than the relative revenue in an honest environment.

Then, the impact on selfish mining attacks is simulated when a stowaway mining pool adopts a stowaway mining strategy.

The Figure 4 shows that the relative revenue of mining pools with the stowaway mining strategy are consistently higher than those of selfish pools under the same arithmetic power, which is consistent with previous theoretical proofs. This is consistent with previous theoretical proofs. As the arithmetic power of the stowaway pool increases, its relative revenue increases rapidly and is always higher than the relative revenue in an honest environment, while the relative revenue of the selfish pool decays rapidly and is lower than its relative revenue in an honest environment. Comparing Figure 3 and Figure 4, we can conclude that when there is a selfish mining attack in the system that affects the pool's revenue at the same computing power, using a stowaway mining strategy not only avoids damage to pool revenue, but also gains "extra revenue" above the revenue in the original honest environment, significantly reversing the advantage of the selfish mining pool, It is an effective strategy to against selfish mining attacks.

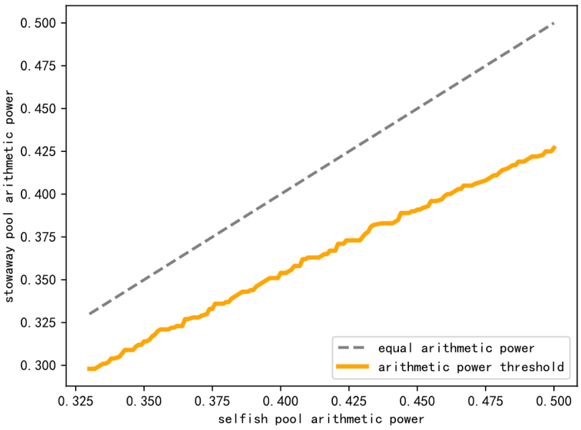


Figure 5 Stowaway pool arithmetic power threshold

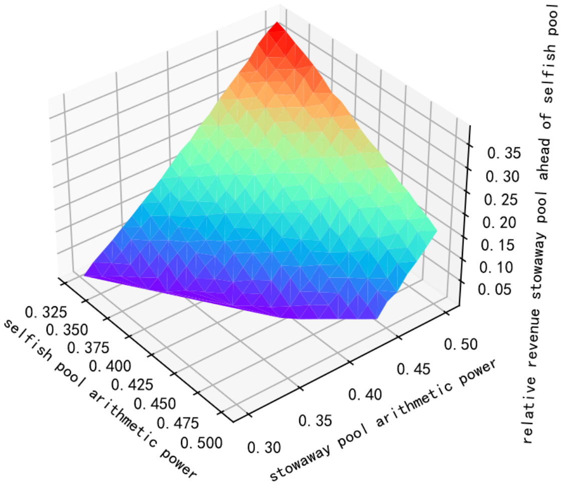


Figure 6 Relative revenue spread

Next, a scenario with different arithmetic power between the selfish and stowaway pools is simulated to consider whether less arithmetic power can also enable the stowaway pool to gain more revenue while resisting attacks from selfish mining(see Figure 5). The solid line in Figure 5 corresponds to the threshold of $RRER_T(\alpha,\beta) = RRER_S(\alpha,\beta)$, above which the relative revenue of the stowaway pools are higher than those of the selfish pools for the range of arithmetic values, as discussed earlier for the case where the selfish and stowaway pools have the same arithmetic power (indicated by the dashed line). This shows that with a stowaway mining strategy, a smaller amount of arithmetic power can be used to counter a selfish mining attack.

Finally, the relative revenue of the stowaway pool ahead of the selfish pool when the selfish pool and the stowaway pool have different arithmetic power is simulated(see Figure 6). As can be seen from the graph, the more the stowaway pool is ahead of the selfish pool in terms of arithmetic power, the more it is ahead in terms of relative revenue, up to 36%.

4. Conclusion

This paper models the various competing states between honest, selfish and stowaway pools as state points in a finite-state two-dimensional Markov reward process, and demonstrates that when there is a selfish pool and a stowaway pool with the same arithmetic power, the stowaway pool is able to achieve a higher expected revenue than the selfish pool. This shows that the stowaway pool is able to reverse the dominance of the selfish pool. In addition to the above theoretical results, a series of simulation experiments are conducted in this paper to verify that the stowaway mining strategy can against selfish mining attacks with both equal and lesser arithmetic power.

Selfish mining attacks and POW consensus are closely related, so this scheme is applicable to blockchain systems based on POW consensus and does not work on systems based on other consensus algorithms. The research in this paper fails to address the game process of multi-selfish mining pools and multi-stowaway mining pools, and future research will focus on the multi-pool game process under the stowaway mining strategy.

5.Reference.

[1] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system [EB/OL]. (2008) [2021-12-01].https://bitcoin.org/files/bitcoin-paper/bitcoin_zh_cn.pdf.

[2] BACK A. Hashcash-a denial of service counter-measure [EB/OL]. (2002) [2021-12-01]. http://www.hashcash.org/hashcash.pdf.

[3] SCHRIJVERS O, BONNEAU J, BONEH D, et al. Incentive compatibility of Bitcoin mining pool reward functions [C]// Proceedings of the International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2016: 477-498.

[4] EYAL I, SIRER E G. Majority is not enough: Bitcoin mining is vulnerable [C]// Proceedings of the International Conference on Financial Cryptography Berlin: Springer, 2014: 436-454.

[5] SAPIRSHTEIN A, SOMPOLINSKY Y, ZOHAR A. Optimal selfish mining strategies in Bitcoin [C]// Proceedings of the International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2016: 515-532.

[6] Yang Zejun. Blockchain selfish mining strategy based on deep learnin [D]. Shenzhen: Shenzhen University, 2020: 46-58

[7] ZUR R B, EYAL I, TAMAR A. Efficient mdp analysis for selfish-mining in blockchains [C]// Proceedings of the Proceedings of the 2nd ACM Conference on New York: ACM, 2020: 113-131.

[8] Gang Yinglin, Cheng Xiaorong. Selfish mining research and analysis in blockchain [J]. Computer Engineering and Applications, 2018, 54(15): 62-66

[9] HEILMAN E. One weird trick to stop selfish miners: fresh bitcoins, a solution for the honest miner (Poster Abstract) [C]// Proceedings of the International Conference on Financial Cryptography andData Security. Berlin: Springer, 2014: 161-162.

[10] ZHANG R, PRENEEL B. Publish or perish: a backward-compatible defense against selfish mining in bitcoin. Cham: Springer, 2017: 277-92.

[11] KARAME G O, ANDROULAKI E, CAPKUN S. Double-spending fast payments in bitcoin [C]// Proceedings of the 2012 ACM Conference on Computer and Communications Security. New York: ACM, 2012: 906-917.

[12] HOU C, ZHOU M, JI Y, et al. SquirRL: automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning [EB/OL]. (2020-08-04) [2021-12-01].https://arxiv.org/pdf/1912.01798v2.pdf.

[13] LIU H, RUAN N, DU R, et al. On the strategy and behavior of Bitcoin mining with N-attackers [C]// Proceedings of the 2018 on Asia Conference on Computer New York: ACM, 2018: 357-368.

[14] MARMOLEJO-COSSIO F J, BRIGHAM E, SELA B, et al. Competing (Semi-) selfish miners in Bitcoin [C]// Proceedings of the 1st ACM Conference on Advances in New York: ACM, 2019: 89-109.

[15] BAI Q, ZHOU X, WANG X, et al. A deep dive into blockchain selfish mining [C]// Proceedings of the 2019 IEEE International Conference on Communications . Piscataway, NJ: IEEE, 2019: 1-6.

[16] Ran Na, Liu Hanqing, SI Xueming. Catfish effect between selfish miners in proof-of-work based blockchain [J]. Chinese Journal of Computer, 2021, 44(01): 177-192

[17] Liu Zizhou, Cheng Xiaorong, Wang Zhibo. Research and analysis of block interception attacks in blockchain [J]. Computer Engineering and Applications, 2022, 58(04): 118-125

[18] Kędziora M, Kozłowski P, Szczepanik M, et al. Analysis of blockchain selfish mining attacks [C]// Proceedings of the 40th Anniversary International Conference on Information Systems Architecture and Technology. cham: Springer, 2019: 231-240.