https://doi.org/10.1007/s11704-026-51791-9

RESEARCH ARTICLE

To be detected or not: a hybrid selfish mining attack and countermeasures

Yilei WANG[1,2], Junxi ZHU[2], Zhaojie WANG[3], Andrea BRACCIALI[4]✉, Minghao ZHAO[5], Wenjian LUO[6,7], Huiyu ZHOU[8]

1. College of Artificial Intelligence, Jiaxing University, Jiaxing 314001, China
2. School of Computer Science, Qufu Normal University, Rizhao 276827, China
3. School of Computer Science and Technology, Soochow University, Suzhou 215031, China
4. Department of Computer Science, University of Turin, UNITO 10124, Italy
5. School of Data Science and Engineering, East China Normal University, Shanghai 200062, China
6. School of Computer Science and Technology, Harbin Institute and Technology, Shenzhen 518055, China
7. Guangdong Provincial Key Laboratry of Novel Security Intelligence Technologies, Shenzhen 518055, China
8. School of Computing and Mathematical Sciences, University of Leicester, Leicester, LE1 7RH, UK

Received November 10, 2025; accepted January 11, 2026

E-mail: andrea.bracciali@unito.it

© The Author(s) 2026. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract

Selfish mining poses a serious threat to PoW(Proof of Work)-based cryptocurrencies. It undermines the incentive mechanism on which cryptocurrencies are based and enables attackers to obtain more rewards than they deserve. However, existing selfish attack schemes are either easily detectable or limited in their effectiveness: the former results in high and obvious forking rates, while the latter sacrifices revenue for covertness. In this paper, we propose a hybrid selfish mining attack, in which attackers covertly earn more with an optimised mining strategy (i.e., intelligently determining the number of blocks to be released in the private chain). Theoretical analysis and simulation results based on the MDP (Markov decision process) stochastic model show that the hybrid selfish mining attack reduces the forking rate considerably (e.g., 14.45%) with only a small loss of revenue (e.g., 4.2%). Additionally, we conduct comprehensive experiments on the implementation costs and actual rewards of the hybrid selfish mining attack. The results demonstrate its effectiveness across almost all PoW-based cryptocurrencies. Finally, we argue that the hybrid selfish mining attack may be mitigated by new consensus mechanisms (e.g., Proof of Stake) or standard countermeasures.

Keywords

selfish mining attacks; network security; optimized releasing strategies; Markov decision process

1 Introduction

Cryptocurrencies, with Bitcoin being a prime example, are subject to a volatile market due to their fluctuating value. Normally, in PoW (Proof of Work) blockchains, miners generate/mine a new block by solving a puzzle with their computing power [1,2]. The miners who mine a new block obtain absolute rewards, and coherently with the market value of the currency, may have a substantial value. In general, the rewards are proportionate to miners' computing power. This method of distributing rewards in equal proportions is relatively fair, and it encourages miners to participate in mining and maintains a benign economic ecology of the blockchain system [3]. However, a group of miners may also enhance their computing power by forming a mining pool, where many miners aggregate their computing power to mine as a solo miner and reallocate the reward.

With more power, the attacker may launch selfish mining attack (SM1) [4] by exploiting the vulnerabilities of PoW consensus mechanisms [5]. In particular, an attacker does not immediately release mined blocks as honest miners do, but keeps them private, exploiting the advantage of mining on a potential main chain that nobody else knows yet. Then, the attacker strategically releases the private blocks, resulting in forks in the main chain. Consequently, the attacker invalidates the blocks mined by honest miners, thereby wasting their computing power and obtaining a higher relative reward. For the honest miners, the relative reward equals their proportion of computing power. While for the attacker, the relative rewards are higher than the proportion [6]. For example, suppose the total rewards for Bitcoin are 10,000 BTC, and the proportion of computing power for an attacker is 35%, where $\gamma = 0$. Here $\gamma = 0$ means that the honest miner always following the branch of the honest miners where there is a fork. If the attacker mines honestly, its

relative reward is 35% of the total reward (about 3,500 BTC). However, if the attacker launches SM1, its relative reward increases to 36.65% (about 3,665 BTC). In this case, the attacker is given an additional 165 BTC. Recall that the total rewards does not change.

In May 2018, the Monacoin blockchain was exploited by a selfish mining attack, leading to a loss of \$90,000 [7]. Although the benefits of the attack are attractive, the conditions for launching such attack are strict. Theoretically, the attacker holding more than 25% computing power of the whole system can profit by launching SM1 [4], which is out of the range of most solo miners [6,8–10]. Furthermore, SM1 can be easily detected since it induce an unnaturally high forking rate [11–16]. Note that other attacks also comprise the security of blockchain, e.g., FAW (fork after withholding) [17] and BWH (block with- holding attacks) attacks [18], where attackers submit invalid blocks to other attackers to gain higher rewards. The former can mine with other attackers, while the latter can not. This paper focuses solely on single-attacker scenarios and thus excludes FAW and BWHs attacks.

In this paper, we propose a novel selfish mining attack, called the one-time releasing attack, in which all private blocks are released at one time when the length of the private chain is longer than that of the public chain.In SM1, the attacker strategically releases one or two private blocks whenever their private chain is longer than the public chain in order to invalidate the honest miners' blocks. However, SM1 induces an unnaturally high forking rate, although the relative rewards are acceptable. Our seemingly simple improvement effectively reduces the forking rate while maintaining high rewards. We further improve the one-time releasing attack by turning it into a hybrid attacking model, which combines selfish mining and one-time releasing attacks. The main contributions of this paper are as follows.

- We propose a novel selfish mining attack, which we will refer to as the one-time releasing attack. In our attack, the attacker releases all private blocks at their convenience, provided their private chain is longer than 2. Simulation results for the one-time releasing attack indicate that it is difficult to detect due to its profile, while the rewards are disappointing.
- We then propose a hybrid selfish mining attack, where the attacker adopts the one-time releasing attack with probability ρ and the SM1 with probability $1-\rho$. Simulation results validate the soundness of our hybrid selfish mining attack. In addition, the hybrid attack achieves a lower forking rate and the rewards are higher than those of honest mining and the semi-selfish attack.
- We discuss how to implement and stage our hybrid selfish mining attack. Because most of the attackers do not meet the threshold requirements for a successful attack, we consider the feasibility of renting mining power for successfully launching the hybrid attack. We then compare the rental costs and net profits, showing how the hybrid selfish mining attack can be profitable even for smaller pools, making it viable in practice, unlike most other selfish mining attacks. Our analysis indicates that the proposed attack is practical and destructive, threatening the blockchain ecosystem.
- Finally, some countermeasures against the hybrid selfish mining attack are presented from the viewpoint of detecting mechanisms, consensus mechanisms, and renting costs.

The rest of this paper is organised as follows: Section 2 presents the basic settings and strategy for SM1. Section 3 presents the one-time releasing attack. Section 4 presents the proposed hybrid attack, including attacker strategies, state transitions and reward analysis. Section 5 presents the simulation results. Section 6 discusses the implementation and countermeasures to the hybrid attack. Section 7 provides more detail on related current work. Section 8 summarises the results and outlines future research directions.

2 Preliminaries

In this section, we first present basic knowledge on cryptocurrency mining in Subsection 2.1, and then describe the background of selfish mining as well as the parameters in Subsection 2.2.

2.1 Honest mining

In selfish mining, honest miners always follow the blockchain backbone protocol [19]. We briefly recall the strategies for honest miners:

- Honest miners always follow the main chain, when it is unique, i.e., they mine on the main chain, defined as "the longest" (i.e., the one associated to the most computational power as standard in Bitcoin). Also, they publish the block as soon as it is mined.
- Under our assumptions, we only consider forks involving two branches, branch-honest and branch-attack. When there is a fork (two branches with the same length), honest miners follow the branch-attack with probability γ and branch-honest $1-\gamma$.

2.2 Selfish mining

In this section, we describe how selfish mining attack is conducted, why it is effective, and the parameters used in this research.

- Mining power. The mining power of whole blockchain, i.e., honest miners and the attacker, is normalized to 1. Note that the attacker could be either a mining pool or a single miner with substantial mining power. Blockchain nodes comprise the honest miners and the attacker. Let $\alpha \in [0,0.5)$ denote the mining power of the attacker[1)] and $1-\alpha$ denote that of the honest miners.
- Public and private chains. The attacker may withhold mined blocks and maintain a private chain (including the dotted squares in Fig. 1(a) and in the following ones). The attacker is aware of the private chain and generally mines the private chain. Other than the private chain, the rest of the blockchain is a standard public chain. Honest miners are only aware of

[1)] We do not consider the case where the attacker controls the majority of the network, i.e., $\alpha \in [0.5,1]$, since in that case the network is compromised.

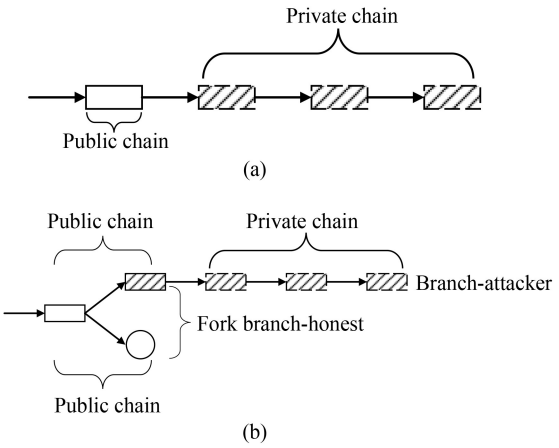


Fig. 1 An example of public and private chain. Circles denote the blocks mined by the honest miners. Squares with solid lines denote the blocks mined by the attacker and published. Squares with dotted lines denote the blocks mined by the attacker and withheld in the private chain. In the fork in (b), $\Delta = 3$ and the curly parenthesis labeled "private chain" cover the last three public blocks. (a) No forks in the public chain; (b) one fork in the public chain

the public blocks. For simplicity, we only illustrate the latest blocks in the public chain. For example, in Fig. 1(a) we only show the latest block in the public chain, while in Fig. 1(b) we have three public blocks, two of which are produced by the attacker. All these blocks are of interest here as they illustrate a fork (from honest miners' view).

- Main chain and branches. Generally, the blockchain is a tree, whose "longest" public chain is considered the main chain. In case of a fork we generally have two public chains with equal length. We use the notion of branches when there is a fork in the public chain (we consider the possibility of only two branches in the public chain[2)]). In Fig. 1(b), we call branch-attacker the branch derived by the latest contribution of the attacker's private chain (including the three squares with dotted lines), and branch-honest the branch lastly developed by honest miners (including the circle).
- The difference between the private and public chain. The strategies of the family of selfish mining are defined according to the difference between the length of the branch-attack and branch-honest $\Delta = l_a - l_h$, where l_a denotes the length of branch-attack, and l_h denotes that of the branch-honest. So, $\Delta = 3$ in Fig. 1(b), and $\Delta = 3$, as well, in Fig. 1(a) (no fork).
- Forks. A natural fork will occur when two miners simultaneously append their newly mined block to the same block. Natural forks (usually ignored) are inevitable in blockchain since the probability of their occurrence is very small[3)]. In a selfish mining setting, artificial forks are those that may be generated by the attacker strategically releasing blocks from the private chain. These forks cannot be ignored because of their higher probability of occurrence. In the following sections, forks refers to artificial forks, if not otherwise specified, and forks can always be detected by honest miners. It is worth noting that the attacker does not generate forks in its private chain.

  The forking rate ($fork_{\text{rate}}$) denotes the ratio of the number of forks ($fork_{\text{num}}$) to the total number of blocks ($blocks_{\text{total}}$). Their relationship is shown in Eq. (1).

$$fork_{\text{rate}} = fork_{\text{num}}/blocks_{\text{total}}. \quad (1)$$

- Strategies. Selfish mining is a mining strategy that departs from the underlying protocol in blockchain, where the attacker either build a private chain or release the blocks thereof according to Δ [4,8]. Specifically, when $\Delta<0$, the attacker abandons its private chain. When $\Delta> = 0$ and the attacker mines a new block, the attacker set up a new private chain. When $\Delta = 1$ and the honest miners appends its new block to the chain, the attacker releases the new block at once to make a fork. When $\Delta = 2$ and the honest miner appends its new block to the chain, the attacker release two blocks in the private chain to establish a lead, rendering the block of the honest miner invalid. When $\Delta>2$ and the honest miner appends its new block to the chain, the attacker only releases the oldest block in the private chain and keep other blocks secret.

3 The one-time releasing attack in bitcoin

In SM1 and its variants, an attacker will only release the oldest block in the private chain—even if they have several hidden blocks—when an honest miner publishes a new block that generates a fork. These forks are caused by the attacker strategically publishing blocks from the private chain in an attempt to increase their relative reward and protect the private chain that has already been developed. In [21], the authors propose semi-selfish mining to reduce the forking rate by allocating some of the attacker's power to honest miners. While this method is effective in reducing forks, it reduces the attacker's mining power. From the attacker's perspective, eliminating such forks generated each time they publish the oldest private block would be effective.

3.1 One-time releasing attack

We propose and discuss the one-time releasing attack: the attacker publishes all blocks currently in the private chain when $\Delta > 2$ and the honest miners find a new block. Recall that in SM1, the attacker release only the oldest block in the private chain, inducing a fork. Thus, there are no more forks in the view of honest miners, since the main chain is immediately identified. Consequently, one strength of such an attack is to maintain a stable blockchain economic system, where the chances of the attack being detected by unnatural forking rates are low, and the attacker could make a profit in the long run.

Following the paradigm of modeling selfish mining (e.g., SM1

[2)] There may be more than two branches in the public chain. However, the probability of this case is small enough to be ignored for our purposes.
[3)] For instance, an average of 169 forks every 10, 000 mined blocks have been observed in some experiments for the Bitcoin blockchain [20].

[4,8] and semi-selfish mining [21]), we describe the one-time releasing attack through a MDP (Markov decision process), where each state is denoted by the parameter Δ. The MDP is shown in Fig. 2. The numbers in polygon, circle and triangle denote the states. The arrows denote the state transition, the number over which denotes the transition probability.

States 0', 0, 1, 2, 3, 4, 5 are defined according to the values of Δ. States 0 and 0′ denote the scenario where branch-attack and branch-honest have the same length. While state 0 represents the case of no fork (and no private chain), state 0′ contains a fork with branch-honest and branch-attack of the same length. States 1, 2, 3, 4, 5 denote the scenario of Δ = 1, 2, 3, 4, 5, respectively and we do not discuss the cases where Δ>5 since the probability of this case is negligible. It is worth noting that the states with a negative Δ are not of interest here: branch-attack would be shorter than the current branch-honest, and therefore the attacker would benefit from immediately abandoning its current private chain [8] and mining from the top of the main chain, where $\Delta = 0$.

Transitions are defined according to the actions of the attacker, who releases its private chain according to the one-time attack releasing strategy (refer to Table 1). For instance, transitions from state 2 to state 3 are defined by the following action, where the attacker mines another block for the private chain with probability α. In general, a path where several transitions happen in sequence. For example, path 1-0'-0 denotes two transitions, first transition from state 1 to state 0′ and then from state 0′ to state 0. More specifically, in state 1, the attacker first publishes its one private block, reacting to a block published by honest miners, and creates a fork (to state 0′). Consequently, in state 0′, either a new block is discovered and published by the attacker (with probability α), or a new block is published by the honest miners (with probability $1-\alpha$). In the latter case, either the block is appended to the branch-attacker (with probability γ) or to the branch-honest (with probability $1-\gamma$). In all cases, the fork gets resolved.

Although any state could theoretically be the initial state, we set state 0 to be the initial state for simplicity (a polygon in Fig. 2). To work with finite models, as MDPs are, we set state 5 as terminal (a triangle in Fig. 2). This is justified since the probability to reach further states, e.g., state 6, 7, ..., where the attacker has developed a private chain of length bigger than that of state 5, is negligible for realistic values of α, as confirmed by our simulations (shown in Fig. 3, probabilities of some states have not been plotted for simplicity, and simulation details are in Subsection 5.1). Technically, we model this with the cycling arc in state 5 on the basis that the possibility of the attacker mining a 6th block is negligible and has no

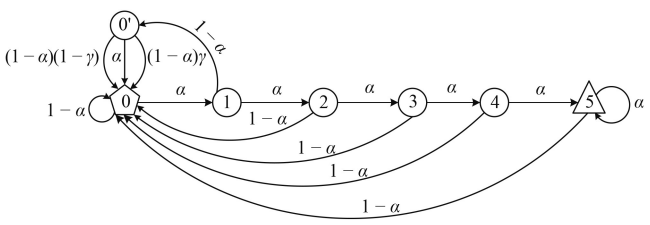


Fig. 2 The MDP of the one-time releasing attack

Table 1 The transition probability and rewards of the one-time attack

Case	$s' \to s$	$tr_{s'\to s}$	$p^A_{s'\to s}$	$p^H_{s'\to s}$
1	$0' \to 0$	α	2	0
2	$0' \to 0$	$(1-\alpha)\gamma$	1	1
3	$0' \to 0$	$(1-\alpha)(1-\gamma)$	0	2
4	$0 \to 0$	$1-\alpha$	0	1
5	$0 \to 1$	α	0	0
6	$1 \to 0'$	$1-\alpha$	0	0
7	$1 \to 2$	α	0	0
8	$2 \to 0$	$1-\alpha$	2	0
9	$2 \to 3$	α	0	0
10	$3 \to 4$	α	0	0
11	$3 \to 0$	$(1-\alpha)\rho$	3	0
12	$4 \to 5$	α	0	0
13	$4 \to 0$	$(1-\alpha)\rho$	4	0
14	$5 \to 0$	$(1-\alpha)\rho$	5	0
15	$5 \to 5$	α	1	0

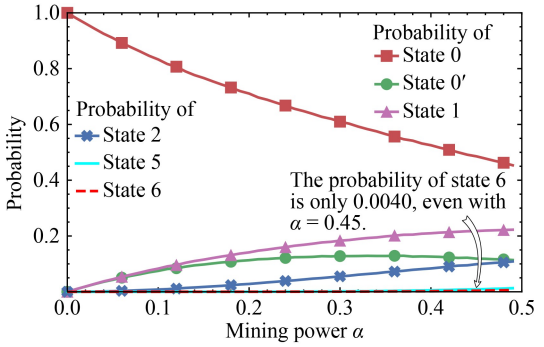


Fig. 3 The states' probabilities in the one-time releasing attack

effect on the system's states. Within the scope of this paper α ranges in $[0, 0.5)$, since we are interested in non-compromised blockchains, with a majority of honest mining nodes. The corresponding rewards are shown in Table 1. Each action induces rewards for the honest miners and the attacker. Refer more details about the rewards to Subsection 4.3.

3.2 Evaluation of the one-time releasing attack

The stochastic model has been used in stochastic simulations to ascertain the efficacy of the proposed attack in terms of relative reward and forking rate. We compare the one-time releasing attack with SM1, honest mining, and semi-selfish mining concerning the forking rate in Fig. 4 and the relative reward in Fig. 5, respectively.

Figure 4 presents the forking rate versus the attacker mining power. As illustrated, the forking rates of SM1 and the one-time releasing are not much different when $\alpha < 0.16$, however, for $\alpha > 0.16$, the one-time releasing attack exhibits better performance.

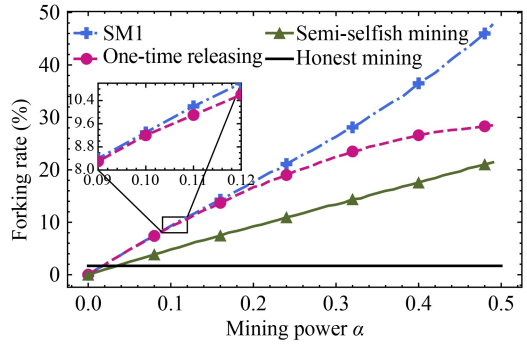


Fig. 4 The forking rate of the one-time releasing attack

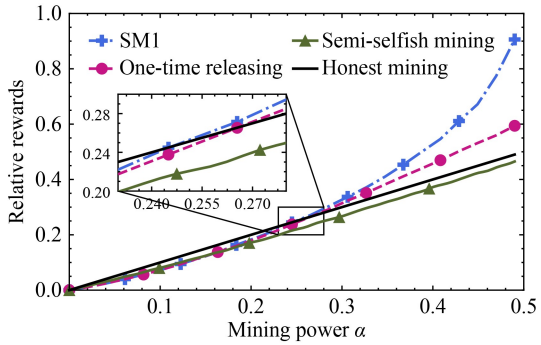


Fig. 5 The relative reward of the one-time releasing attack

Unsurprisingly, semi-selfish mining performs well even with a higher α. However, the performance of the forking rate is of interest for the attack only when compared with the relative reward metrics.

In addition, simulation results indicate that the profit threshold for SM1 and one-time releasing, i.e., when they become more convenient than honest mining, are at $\alpha = 0.25$ and $\alpha = 0.265$ respectively, as shown in Fig. 5. That is, the profit thresholds are close for these two attacks and their relative reward is even closer when the mining power is lower than these thresholds. However, while SM1 performs well when the mining power is larger than the threshold, the semi-selfish mining still exhibits a weak, far from expectations, relative reward when compared with the other attacks.

In general, if attackers (e.g., one-time releasing) want to decrease the possibility of being detected, then they need to maintain the average forking rate within an acceptable range. Our results may inform the choice of α, also according to the target maximum forking rate that an attacker is prepared to accept. Suppose, for instance, a target of 20%, then in the case of a one-time releasing attack, $\alpha = 0.25$ will already cause the target forking rate (Fig. 4). If the attacker's mining power is less than 0.25, however, the relative reward of the attacker may be equal to or even lower than that of honest mining (Fig. 5). Recall that the relative reward for honest miners is proportional to their mining power. In order to obtain a higher relative reward, the attacker needs to increase their mining power and act maliciously. As a result, the forking rate will also increase. In the considered attacks, a low forking rate and a high relative reward cannot coexist. Based on these experiments, we propose a new attack model that strikes a better balance between the forking rate and relative reward.

4 Evolving one-time releasing attack: the hybrid selfish mining attack

In this section, we propose the hybrid selfish mining attack, aiming at a trade-off between forking rate and relative rewards. Strategies of honest miners are defined the same as those in Subsection 2.1.

4.1 The hybrid attacking strategies

The attacker's strategies are derived from SM1 and the one-time release attack. The basic idea is to allow the attacker to choose between SM1 and the one-time release attack with a certain probability. A hybrid attacker must decide on their strategy when they have mined a new private block or when honest miners publish a new block. The specific strategies for the attacker are defined according to Δ, the difference between the length of the branch attack and branch-honest. Note that Δ and Δ' refer to the difference before and after the attacker adopts the specific action respectively. The strategies for the hybrid attacker are as follows.

1) Strategies for appending a newly mined private block. The attacker appends his newly mined blocks to the blockchain according to the following strategies.

- Abandon ($\Delta < 0$). As soon as the attacker is in a disadvantageous position when branch-honest is longer than branch-attack, the attacker abandons current efforts to further extend its private chain, and starts to mine on the honest-branch. Indeed, when $\Delta < 0$, the most optimistic possibility for the attacker would mine enough private blocks to make the private chain as long as the private chain. That is, $\Delta' = 0$. But even in such a case, it would not be easy for the attacker, with generally a minority of mining power, to overcome the main branch-honest, which has already received mining power from the majority of participants. Therefore, the attacker conservatively abandons the current private chain and chooses to set up a new private chain extending (the latest block of) branch-honest (like any other miner).
- Create ($\Delta = 0$). In this case, either there is no private chain, in which case the attacker mines a new block and creates a new private chain, or there is a branch attack and a branch-honest chain of the same length, in which case the attacker keeps extending the private chain. In both cases, $\Delta' = 1$. Generally, unless they are in a disadvantaged position, the attacker will keep building their private chain by adding newly mined blocks to it in order to gain an advantage.
- Wait ($\Delta > 0$). The attacker always appends new blocks to the private chain if the length of the private chain is bigger than the branch-honest in the public chain, leading to $\Delta' = \Delta + 1$.

2) Strategies for publishing private blocks, if any, after a new block has been published by honest miners. The attacker aims to maintain an advantageous position, minimizing forks. When honest miners publish a new block in the public chain, the attacker optimizes the number of released private blocks to avoid disadvantaged positions.

- Abandon ($\Delta < 0$). If the public chain (branch-honest) is longer than the private chain after the addition of a new block by

honest miners, the attacker abandons the existing and "losing" private chain, as well as any possible new block developed on that chain. This leads back to $\Delta' = 0$ since the attacker starts mining on the public chain, branch-honest.

- Keep ($\Delta = 0$ and no private chain). There are no private blocks to publish and the attacker keeps mining on the public chain trying to start a private chain, $\Delta' = 0$.
- Match ($\Delta = 0$ and private chain exist). This is the case where the advantage of a branch attack is lost when a newly published block vanishes. The attacker publishes the oldest block from the private chain (there is only one block in this case), creating a fork and offsetting the risk of losing an advantageous position. $\Delta' = 0$. Please note that this corresponds to state $0'$ in Fig. 6. Honest miners will now be spread over the two branches, and the branch attack could potentially obtain the majority of the mining power.
- Match-one ($\Delta = 1$). Honest miners have added a block to the public chain and the attacker branch has two private blocks. If it had just one, the addition of a new block would lead to $\Delta = 0$ via Match and the attacker publishes both blocks to avoid losing an advantageous position. The attacker branch now becomes the main chain and $\Delta' = 0$, with no honest branch.
- Waver ($\Delta >= 2$). the attacker publishes all blocks from the private chain (more than two blocks) with probability $\rho \in [0, 1]$ and publishes only the oldest block in the private chain with probability $1-\rho$. The attacker has a clear advantage and may opt for an all-in strategy to minimise forks or a more conservative strategy from SM1, which preserves some advantage and potentially generates more revenue (at the cost of potentially higher fork rates).

We specify the hybrid selfish mining strategy in Algorithm 1. It is worth remarking that the core of hybrid attack is the Waver strategy, which is presented in Algorithm 1, lines 21−27. In this module, the attacker uniformly samples a number from 0 to 1. If the number is smaller than the parameter ρ, the one-time releasing strategy is adopted to release all private blocks, otherwise the SM1 strategy is followed. Specifically, strategies are identical to SM1 when $\rho = 0$. That is, the attacker publishes the oldest block in the private chain, leading to a fork in the public chain and $\Delta' = \Delta - 1$. On the other hand, strategies are identical to one-time releasing when $\rho = 1$. The intuition of the novel strategy is to reduce the forks in the public chain at an accepted reward loss. Hence, the definition of ρ is critical.

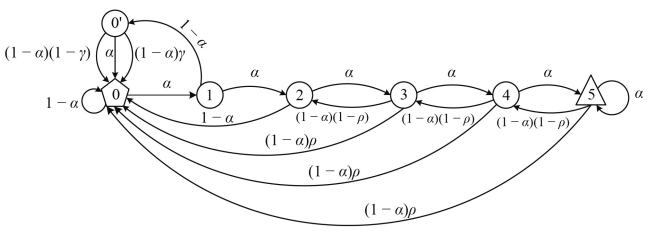


Fig. 6 The MDP of the hybrid attack model

Algorithm 1 The hybrid selfish mining attack

```
1  Initialize parameters α, γ, ρ;
2  branchAttack, branchHonest ← 0, 0;
3  for i ← 0 to N do
4      δ ← branchAttack − branchHonest;
5      miner ← miningProcess(α);
6      if miner ="attacker" then
7          branchAttack ← branchAttack + 1;
8          if δ = 0 & branchAttack = 2 then
9              publish all private blocks;
10             branchAttack, branchHonest ← 0, 0;
11         // continue to mine on the private chain;
12     else // miner = "honest"
13         branchHonest ← branchHonest + 1;
14         if δ = 0 then
15             publish the private block;
16             // blockchain is forking, honest miners mine on one
                 branch with probability γ;
17             branchAttack, branchHonest ← 0, 0;
18         else if δ = 2 then
19             publish the private blocks;
20             branchAttack, branchHonest ← 0, 0;
21         else // δ > 2
22             r ← random.uniform(0,1);
23             if r < ρ then
24                 releasing all private blocks;
25                 branchAttack, branchHonest ← 0, 0;
26             else
27                 publish the oldest block in private chain;
```

4.2 Attacker stochastic model

Both the attacker and the honest miners mine according to their respective mining power and strategies. We collectively refer to the length and structure of the evolving blockchain as the state of the blockchain. In the definition of a stochastic model for the attacker, the state of the blockchain is again abstracted away as the value of Δ, i.e., the difference between the length of branch-attack and branch-honest. The state of the blockchain evolves with a certain probability as mining progresses.

Similarly to the one-time releasing attack, the hybrid selfish mining attack is modeled as an MDP, shown in Fig. 6. Some considerations and reminders are presented here.

- As before, we assume that the maximum length of the private chain is 5 since the probability of reaching a state with a 6-block advantage is regarded as negligible for the values of α of interest in this paper. Informally speaking, the looping transition in 5 will basically "rarely" be reached and we consider it has no effect.
- In Bitcoin, the "longest" public chain is the main chain. If the longest public chain is not unique, we can assume that a published branch attack and an honest branch of the same length will coexist temporarily. In this case, honest miners will choose to mine one or the other (as they cannot

distinguish between them), rather than shorter branches. Given the low probability of multiple longest chains existing, it is reasonable to ignore information about multiple honest branches in the model's states; states only need to consider the difference between branch attacks and branch honesty. This enables us to combine states with the same Δ in the stochastic model.

Figure 6 shows the state transitions in the hybrid attack model, with each state labeled with a value of Δ. It is worth recalling that state $0'$ denotes a state with a fork, branch-honest and branch-attack have the same length and there is no private chain.

Considering state 1, for example, there are three possible transitions:

- State $0 \rightarrow$ state 1. In state 0, there is no fork and no private chain. When the attacker mines a new block with probability α, it does not publish it on the public chain, but generates a new private chain of length 1, resulting in $\Delta' = 1$ (state 1).
- State $1 \rightarrow$ state $0'$. In state 1, honest miners mine and append a new block to the public chain with probability $1-\alpha$. In this case, the attacker publishes the oldest (the only one) block from its private chain, resulting in a fork with equal length ($\Delta' = 0$, state $0'$).
- State $1 \rightarrow$ state 2. In state 1, the attacker mines and appends a new block to its private chain with probability α, leading to $\Delta' = 2$ (state 2).

Other transitions are defined analogously.

4.3 Theoretical analysis of the hybrid attack: rewards

Honest miners and attackers obtain rewards by mining blocks in the blockchain. We measure reward in terms of the number of mined blocks.

The reward of the attacker $R^A = \sum r_s^A$ (honest miners $R^H = \sum r_s^H$) is the sum of the revenue from each state $s \in \{0', 0, 1, 2, 3, 4, 5\}$. More precisely, the attacker revenue r_s^A (r_s^H for the honest miners) associated to a state s, depends on three factors for each state s' leading to state s:

- The probability $Pr_{s'}$ of reaching a state s'.
- The probability $tr_{s' \rightarrow s}$ of the transition from state s' to state s.
- The attacker reward $p_{s' \rightarrow s}^A$ ($p_{s' \rightarrow s}^H$ for the honest miners) associated with that transition, i.e., +1 every time the attacker publishes a block on the main chain.

For instance,

$$\begin{aligned} r_2^A &= Pr_1 \cdot tr_{1\rightarrow 2} \cdot p_{1\rightarrow 2}^A + Pr_3 \cdot tr_{3\rightarrow 2} \cdot p_{3\rightarrow 2}^A \\ &= Pr_1 \cdot \alpha \cdot 0 + Pr_3 \cdot (1-\alpha)(1-\rho) \cdot 1. \end{aligned} \tag{2}$$

In other words, the revenue for the attacker in state 2 includes the rewards for the transitions from states 1 and 3 to state 2. In the former case, the attacker finds a new block but does not publish it in the public chain. Therefore, the reward for the attacker is 0. In the latter case, the attacker receives reward 1 when state 3 transitions to state 2 by publishing the oldest block in the private chain. Despite the presence of a fork in the public chain, the branch attack will eventually prevail as the main chain because the attacker has an absolute lead in the private chain.

The transition probabilities are presented in Fig. 6. The rewards for the attacker and the honest miners are presented in Table. 2. The rewards for the attacker and the honest miners are derived according to state transitions. The first column denotes the transition from state s' to state s. The second column denotes the probability $tr_{s' \rightarrow s}$ of the transition. The third and the fourth columns denote the reward for the attacker $p_{s' \rightarrow s}^A$ and the honest miners $p_{s' \rightarrow s}^H$, respectively. It is worth noting that the rewards reflect the notion of finality, whereby blocks, and therefore rewards, are considered final as soon as it is possible to identify the prevailing chain, the public chain, that is longer than all the others. In our model, this chain cannot be superseded by extending a shorter chain.

The rewards for each case are as follows.

- In case 1, the attacker appends and publishes the newly mined block to the branch attack, which makes the branch attack the main chain. The attacker gets 2 rewards, while honest miners get 0 rewards.
- In case 2, honest miners append their new block to the branch attack, which validates the oldest block in the branch attack. Therefore, both the honest miners and the attacker gets 1 reward.

Table 2 Rewards for the attacker and honest miners according to transitions

Case	$s' \rightarrow s$	$tr_{s' \rightarrow s}$	$p_{s' \rightarrow s}^A$	$p_{s' \rightarrow s}^H$
1	$0' \rightarrow 0$	α	2	0
2	$0' \rightarrow 0$	$(1-\alpha)\gamma$	1	1
3	$0' \rightarrow 0$	$(1-\alpha)(1-\gamma)$	0	2
4	$0 \rightarrow 0$	$1-\alpha$	0	1
5	$0 \rightarrow 1$	α	0	0
6	$1 \rightarrow 0'$	$1-\alpha$	0	0
7	$1 \rightarrow 2$	α	0	0
8	$2 \rightarrow 0$	$1-\alpha$	2	0
9	$2 \rightarrow 3$	α	0	0
10	$3 \rightarrow 2$	$(1-\alpha)(1-\rho)$	1	0
11	$3 \rightarrow 4$	α	0	0
12	$3 \rightarrow 0$	$(1-\alpha)\rho$	3	0
13	$4 \rightarrow 3$	$(1-\alpha)(1-\rho)$	1	0
14	$4 \rightarrow 5$	α	0	0
15	$4 \rightarrow 0$	$(1-\alpha)\rho$	4	0
16	$5 \rightarrow 4$	$(1-\alpha)(1-\rho)$	1	0
17	$5 \rightarrow 0$	$(1-\alpha)\rho$	5	0
18	$5 \rightarrow 5$	α	1	0

- In case 3, honest miners append their new block to the branch-honest, which validates the oldest block in the branch-honest. Therefore, honest miners get 2 rewards, while the attacker gets 0 rewards.
- In case 4, honest miners append one block to the public chain and get 1 reward, while the attacker gets 0.
- The reward for the attacker and honest miners is 0 in case 6, where two branches have the same length. In this case, it is not sure which branch will end up as a final public chain and the reward is 0.
- In cases 10, 13, and 16, the attacker publishes the oldest block in the private chain to get 1 reward since the branch attack will eventually be selected as the main chain.
- The rewards for the attacker and honest miners are 0 in cases 5, 7, 9, 11, and 14, where one block is appended to the private chain without being published.
- In cases 8, 12, 15, and 17, the attacker publishes 2,3,4, and 5 blocks of the private chain and gets the corresponding 2,3,4, and 5 rewards, respectively.
- The transition $5 \rightarrow 5$ in case 18 has been introduced to make the stochastic model finite, given that the probability of moving to a possible state 6 is negligible. We decid to grant the attacker a reward if this negligible-probability transition is fired, to summarise the possible reward that has been cut when approximating the system with a finite model. Even if this is a somehow arbitrary approximation choice, its impact, either way, is not overall significant, given its negligible probability.

In the following, we show how to compute the probability Pr_s of reaching state s, where $s \in \{0', 0, 1, 2, 3, 4, 5\}$. The probability of each state can be derived from the transition relationship between the states. For example, state 0 can be reached from state 0 itself with probability $1-\alpha$, from state $0'$ with probability α, $(1-\alpha)\gamma$, and $(1-\alpha)(1-\gamma)$ [4], from state 2 with probability $1-\alpha$, and state 3, 4, and 5 with probability $(1-\alpha)\rho$. Therefore, following a standard approach as in [17,22], the probability Pr_0 can be expressed as:

$$\begin{aligned} Pr_0 = {} & (1-\alpha)\cdot Pr_0 + (1-\alpha)\cdot Pr_2 \\ & + [\alpha + (1-\alpha)\gamma + (1-\alpha)(1-\gamma)]\cdot Pr_{0'} \\ & + (1-\alpha)\rho\cdot[Pr_3 + Pr_4 + Pr_5]. \end{aligned} \tag{3}$$

And so on in a similar fashion, the recursion formula for Pr_s is as follows:

$$\begin{aligned} Pr_{0'} &= (1-\alpha)\cdot Pr_1, \\ Pr_1 &= \alpha\cdot Pr_0, \\ Pr_2 &= \alpha\cdot Pr_1 + (1-\alpha)(1-\rho)\cdot Pr_3, \\ Pr_3 &= \alpha\cdot Pr_2 + (1-\alpha)(1-\rho)\cdot Pr_4, \\ Pr_4 &= \alpha\cdot Pr_3 + (1-\alpha)(1-\rho)\cdot Pr_5, \\ Pr_5 &= \alpha\cdot Pr_4 + \alpha\cdot Pr_5. \end{aligned} \tag{4}$$

In addition, we have the constraint:

$$Pr_{0'} + \sum_{i=0}^{5} Pr_i = 1. \tag{5}$$

Then the probability for each state, by solving the Eqs. (3), (4), and (5), can be expressed as follows:

$$\begin{aligned} Pr_0 &= \frac{\mathcal{M}}{\mathcal{H}}\cdot(1-\alpha), \\ Pr_{0'} &= \frac{\mathcal{M}}{\mathcal{H}}\cdot\alpha(1-\alpha)^2, \\ Pr_1 &= \frac{\mathcal{M}}{\mathcal{H}}\cdot\alpha(1-\alpha), \\ Pr_2 &= \frac{\alpha^2[1+\alpha(\alpha-2)(1-\rho)]}{\mathcal{H}}\cdot(1-\alpha), \\ Pr_3 &= \frac{\alpha^3(\alpha\rho-\alpha+1)}{\mathcal{H}}\cdot(1-\alpha), \\ Pr_4 &= \frac{\alpha^4}{\mathcal{H}}\cdot(1-\alpha), \\ Pr_5 &= \frac{\alpha^5}{\mathcal{H}}, \end{aligned} \tag{6}$$

where $\rho \in [0,1]$,

$$\begin{aligned} \mathcal{M} = {} & \alpha^3(2\rho-\rho^2-1) + \alpha^2(\rho^2-4\rho+3) \\ & + 3\alpha(\rho-1) + 1, \end{aligned} \tag{7}$$

and

$$\begin{aligned} \mathcal{H} = {} & \alpha^6(2\rho-\rho^2-1) + \alpha^5(4\rho^2-10\rho+6) \\ & + \alpha^4(15\rho-4\rho^2-11) + \alpha^3(10-9\rho) \\ & + \alpha^2(\rho^2-\rho-2) + \alpha(3\rho-2) + 1. \end{aligned} \tag{8}$$

Given the transition probabilities, rewards, and the probability of each state in Table 2 and Eq. (6), the rewards for the attacker (R^A) and honest miners (R^H) can be presented in Eqs. (9) and (10). Note that we only sum the non-zero rewards, for brevity.

$$\begin{aligned} R^A = {} & Pr_{0'}\cdot\alpha\cdot 2 + Pr_{0'}\cdot(1-\alpha)\gamma\cdot 1 + Pr_2\cdot(1-\alpha)\cdot 2 \\ & + Pr_3\cdot(1-\alpha)(1-\rho)\cdot 1 + Pr_3\cdot(1-\alpha)\rho\cdot 3 \\ & + Pr_4\cdot(1-\alpha)(1-\rho)\cdot 1 + Pr_4\cdot(1-\alpha)\rho\cdot 4 \\ & + Pr_5\cdot(1-\alpha)(1-\rho)\cdot 1 + Pr_5\cdot(1-\alpha)\rho\cdot 5 \\ & + Pr_5\cdot\alpha\cdot 1, \end{aligned} \tag{9}$$

$$\begin{aligned} R^H = {} & Pr_{0'}\cdot(1-\alpha)\gamma\cdot 1 + Pr_{0'}\cdot(1-\alpha)(1-\gamma)\cdot 2 \\ & + Pr_0\cdot(1-\alpha)\cdot 1. \end{aligned} \tag{10}$$

R^A and R^H are absolute rewards, i.e., they represent the share of the overall blocks mined by the attacker and honest miners, according to the computed probabilities. In general, it is worth considering the normalized relative reward $(R^A)'$ and $(R^H)'$ for the attacker and the honest miners:

$$(R^A)' = \frac{R^A}{R^A + R^H}, \quad (R^H)' = \frac{R^H}{R^A + R^H}. \tag{11}$$

As a sanity check of Eq. (11), we compare in Fig. 7 $(R^A)'$ and

[4] Here the probabilities α, $(1-\alpha)\gamma$, and $(1-\alpha)(1-\gamma)$ can not be merged in just one transition of probability 1 since we need to keep the transitions distinct to model the rewards for the attacker and the honest miners, which are distinct in the two transitions.

$(R^H)'$ with results of simulations of the system, with parameter ρ set to 0, 0.5, and 1.0. Analytical definitions $(R^A)'$ and $(R^H)'$ are consistent with probabilistic simulation data, with a minimal deviation in Fig. 7(a) when $\alpha > 0.40$. Note that the results shown in Fig. 7(a) describe SM1 since the attacker's strategy is equivalent to SM1 when $\rho = 0$. Analogously, Fig. 7(c) with $\rho = 1.0$ describes the one-time releasing strategy, because the one-time releasing attack is a special case of the hybrid attack at $\rho = 1.0$. Fig. 7(b) shows the reward distribution when the attacker performs a hybrid attack and sets $\rho = 0.5$.

5 Simulation of the hybrid attack

The simulation for the hybrid attack is similar to that of a one-time releasing. The main difference is that $\rho = 1$ is constant for the one-time releasing attack, while ρ varies between 0 and 1 for the hybrid attack. The analysis of simulation results for the hybrid attack, and its impact on the forking rate, relative reward, and threshold follows.

5.1 Settings and parameters

Settings. The simulation is implemented in Python (version 3.9.0). Since the simulation focuses on the logic of blockchain state evolution, regardless of the specific deployment environment of the blockchain. Therefore, the simulation can be run on all machines that support Python and will obtain the same results. That is, the simulation results have no relationship with hardware.

Parameters. The simulation is run for different combinations of parameters, each of which runs for 200,000 iterations. In each round, one new block is mined and ownership of the block is assigned by random sampling, proportional to the amount of mining power owned. The owner of each new block is determined using a Monte Carlo approach. Specifically, in each sampling, the attacker mines a block with probability $\alpha \in [0, 0.5)$, and honest miners do it with probability $1 - \alpha$. The attacker and the honest miners follow their respective strategies. For simplicity, we ignore the block propagation delay and verification time when participants follow these strategies. In other words, we ignore the differences in the communication capabilities of individual nodes in the network. If two blocks are mined in one round, each node will receive both with the same probability and then select one. During simulations, the frequency of each state is counted and then used to calculate the probability distribution of states. Finally, when the simulation is complete, the forking rate and relative reward are calculated according to Eq. (1) and Eq. (11), respectively.

For simplicity, we do not consider the impact of network latency on attacks, as numerous methods already exist to reduce such delays. For instance: Network Layer Optimization: Adopting more efficient P2P protocols (e.g., Gossip protocol) to reduce inter-node communication delays. Consensus Mechanism Improvements: Transitioning from PoW to PoS or hybrid mechanisms (e.g., Ethereum 2.0) to enhance block production efficiency. Sharding and Rollup Technologies: Distributing transactions across multiple shards for processing, combined with Rollups to batch transactions before submitting to the main chain, thereby boosting overall throughput. Economic Incentive Adjustments: Dynamically adjust transaction fee mechanisms to encourage users to pay higher fees during peak congestion periods for faster confirmation. Implement penalty mechanisms for validators (e.g., slashing staked tokens) to reduce authentication issues caused by intentional delays or offline states.

5.2 Forking rate of the hybrid attack

The simulation results show that the hybrid attack successfully lowers the forking rate. Although the attacker's reward loss is modest, the required attacking power threshold increases slightly. In other words, the hybrid attack achieves a low forking rate by sacrificing a small amount of reward, which proves worthwhile in the long term. By reducing the forking rate, the attack successfully avoids detection. Furthermore, it promotes the efficient operation of the blockchain economic ecosystem [23].

As mentioned above, a blind pursuit of high relative reward with a potentially high forking rate, prone to easy detection, e.g., [24], may jeopardize the stability of the blockchain. Normally, attackers might launch short-term benefit attacks, e.g., attacking different blockchains for short periods. Such type of attacks is out of scope here.

Figure 8 shows the forking rates of SM1, the hybrid attack and semi-selfish mining for different values of ρ. The semi-selfish mining rate is the lowest of these three rates. Note that the forking rate of honest mining is approximately 1.69% [20] (solid line in Fig. 8). The forking rate of the hybrid attack lies between that of SM1 and semi-selfish mining. More specifically, the forking rate of the hybrid attack is lower than that of SM1 when $\alpha > 0.18$. In

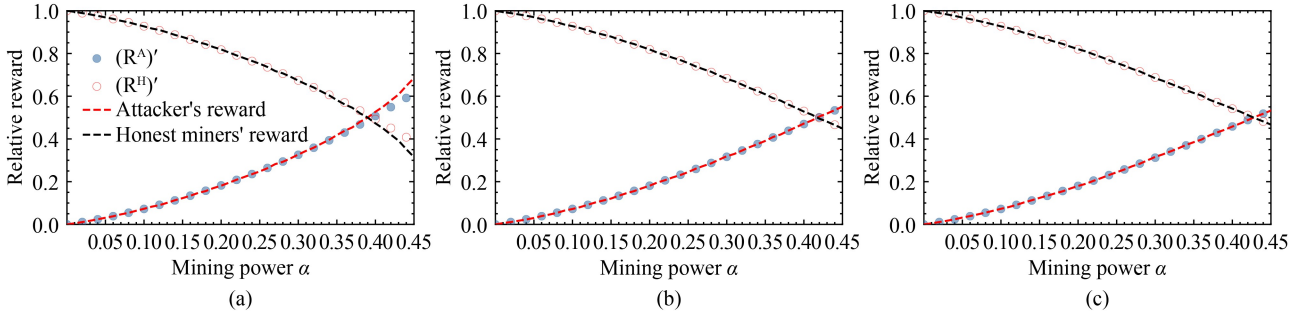


Fig. 7 The comparison of analytical results, $(R^A)'$ and $(R^H)'$, with simulation results (dashed line). The results shown in Fig. 7(a) are for the SM1 since the attacker's strategy is equivalent to SM1 when $\rho = 0$. Results shown in Fig. 7(c), when $\rho = 1.0$, are for the one-time releasing strategy because the one-time releasing attack is a special case of the hybrid attack at $\rho = 1.0$. (a) $\rho = 0$; (b) $\rho = 0.5$; (c) $\rho = 1.0$

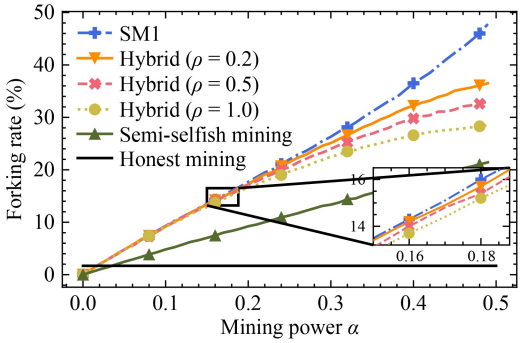


Fig. 8 The forking rate of hybrid attack compared with SM1 and semi-selfish mining

addition to being related to the mining power, the forking rate is also affected by ρ, as shown in Fig. 9. With a fixed α, the forking rate is inversely proportional to ρ. Therefore, if the attacker wants to obtain a lower forking rate, they should choose a larger value of ρ and a smaller value of α.

The parameter γ reflects the proportion of honest miners who follow the attacker's branch (i.e., branch-attack) when the blockchain forks. More honest miners mining on branch-attack will raise the attacker's chance of winning the fork racing. Hence, this parameter undoubtedly will affect the forking rate and miners' reward. We first discuss the impact of γ on forking rate, and then present the effect on rewards later (Subsection 5.3). Without loss of generality, we specify the attacker's power $\alpha = 0.3$, and then observe the impact of γ on the forking rate. The results, presented in Fig. 10, show that a large γ leads to an increased forking rate for both SM1 and the hybrid attack, especially, with a small ρ. In addition, γ has less influence when ρ is closer to 1 because a larger ρ results in a smaller number of forks.

5.3 The reward and threshold of the hybrid attack

The relative reward and the threshold are two indicators used to evaluate different types of attack. The former denotes the additional rewards received by an attacker when they launch an attack, while the latter denotes the minimum mining power required to launch an attack. Generally speaking, provided the forking rate remains within an acceptable range, the attacker will aim to maximise the long-term relative reward. We compare the hybrid attacking model, honest mining, SM1 and semi-selfish mining in terms of relative reward in

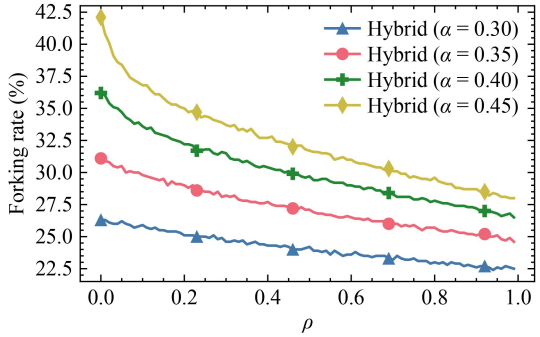


Fig. 9 Significant reductions in the forking rate of the hybrid attack, when ρ varies

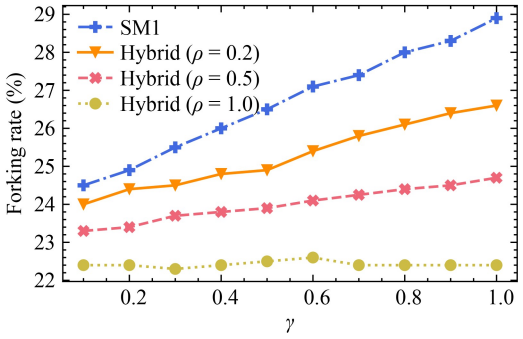


Fig. 10 Forking rate of the hybrid attack ($\alpha = 0.3$), when γ varies

Fig. 11. The relative reward of the hybrid attack is lower than that of the honest mining and the semi-selfish mining when α is smaller than the threshold 0.265 and 0.165, respectively. Compared with honest mining, the hybrid attack has a profit threshold of $\alpha = 0.26$ ($\alpha = 0.265$) when $\rho = 0.5$ ($\rho = 1.0$), which is only a slight threshold increase when compared with the threshold for SM1 of $\alpha = 0.25$.

Noticeably, the relative reward of the hybrid attack, shown in Fig. 11, almost coincides with that of SM1 when $\alpha \leqslant 0.23$. The figure shows that the variation of ρ does not have a significant impact on the attacker's relative reward when its mining power is smaller than the threshold.

In Fig. 12, the relative reward of the hybrid attack is shown to be proportional to α, while being inversely proportional to ρ, particularly in the lower range. Intuitively, the larger the value of α, the longer and more lucrative the attacker's private chains can be maintained. Crucially, the rewards of the hybrid attack far exceed

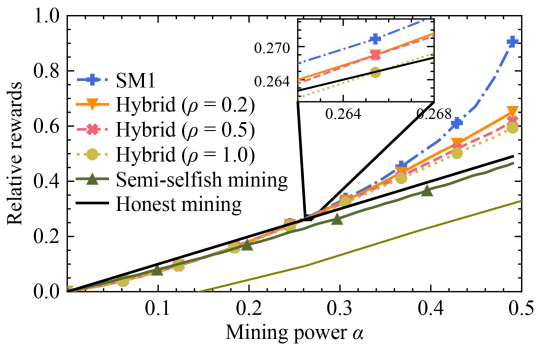


Fig. 11 The relative reward and the threshold compared with others

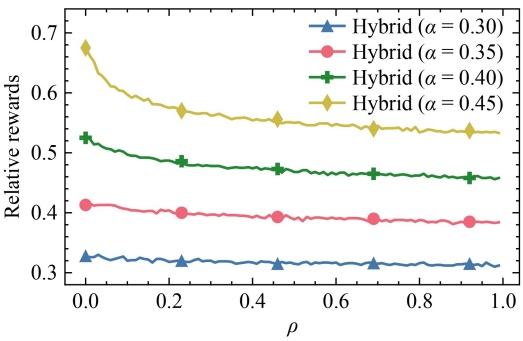


Fig. 12 The impact of ρ on the attacker's reward with different α

those of honest mining above the threshold. Therefore, if the attacker wants a high relative reward, they should choose a relatively small (but not too small) value of ρ and, clearly, a large value of α, up to the point where forking rates become intolerable.

Figure 13 illustrates the impact of γ on the attacker's relative reward. We can see that a larger γ (means more miners follow the attacker's branch when the blockchain forks) will improve the rewards both for SM1 and the hybrid attack (even with a large ρ). Therefore, on the basis of the hybrid attack, the attacker can integrate other strategies, e.g., the bribery strategy [25], to attract more honest miners to follow his branch, thereby increasing his relative reward.

5.4 The impact on chain quality of the hybrid attack

The chain quality property typically indicates that for any given number of consecutive blocks on the main chain, the proportion of blocks mined by honest nodes should exceed a specific threshold value [19]. Chain quality is a metric that measures the proportion of honest blocks on the longest chain within a blockchain network. Here, the "longest chain" refers to the chain currently containing the most blocks, while "honest blocks" denote blocks that have not been tampered with or illegally created, adhering to the blockchain's rules and consensus mechanism. High chain quality indicates that the vast majority of blocks are generated by honest miner, while low chain quality reflects significant illegal or tampering activities that may undermine the integrity and credibility of the blockchain. The impact of chain quality on blockchain security in 51% attack manifests primarily in the following aspect: When attackers control over 50% of the computational power (or stake), they can disrupt the system by tampering with transactions, double-spending, and so on. High chain quality indicates that honest nodes dominate, meaning that attackers would need to incur extremely high costs to control the network. For instance, in the Bitcoin network, controlling 51% of the computational power requires significant hardware and electricity expenditure, making such attacks unsustainable due to the continuous generation of honest blocks.

This property reflects the security of the blockchain. Operators can detect mining attacks by monitoring the decline of this property; therefore, we need to study the impact of a hybrid attack on it. To this end, we simulate a hybrid attack on the blockchain and record all the blocks generated during the attack. Once the attack has finished, we check all main chain blocks and identify the miner of each block in order to calculate the chain quality.

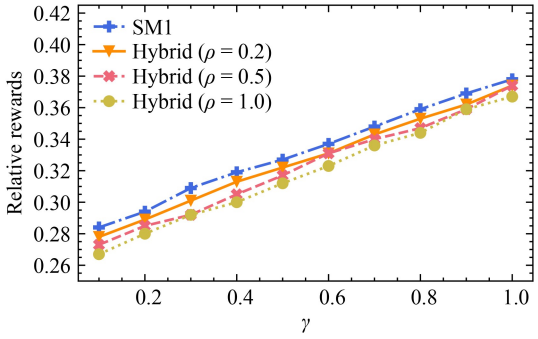


Fig. 13 Relative reward of the attacker ($\alpha = 0.3$) with different ρ and γ

Figure 14 shows the effect of mining power α on chain quality in hybrid attacks with different ρ. It can be seen that the chain quality decreases both in SM1 and hybrid attacks as the power α increases, while the SM1 strategy induces a more serious quality degradation, especially in high-power scenarios. In other words, in terms of chain quality monitoring, the SM1 strategy is easier to detect than our hybrid attack. This point can also be demonstrated in Fig. 16, as γ increases, the chain quality of both SM1 and the hybrid attack decreases. However, for any specific value of γ, the hybrid attack has better performance in chain quality property.

In Fig. 15, we evaluate the impact of the parameter ρ on chain quality in the hybrid attack. The results show that increasing the value of ρ improves chain quality. However, a large value of ρ will result in a loss of reward. Therefore, similar to the trade-off between forking rate and reward, the attacker needs to make a trade-off between chain quality and reward according to the actual situation.

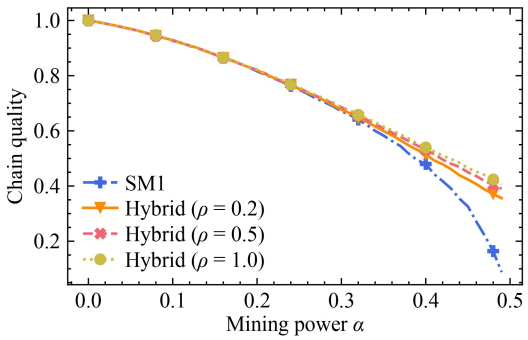


Fig. 14 The chain quality of the blockchain when α varies

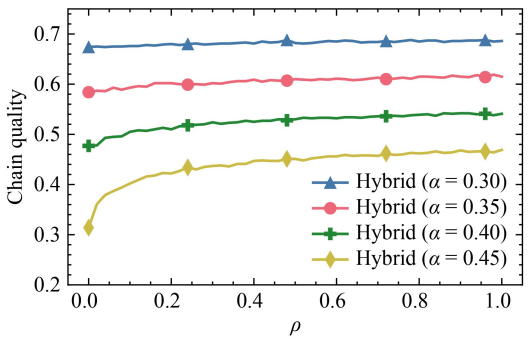


Fig. 15 The impact of ρ on chain quality with different α

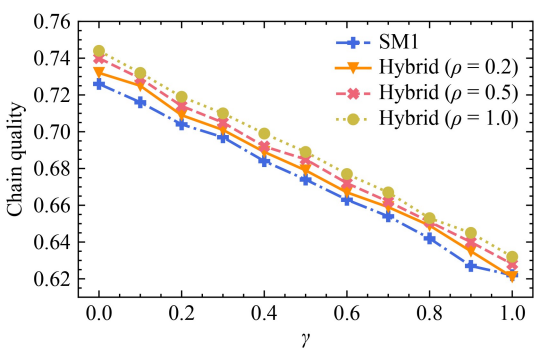


Fig. 16 The chain quality with different ρ and γ, when $\alpha = 0.3$

6 The feasibility and anti-measures of hybrid attack

6.1 On increasing attack's feasibility by renting mining powers

One of the notable features of attacks such as selfish mining is that they are difficult to implement in practice due to constraints on mining power. For example, AntPool, the largest one in Bitcoin, does not exceed the ideal threshold. Smaller pools, such as F2Pool and Poolin, have no incentive to launch such attacks, either. But does this mean that the system is immune to selfish mining attacks? Can the security of the system be guaranteed?

The above questions should be considered in relation to the possibility of increasing mining power by renting it, as well as the trade-off between costs and rewards in this case. In order to launch a selfish mining-like attack, the attacker's mining power must reach a certain threshold, such as 26.5% of the network's total mining power. This threshold is difficult for ordinary miners to reach, yet the substantial profits from such attacks are highly tempting. Consequently, attackers may consider renting computing power to launch such attacks. As Fig. 17 shows, renting enough computing power to reach 26.5% in mainstream mining pools currently yields significant returns. Therefore, attackers are strongly incentivised to rent computing power for their operations.

For example, the attacker may simply rent hashing power on a service like Nicehash for a few hours, which may make the cost/benefit significantly favorable, and the attack feasible. We investigate the cost and rewards of the selfish mining attacks for some pools[5)] under the hypothesis of buying hashing power on Nicehash, or similar platforms, to reach the mining power threshold, e.g., 0.265 for the hybrid attack. Here, we consider the hybrid attack with $\rho = 1.0$ for our simulations (this is a conservative choice, being $\rho = 1.0$ the less profitable of the hybrid attacks).

For example, the AntPool would rent 0.092 hashing power to reach the threshold for hybrid attacks. Costs and rewards are computed by fixing costs at a given date (as of 16:00 on September 13th, 2021), as well as the total mining power of the Bitcoin network, which was 133,731.55 PH/s. On Nicehash, the world's largest online trading platform for blockchain mining power, the price of renting 1 PH/s computing power was \$12.89 per hour. The market value of one bitcoin was \$46,108.37.

Let us consider the case of AntPool, which needs to rent 0.092 of the mining power to reach 0.27, and exceed the threshold of 0.265. In other words, AntPool currently has 133,731.55 PH/s × 0.178 ≃ 23,804.22 PH/s mining power, and requires renting an additional 133,731.55 PH/s × 0.092 = 12,303.3026 PH/s mining power.

We assume that AntPool launches a hybrid attack for 24 hours. Given the price, the rental cost is 12,303.3026 × \$12.89 × 24 ≃ \$3,806,150.

Assuming that a block is mined every 10 minutes, the whole network can mine 144 blocks in 24 hours. According to the percentage of AntPool's mining power and the attack strategy, if it launches the attack, AntPool can mine 39.347 blocks. Such results have been obtained by averaging 100,000 simulations. In each simulation, a pool equivalent to AntPool performs the hybrid attack (with $\rho = 1.0$), while the other miners behave honestly. The blockchain mines 144 blocks total on the longest public chain. Then, the number of blocks mined by both parties are counted separately (simulation scripts available at our repository, Subsection 5.1). Mining honestly, with reward proportional to its mining power, AntPool would have mined 38.88 blocks, that is the attack grants 39.347 – 38.88 = 0.467 extra blocks. With 6.25 bitcoins per block as a reward at \$46,108.37 per bitcoin, the hybrid attack (with parameter $\rho = 1.0$) grants an extra 0.467 × 6.25× \$46,108.37 ≃ \$134,578.80, every 24h.

Overall, AntPool's total reward over the 24h is 39.347 × 6.25 ×

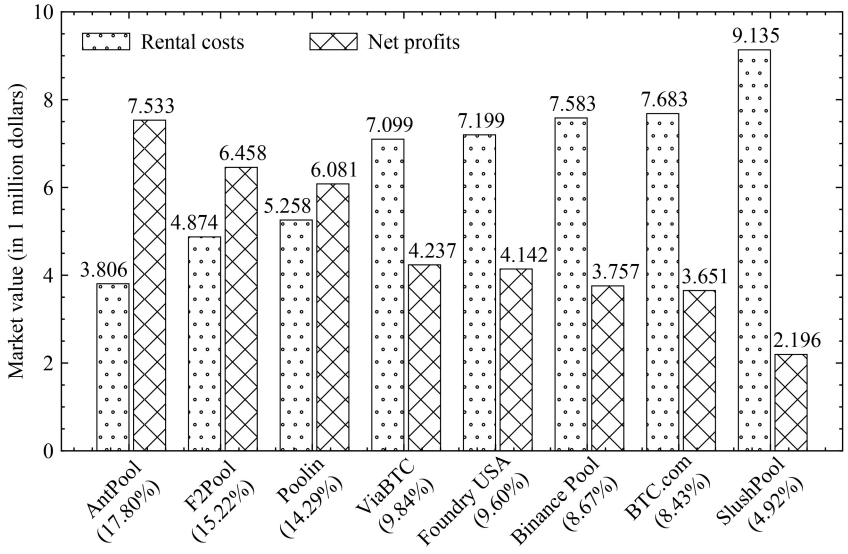


Fig. 17 The costs and net profits of major mining pools in Bitcoin. The value in parentheses is the hashing power fraction of mining pool

[5)] In this paper and examples therein, for simplicity, we may refer to the existing mining pool to indicate a hypothetical pool with the same mining pool as the one mentioned.

$46,108.37 ≃ $11,338,913, with a net profit of $11,338,913 – $3,806,150 = $7,532,763.

Figure 17 presents the cost for renting the hashing power, and the net rewards for each pool, according to our simulation results (see Subsection 5.1 for simulation details). During the simulation, each pool rents hash power to reach the threshold of 0.27 and then launches a hybrid attack with $\rho = 1.0$. The pools may profit by renting mining power. More specifically, the larger the mining pool, the lower the cost to rent the mining powers and the greater the net profit. For example, for AntPool, the cost to rent mining powers is $3.806 million and the net profits are $7.533 million. Even for the smallest pool SlushPool, renting mining power is profitable, where the net profit is $2.196 million with an investment of $9.135 million. Therefore, unlike other selfish mining, the hybrid attack is implementable, because of its low (controllable) forking rate, and profitability for pools in reality.

6.2 Countermeasures against the hybrid attack

As demonstrated, the hybrid attack is feasible in both theory and practice. The threshold and forking rate of the hybrid attack are lower than those of some analogous attacks (e.g., SM1 and semi-honest mining). Under these conditions, the hybrid attack is easier to implement and harder to detect, making it potentially more harmful than SM1 and semi-honest mining. Given the broad applications of blockchain in various fields, e.g., the Internet of Things [26–29], the countermeasures must be investigated to defend the blockchain systems. While many studies have been conducted on countermeasures against selfish mining attacks [30–32], there has been little research on hybrid attacks. Therefore, we propose some initial countermeasures to mitigate the potential losses caused by hybrid attacks.

Detection mechanism. A hybrid attacker publishes blocks on the private chain and invalidates those of honest miners, thereby increasing their relative rewards. A notable feature of the blocks on the private chain is the significant time difference between mining and publication (aka. holding time). Typically, the holding time for a block on the public chain is very short and can be considered negligible when network delay is not factored in. However, the holding time of blocks from the private chain is relatively long. This is because blocks on the private chain usually have to be hidden for some time before they can be published. Therefore, an attack detection mechanism can be proposed based on the length of the holding time. The idea is to set a threshold for the holding time. If blocks with a holding time larger than the threshold, an early warning will be raised to alert honest miners.

Consensus mechanism. Another factor contributing to the success of the hybrid attack is the PoW consensus mechanism. During the attack, attackers exploit vulnerabilities in the consensus mechanism to establish a private chain. Therefore, to stifle the hybrid attack, other consensus mechanisms may be considered, e.g., Proof of Stake [33–35].

Renting cost. Although a hybrid attack is theoretically feasible, most pools do not reach the profitability threshold. Fortunately, they can rent mining power to reach this threshold and launch the hybrid attack (Fig. 17). The net profits are attractive to attackers. For example, Antpool, which has 17.80% of the system's total power, can generate net profits of around 7.533 million dollars. Even small pools can benefit from renting power. SlushPool, for instance, with only 4.92% of the system's power, can also earn net profits of around 2.196 million dollars. Launching such attacks reduces net profits. One option would be to increase the cost of renting so that attackers do not make a profit, removing any incentive to launch a hybrid attack. Similarly, monitoring the rental of large amounts of mining power can be used for monitoring and early detection purposes.

7 Related work

In this section, we briefly survey the prior works on blockchain attacks that are related to our work.

Selfish mining. Selfish mining is a mining strategy that departs from the intended protocol and consists of withholding mined blocks, which constitute a private chain [6,7,36–38]. Building a private chain instead of publishing mined blocks at once, gives a competitive advantage: selfish miners can start mining the next block earlier than anyone else on the private chain. Then they release blocks from the private chain when convenient, to stay in control of the longest so-far mined chain with a favorable probability. Other blocks mined by honest miners, no longer on the longest chain, are hence invalid, and attackers obtain bigger rewards than what would be due proportionally to their mining power.

Although it is proved that selfish mining can efficiently improve the relative reward, such an attack results in a higher forking rate than expected, which increases the possibility of being detected by honest miners. Semi-selfish mining is proposed to reduce the forking rate by infiltrating some power to the honest parties [39]. The attacker mines on the blockchain partially in the name of the honest miners such that the mining power of the honest miners increases accordingly. Simulation results for such models show that semi-selfish mining can reduce the forking rate to some extent. However, the rewards decrease due to the infiltration power used to mine together with honest miners on the public chain.

Recently, reinforcement learning (RL) has become a popular tool to analyze the mining strategies [40,41]. SquirRL is a deep RL-based framework to analyze attacks, e.g., optimal selfish mining attacks and block withholding attacks, on blockchain incentive mechanisms [42,43]. WeRLman is a novel deep RL-based method to analysis selfish mining attacks with more complex settings, where block rewards vary in the form of transaction fees. The results indicate that reward variability can significantly damage blockchain security. Roozbeh et al. proposed a RL-based model for evaluating adversarial mining strategies —such as selfish mining and low-cost mining —before and after difficulty adjustment [44]. As honest mining will eventually be abandoned, the impact of the reward mechanism will be significantly amplified.

Other attacks of interest. Selfish mining is one of the block withholding attacks (BWH) [18,45,46], where attackers withhold the blocks without publishing them in the public chain. Note that BWH

may take any strategies by utilizing the withholding blocks (e.g., discards these blocks). Building a private chain is one of the strategies. Generally, it is a pool formed by selfish miners that launch the attack, and it is reasonable to assume that only one attacking pool exists in the blockchain. Indeed, a miner's dilemma appears when two pools simultaneously launch BWH attacks. That is, there are no profitable pools, if both sides adopt a BWH attack, just like in the prisoner's dilemma. Crystal aims to resist BWH attack by using quorum certificates [47].

Kwon et al. broke the dilemma by proposing fork after withholding (FAW) [17], which includes infiltrating competitors' pools, discarding the blocks that are mined in that pool by the infiltrated power, so reducing the opponent profitability. FAW has limitations in the case of flexible power assignments. To optimize the power adjustment, Gao et al. proposed a novel attack, where attackers dynamically adjust their power (PAW) [22]. Wang et al. proposed a mixed attack, where attackers switch among BWH, FAW, and PAW [48]. Yang et al., following the work of [22], proposed a new attack model, where attackers consider bribery in selfish mining and optimize their rewards by utilizing reinforcement learning. Additionally, other common attacks in distributed systems are also implemented in Blockchain systems, e.g., Sybil attacks [49] or algebraic fault attacks [50]. Recently, a selfish mining-based denial of service (SDoS) attack is proposed. The authors proved that the SDoS attack can be more threatening to the blockchain system, the attacker only needs to own more than 19.6% of the mining power to increase the revenue [51]. Attacks may additionally threaten the security of user data [52], of course, corresponding countermeasures/techniques have also been proposed to address privacy protection issues [53].

8 Conclusion

Although traditional selfish mining attacks are theoretically profitable, they are rarely attempted in practice due to the high forking rate they induce, which makes them detectable. To reduce the forking rate, we proposed a one-time release attack, which has an acceptable forking rate. However, the reward for this attack is not high enough to make it worthwhile. Therefore, we proposed the hybrid attack by combining SM1 and one-time releasing attack to improve the revenue opportunities.

We defined a stochastic model for both the one-time release attack and the hybrid attack. We then simulated the behaviour of the attacks, finding that the hybrid attack achieves a better trade-off between a lower forking rate and a higher relative reward. Additionally, we discussed the practical feasibility of the hybrid attack under the assumption of rented mining power. Our results show that the hybrid attack can be implemented in practice, even by relatively small pools. Finally, we presented some countermeasures against the hybrid attack. Future work will extend the attacks to other blockchains and consider mutual attacks among multiple attackers.

Acknowledgements

This study was supported by the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005), the Open Fund of National Key Labortory of Parallel and Distributed Computing (PDL) (WDZC20255290113).

Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

Funding note

Open access funding provided by Università degli Studi di Torino within the CRUI-CARE Agreement.

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

[1] Lu Y. The blockchain: state-of-the-art and research challenges. Journal of Industrial Information Integration, 2019, 15: 80−90

[2] Kalla A, De Alwis C, Porambage P, Gür G, Liyanage M. A survey on the use of blockchain for future 6G: technical aspects, use cases, challenges and research directions. Journal of Industrial Information Integration, 2022, 30: 100404

[3] Li X, Liu Q, Wu S, Cao Z, Bai Q. Game theory based compatible incentive mechanism design for non-cryptocurrency blockchain systems. Journal of Industrial Information Integration, 2023, 31: 100426

[4] Eyal I. The miner's dilemma. In: Proceedings of the IEEE Symposium on Security and Privacy. 2015, 89−103

[5] Platt M, Sedlmeir J, Platt D, Xu J, Tasca P, Vadgama N, Ibañez J I. The energy footprint of blockchain consensus mechanisms beyond proof-of-work. In: Proceedings of the 21st IEEE International Conference on Software Quality, Reliability and Security Companion. 2021, 1135−1144

[6] Sapirshtein A, Sompolinsky Y, Zohar A. Optimal selfish mining strategies in bitcoin. In: Proceedings of the 20th International Conference on Financial Cryptography and Data Security. 2016, 515−532

[7] Gutteridge D. Japanese cryptocurrency monacoin hit by selfish mining attack. See Programmerclick.com/article/62902065214/ website, 2018

[8] Eyal I, Sirer E G. Majority is not enough: bitcoin mining is vulnerable. In: Proceedings of the 18th International Conference on Financial Cryptography and Data Security. 2014, 436−454

[9] Chen Z, Tian Y, Peng C. An incentive-compatible rational secret sharing scheme using blockchain and smart contract. Science China

Information Sciences, 2021, 64(10): 202301

[10] Miao Y, Xie R, Li X, Liu Z, Choo K K R, Deng R H. Efficient and secure federated learning against backdoor attacks. IEEE Transactions on Dependable and Secure Computing, 2024, 21(5): 4619−4636

[11] Möser M, Böhme R, Breuker D. Towards risk scoring of bitcoin transactions. In: Proceedings of the International Conference on Financial Cryptography and Data Security. 2014, 16−32

[12] Yu X, Wang H, Zheng X, Wang Y. Effective algorithms for vertical mining probabilistic frequent patterns in uncertain mobile environments. International Journal of Ad Hoc and Ubiquitous Computing, 2016, 23(3-4): 137−151

[13] Liu H, Ruan N, Du R, Jia W. On the strategy and behavior of bitcoin mining with N-attackers. In: Proceedings of 2018 on Asia Conference on Computer and Communications Security. 2018, 357−368

[14] Yu X, Feng W, Wang H, Chu Q, Chen Q. An attention mechanism and multi-granularity-based Bi-LSTM model for Chinese Q&A system. Soft Computing, 2020, 24(8): 5831−5845

[15] Tian Y, Wang Z, Xiong J, Ma J. A blockchain-based secure key management scheme with trustworthiness in DWSNs. IEEE Transactions on Industrial Informatics, 2020, 16(9): 6193−6202

[16] Guo J, Liu Z, Tian S, Huang F, Li J, Li X, Igorevich K K, Ma J. TFF-DT: a trust evaluation scheme for federated learning in digital twin for mobile networks. IEEE Journal on Selected Areas in Communications, 2023, 41(11): 3548−3560

[17] Kwon Y, Kim D, Son Y, Vasserman E, Kim Y. Be selfish and avoid dilemmas: fork after withholding (FAW) attacks on bitcoin. In: Proceedings of 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017, 195−209

[18] Dong X, Wu F, Faree A, Guo D, Shen Y, Ma J. Selfholding: a combined attack model using selfish mining with block withholding attack. Computers & Security, 2019, 87: 101584

[19] Saad M, Anwar A, Ravi S, Mohaisen D. Revisiting nakamoto consensus in asynchronous networks: a comprehensive analysis of bitcoin safety and ChainQuality. In: Proceedings of 2021 ACM SIGSAC Conference on Computer and Communications Security. 2021, 988−1005

[20] Decker C, Wattenhofer R. Information propagation in the bitcoin network. In: Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing. 2013, 1−10

[21] Li T, Wang Z, Yang G, Cui Y, Chen Y, Yu X. Semi-selfish mining based on hidden Markov decision process. International Journal of Intelligent Systems, 2021, 36(7): 3596−3612

[22] Gao S, Li Z, Peng Z, Xiao B. Power adjusting and bribery racing: novel mining attacks in the bitcoin system. In: Proceedings of 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019, 833−850

[23] Dai W, Dai C, Choo K, Cui C, Zou D, Jin H. SDTE: a secure blockchain-based data trading ecosystem. IEEE Transactions on Information Forensics and Security, 2020, 15: 725−737

[24] Chicarino V, Albuquerque C, Jesus E, Rocha A. On the detection of selfish mining and stalker attacks in blockchain networks. Annals of Telecommunications, 2020, 75(3): 143−152

[25] Bonneau J. Why buy when you can rent?. In: Proceedings of International Workshops on Financial Cryptography and Data Security. 2016, 19−26

[26] Ferrag M, Shu L. The performance evaluation of blockchain-based security and privacy systems for the internet of things: a tutorial. IEEE Internet of Things Journal, 2021, 8(24): 17236−17260

[27] Liu Y, Su Z, Wang Y. Energy-efficient and physical-layer secure computation offloading in blockchain-empowered internet of things. IEEE Internet of Things Journal, 2023, 10(8): 6598−6610

[28] Maiti M, Ghosh U. Next-generation internet of things in Fintech ecosystem. IEEE Internet of Things Journal, 2023, 10(3): 2104−2111

[29] Gai K, Wu Y, Zhu L, Zhang Z, Qiu M. Differential privacy-based blockchain for industrial internet-of-things. IEEE Transactions on Industrial Informatics, 2020, 16(6): 4156−4165

[30] Zhang R, Preneel B. Publish or perish: a backward-compatible defense against selfish mining in bitcoin. In: Proceedings of the Cryptographers' Track at the RSA Conference 2017 on Topics in Cryptology. 2017, 277−292

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

[32] Sun W, Xu Z, Chen L. Fairness matters: a tit-for-tat strategy against selfish mining. Proceedings of the VLDB Endowment, 2022, 15(13): 4048−4061

[33] Reijsbergen D, Szalachowski P, Ke J, Li Z, Zhou J. LaKSA: a probabilistic proof-of-stake protocol. In: Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2021. 2021, 1−18

[34] Fanti G, Kogan L, Oh S, Ruan K, Viswanath P, Wang G. Compounding of wealth in proof-of-stake cryptocurrencies. In: Proceedings of the 23rd International Conference on Financial Cryptography and Data Security. 2019, 42−61

[35] Zhao W, Yang S, Luo X, Zhou J. On PeerCoin proof of stake for blockchain consensus. In: Proceedings of the 3rd International Conference on Blockchain Technology. 2021, 129−134

[36] Davidson M, Diamond T. On the profitability of selfish mining against multiple difficulty adjustment algorithms. Cryptology ePrint Archive, 2020

[37] Yang G, Wang Y, Wang Z, Tian Y, Yu X, Li S. IPBSM: an optimal bribery selfish mining in the presence of intelligent and pure attackers. International Journal of Intelligent Systems, 2020, 35(11): 1735−1748

[38] Niu J, Feng C. Selfish mining in Ethereum. In: Proceedings of the 39th IEEE International Conference on Distributed Computing Systems (ICDCS). 2019, 1306−1316

[39] Li T, Wang Z, Chen Y, Li C, Jia Y, Yang Y. Is semi-selfish mining available without being detected?. International Journal of Intelligent Systems, 2022, 37(12): 10576−10597

[40] Bar-Zur R, Abu-Hanna A, Eyal I, Tamar A. WeRLman: to tackle whale (transactions), go deep (RL). In: Proceedings of the 15th ACM International Conference on Systems and Storage. 2022, 148

[41] Wang Y, Li C, Zhang Y, Li T, Ning J, Gai K, Choo K K R. A detection method against selfish mining-like attacks based on ensemble deep learning in IoT. IEEE Internet of Things Journal, 2024, 11(11): 19564−19574

[42] Hou C, Zhou M, Ji Y, Daian P, Tramèr F, Fanti G, Juels A. SquirRL: automating attack analysis on blockchain incentive mechanisms

with deep reinforcement learning. In: Proceedings of the 28th Network and Distributed System Security Symposium. 2021, 1−18

[43] Niu J, Wang Z, Gai F, Feng C. Incentive analysis of bitcoin-NG, revisited. ACM SIGMETRICS Performance Evaluation Review, 2021, 48(3): 59−60

[44] Sarenche R, Aghabagherloo A, Nikova S, Preneel B. Bitcoin under volatile block rewards: how mempool statistics can influence bitcoin mining. In: Proceedings of 2025 ACM SIGSAC Conference on Computer and Communications Security. 2025, 903−917

[45] Nayak K, Kumar S, Miller A, Shi E. Stubborn mining: generalizing selfish mining and combining with an eclipse attack. In: Proceedings of 2016 IEEE European Symposium on Security and Privacy (EuroS&P). 2016, 305−320

[46] Bag S, Ruj S, Sakurai K. Bitcoin block withholding attack: analysis and mitigation. IEEE Transactions on Information Forensics and Security, 2017, 12(8): 1967−1978

[47] Niu J, Gai F, Han R, Zhang R, Zhang Y, Feng C. Crystal: enhancing blockchain mining transparency with quorum certificate. IEEE Transactions on Dependable and Secure Computing, 2023, 20(5): 4154−4168

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

[49] Douceur J R. The Sybil attack. In: Proceedings of the 1st International Workshop on Peer-to-Peer Systems. 2002, 251−260

[50] Zhang F, Guo S, Zhao X, Wang T, Yang J, Standaert F, Gu D. A framework for the analysis and evaluation of algebraic fault attacks on lightweight block ciphers. IEEE Transactions on Information Forensics and Security, 2016, 11(5): 1039−1054

[51] Wang Q, Xia T, Wang D, Ren Y, Miao G, Choo K K R. SDoS: selfish mining-based denial-of-service attack. IEEE Transactions on Information Forensics and Security, 2022, 17: 3335−3349

[52] Gai K, Wu Y, Zhu L, Qiu M, Shen M. Privacy-preserving energy trading using consortium blockchain in smart grid. IEEE Transactions on Industrial Informatics, 2019, 15(6): 3548−3558

[53] Gai K, Wu Y, Zhu L, Xu L, Zhang Y. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. IEEE Internet of Things Journal, 2019, 6(5): 7992−8004



Yilei WANG received her PhD degree in computer science from Shandong University, China in 2014. She is currently a professor with School of Computing Sciences, Qufu Normal University, China and a guest professor with College of Artificial Intelligence, Jiaxing University, China. She mainly engages in research on network security, blockchain, and smart contracts. In recent years, she has published more than 30 academic papers in international authoritative journals and conferences.



Junxi ZHU currently is a postgraduate of Qufu Normal University, China. His research interests include machine learning and smart contracts security.



Zhaojie WANG currently is a PhD student of Soochow University, China. His research interests include machine learning and smart contracts security.



Andrea BRACCIALI is currently a lecturer in the University of Turin, Italy. He has been studying blockchains and smart contracts since 2014, chairing workshops and conferences and serving in conference and editorial committees. He got his PhD in computer science from The University of Pisa, Italy, and has previously held positions at the University of Pisa, Italy and ISTI-CNR.



Minghao ZHAO received his PhD degree from Tsinghua University, China. He is currently an assistant professor at the East China Normal University, China. He has co-authored over 60 papers in recognized journals and conferences, such as FAST, ICDCS, ICPP, IWQoS, TPDS, and TDSC. His research includes cloud computing/storage, operating systems, and software engineering. He is a member of IEEE, ACM, and CCF (China Computer Federation).



Wenjian LUO currently is a professor and doctoral supervisor at the school of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. He is a senior member of IEEE/ACM/CCF. His research focuses on secure intelligent systems, encompassing fundamental theories and applications in machine learning, immune computing, and swarm intelligence, with applications in information security, privacy protection, and data mining.



Huiyu ZHOU received his BEng degree in radio technology from Huazhong University of Science and Technology, China, a MSc degree in biomedical engineering from University of Dundee, UK, and a PhD degree in computer vision from Heriot-Watt University, UK. He is currently a full professor with the School of Computing and Mathematical Sciences, University of Leicester, UK. He has authored and co-authored over 400 peer reviewed papers in the field. His research work has been or is being supported by UK EPSRC, MRC, AHRC, ESRC, EU ICT, Royal Society, Innovate UK, Leverhulme Trust, Invest NI, Puffin Trust, Alzheimer Research (UK), and industry.