Partial Selfish Mining for More Profits

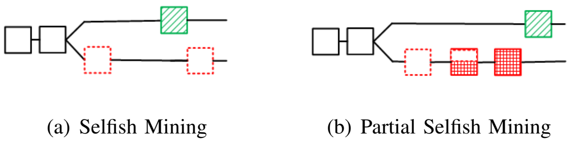
Jiaping Yu*†, Shang Gao†, Rui Song†, Zhiping Cai* and Bin Xiao†
*College of Computer
National University of Defense Technology
email:{yujiaping19, zpcai}@nudt.edu.cn
†Department of Computing
The Hong Kong Polytechnic University
email:{shanggao, csrsong, csbxiao}@comp.polyu.edu.hk

***Abstract*—Mining attacks aim to gain an unfair share of extra rewards in the blockchain mining. Selfish mining can preserve discovered blocks and strategically release them, wasting honest miners' computing resources and getting higher profits. Previous mining attacks either conceal the mined whole blocks (hiding or discarding), or release them completely in a particular time slot (e.g., causing a fork). In this paper, we extend the mining attack's strategy space to partial block sharing, and propose a new and feasible Partial Selfish Mining (PSM) attack. We show that by releasing partial block data publicly and attracting rational miners to work on attacker's private branch, attackers and these attracted miners can gain an unfair share of mining rewards. We then propose Advanced PSM (A-PSM) attack that can further improve attackers' profits to be no less than the selfish mining. Both theoretical and experimental results show that PSM attackers can be more profitable than selfish miners under a certain range of mining power and network conditions. A-PSM attackers can gain even higher profits than both selfish mining and honest mining with attracted rational miners.**

I. Introduction

Decentralized digital currencies like Bitcoin have captured public interest for years [34]. By 2022, Bitcoin has a market value of more than 851.8 billion USD. In Bitcoin, the first miner who solves the puzzle and broadcasts the result will be rewarded with 6.25 bitcoins (BTC), which is worth of 194,319.5 USD by June, 2022 [8]. The more computing resources the miner applies, the more likely it can solve the puzzle and get the bonus first[32], [22]. The result of this puzzle is known as ”Proof-of-Work”(PoW) in Bitcoin.

However, the high rewards of Bitcoin mining have also made it a valuable target for attackers [36], [17], [1], [13]. Previous work has shown that malicious miners can launch attacks by not following the standard mining process, e.g., hiding or discarding mined blocks, releasing a block to cause a fork [29], [33], [31], [30]. This kind of attack is generally referred to as *mining attacks*.



(a) Selfish Mining

(b) Partial Selfish Mining

Fig. 1. (a) Selfish mining withholds discovered blocks in its private chain; (b) Partial selfish mining can firstly withhold a discovered block, then share partial block data, and finally broadcast the whole block.

Selfish mining [10] is one of the most fundamental and well-known mining attacks. In selfish mining, instead of following mining rules and releasing a discovered block immediately, an attacker can withhold the block and continue mining on it as a private branch, as shown in Figure 1(a). When other miners find a new block, the attacker can cause a fork by releasing the withheld block immediately. If attacker's fork is selected as the main chain, honest miners' mining power is wasted.

Following the idea of concealing blocks in the selfish mining, extensive lines of work have explored variant mining attack tactics against the PoW blockchain. Attackers can either apply these tactics individually or strategically combine them to launch sophisticated mining attacks [15]. For instance, using the power splitting, block withholding and selfish mining, Kwon et al. propose Fork After Withholding (FAW) attacks [16]. Besides mining attacks, researchers also proposed other incentive-based DoS attacks in recent years. For example, Michael et al. believe most miners are *rational* in the sense that miners incline to choose the most profitable mining strategy to work (i.e., mining on which chain) [21].

In this paper, we propose a new block sharing strategy in the mining, called **partial block sharing**. Different from previous new block hiding or revealing, partial block sharing will only reveal part of a block (named *partial block*) while some fields are hiding, e.g., *nonce* and part of arbitrary bytes in the coinbase transaction. Here, we denote the hidden data as *secret*. Partial block suffices to mine a consecutive second block, with all transaction outputs (related to UTXO) and block header hash value available. A partial block is regarded as an invalid block by honest miners but does not hinder rational miners to mine after it if it is more profitable.

Based on the partial block sharing strategy, we propose a new mining attack called **Partial Selfish Mining** (**PSM**). As shown in Figure 1(b), PSM starts as selfish mining to withhold a newly mined block. Then, the attacker can launch the partial block sharing strategy, and finally releases the secret by broadcasting the whole block before or right after another miner finds a new block. Previous works show that some (not

Network and Distributed System Security (NDSS) Symposium 2023
28 February - 4 March 2023, San Diego, CA, USA
ISBN 1-891562-83-5
https://dx.doi.org/10.14722/ndss.2023.23xxx
www.ndss-symposium.org

all) miners may choose the most profitable way to mine. In this paper, we define these miners as *rational miners*. A rational miner can get the partial block before honest miner's block and may be attracted to work in the attacker's private branch, which is named as *greedy* miner in this paper. More greedy miners joining the private branch, the higher probability of being the main chain. Thus, compared with selfish mining, PSM can not only waste the mining power of honest miners, but also greatly enhance the success rate of attacker's private branch to be chosen as the longest chain. Both the attacker and greedy miners can gain an unfair share of mining rewards.

To make PSM practical, we must have mechanisms to convince rational miners that it is profitable to mine in attacker's private branch. First, the attacker needs to ensure that it indeed has the secret, i.e., a complete and valid new block. Otherwise, all the mining power that rational miners spend on the private branch is in vain. Second, the attacker may deny broadcasting its secret as promised. We name this sabotaging behavior as PSM Denial of Service (PSM-DoS) attack. To address the first concern about block validation, we propose a zero-knowledge-proof-based mechanism to prove the block possession, which is entirely owned by the attacker. To counteract PSM-DoS attacks, we design an economic based profit protection mechanism for rational miners. If the attacker fails to broadcast the promised secret/block timely, his deposit in a trusted third party could be forfeited by a rational miner. This mechanism also ensures that the secret can be calculated by others with limited computing power in an acceptable period of time, and makes the attacker's PSM-DoS attack economically not worthwhile. In this paper, we implement the profit protection mechanism with a smart contract. Please note that this mechanism does not need to be deployed on the target chain. It can be constructed on any third-party platform that is trusted by both rational miners and the attacker.

Since estimating the fraction of rational miners in the wild is hard, we discuss the attacker's revenue with different fractions of the rational miners. We model PSM and analysis result shows that PSM can have a higher reward than both the honest and selfish mining in a certain range. It is always beneficial for a rational miner to join the attacker's private branch if his mining power is smaller than the attacker's. We also show the extra revenue obtained by greedy miners when following the partial block.

Moreover, we propose an **Advanced PSM** (**A-PSM**) strategy for attackers to gain more profits. An attacker has economic incentives to monitor the block height of the public branch to apply A-PSM actively. This block height can also be reported by greedy miners or obtained via decentralized oracle [35] in case of any dispute to guarantee greedy miners' rewards.

We show that the attacker's profit is no less than selfish mining in A-PSM. In most cases, greedy miners will follow attacker's branch to have more mining rewards. Thus, we believe that A-PSM is a lucrative alternative to the selfish mining in mining-related attacks.

We analyze the scenario when multiple attackers adopt either PSM or A-PSM. Greedy miners should join the branch that publishes the partial block early to get more mining time and expected rewards. If attackers release partial blocks simultaneously in distinct branches, joining any branch has no impact on greedy miner's profits in PSM. However, they should join the branch with a larger attacker mining power in A-PSM.

Our theoretical analysis and evaluations show consistent results that the PSM attack can increase the attacker's profits for a large fraction of the parameter space, and A-PSM attack can always earn more profits than selfish mining. In PSM, when 50% of miners are rational, an attacker with 20% mining power can get 1.25% and 9.79% more profits than the honest mining and selfish mining, respectively. In A-PSM, when 30% of miners are rational, an attacker with 10% mining power can have a higher reward, gaining at most 23.6% and 13.1% more profits than the honest mining and selfish mining, respectively.

In summary, we have made the following contributions:

- We develop a new block sharing strategy called partial block sharing. Based on it, we propose a new mining attack protocol, PSM. By sharing the partial block data and attracting greedy miners to work on attacker's private branch, the attacker and greedy miners can gain an unfair share of the mining reward.
- To make PSM feasible and prevent potential PSM-DoS attacks, we propose two mechanisms to convince rational miners to join the attacker's private branch. The first zero-knowledge-proof-based mechanism can prove the new whole block possession owned by the attacker. The second secret computation mechanism can make PSM-DoS attacks not worthwhile for attackers.
- To further increase attacker's profits, we proposed A-PSM. A-PSM assures an attacker of revenue no less than selfish mining. It can even outperform honest mining when the overall mining power is no more than 50% in the attacker's private branch (including attracted greedy miners).
- We analyze and evaluate the profit of an attacker, multiple attackers and greedy miners who follow PSM and A-PSM respectively. In PSM, greedy miners can have more profits when their mining power is less than the attacker. An attacker adopting PSM can be more profitable than adopting selfish mining under a certain range of mining power and network conditions. With a realistic fraction of rational miners, A-PSM can always earn a higher profit than both selfish mining and honest mining.

In Section II, we present an overview of the blockchain background and related work. Section III describes the mining model and assumptions. Then we describe the PSM attack in Section IV in detail, including the profit analysis of rational miners. In Section V, we propose mechanisms to make PSM feasible. In Section VI, we use numeric analysis to evaluate the profitability of PSM strategy by comparing with both the honest and selfish mining. Then we propose the A-PSM strategy in Section VII. In Section VIII, we further illustrate in what conditions PSM and A-PSM can be the dominant mining strategy respectively. In Section IX, we discuss the double-attacker scenario and countermeasures against PSM and A-PSM strategies. Section X concludes the paper.

II. Preliminaries

A. Bitcoin Background

Mining Process: The bitcoin is created as a reward for the mining process. The mining process is performed by the blockchain network peers, known as "miners". By conducting the Proof-of-work (PoW) process, miners race against each other to get the chance to generate the new block.

Specifically, to mine a new block, miners need to know at least all transaction outputs related to UTXO and the block header hash value of the previous blocks. Denote the block header as bh. To find a valid block, miners need to find a valid nonce that satisfies $sha256(sha256(bh)) < L$. Here, L is the target value determined by the blockchain protocol, which indicates the difficulty of finding a new block. In Bitcoin, miners record the difficulty D in the block header. It is derived from the target L and adjusted after every generation of 2016 blocks. This process ensures that each block is generated within approximately 10 minutes. The difficulty value D is defined as: $D = \frac{L_1}{L}$, where $L_1 =$0x1d00ffff, is the target value when $D = 1$. By May 2022, the $D \approx 31.25 \times 10^{12}$, which means in the Bitcoin network, miners can perform more than 1.83×10^{20} hashes per second.

Fork and Race: A race occurs when multiple miners broadcast a new block simultaneously. Other miners will consider the longest chain from its perspective as the main chain and work on it. If there are multiple longest chains, miners will consider the longest chain it receives first as the main chain. Only the blocks on the mostly regarded main chain can benefit the miners. Since all the mining power spent on the other branch is wasted, attackers can intentionally cause forks and keep the fork state as long as possible to waste the mining power and launch double spending attacks [27].

B. Related Work

Selfish Mining: Selfish mining was first proposed by Eyal et al. [10]. A selfish mining attacker can earn extra rewards by intentionally generating a fork. When an attacker discovers a new block in selfish mining, it will keep the block as its private branch and keeps mining after it. When other miners find a block, the attacker will release the withheld blocks to cause a fork. Once its private branch is selected as the main chain, the attacker can earn extra rewards. According to [10], miners with more than 33% computational power can surely get an extra reward compared with honest mining. In [23], researchers find that for a larger parameter space, by following a more 'stubborn' strategy, miners can gain more rewards than selfish mining when their mining power is large enough. Liao et al. [19] present the whale transactions. By deploying large fees, attackers can incentivize miners to fork the blockchain. In recent years, Negy et al. [24] further analyzed the profits of selfish mining and proposed intermittent selfish mining. It assures the attacker can get more profits than honest mining even after the difficult adjustment. Li et al. [18] analyze the mining attack strategy from the honest miner's perspective and optimize the selfish mining with a hidden Markov decision process.

Selfish mining is not frequently observed in the wild because on the one hand, it is not easy to detect selfish mining. On the other hand, for famous cryptocurrencies like Bitcoin and Ethereum, it is not easy to occupy more than 33% mining power. But the threat of selfish mining still remains a concern. For smaller cryptocurrencies, there are some conducted selfish mining cases reported [6]. Ethereum communities have already taken measures to avoid the potential selfish mining attacks [25].

Other Cryptocurrency-Related Attacks: After the proposal of selfish mining, researchers have proposed various mining attack tactics against the PoW Blockchain. For those who do not have enough mining power, Loi et al. [20] proposed the block withholding (BWH) attack. By strategically splitting the mining power, attackers can get more rewards than honest mining in the long run. Kwon et al. [16] combine the BWH with selfish mining and propose the Fork After Withholding (FAW) attack. Instead of simply discarding the full block finding in the victim pools, in FAW, attackers mine on the newly mined block to generate a private branch. In recent years, Gao et al. [11] proposed Power Adjusting Withholding (PAW) attack. PAW attackers adjust the mining power between the victim pools and solo mining, unlike FAW attackers. By doing so, PAW attackers can gain twice as much revenue as FAW attacks.

Besides mining attacks, researchers also proposed other incentive-based attacks. For example, based on the assumption that miners incline to choose the most profitable mining strategy to work, Michael et al. [21] propose the Blockchain Denial of Service (BDoS) attack. Instead of getting a higher reward, BDoS attacker invests resources to reduce other miners' profits and lure them away from mining. By publishing the block header of the newly mined block, the attacker signal to the miners that the system is in a state that reduces its profits. In our study, PSM attackers share the full block except for the secret data instead of sharing the block with the rational miner. And miners who receive the partial block data are not likely to suffer a loss. Instead, it can reevaluate its profits and choose the most profitable way to work.

When race occurs, miners choose the block they received first as the legal block. In most of the works [4], [28], [21], researchers tend to believe that each branch has about 50% of the miners to work on. However, Saad et al. researched the Bitcoin safety properties and concluded from different angles. In [26], they pointed out that the assumption about the strong network synchrony does not hold in the real-world deployment. They also realized a HashSplit attack that allows the attacker to orchestrate the mining power distribution when race occurs. Then Saad et al. [27] further researched the bitcoin network thoroughly and found out the unstable hash rate distribution can make it possible for attackers to launch a double-spending attack without mining power.

III. Model and Assumptions

A. Mining Model

We consider a blockchain system with n miners whose normalized mining powers are denoted by $\alpha_1, \alpha_2 ... \alpha_n$, and an attacker with mining power α_A, such that $\alpha_A + \sum_{i=1}^{n} \alpha_i = 1$.

As shown in figure 2, in PSM attack, without loss of generality, we assume that miners are divided into the following groups:

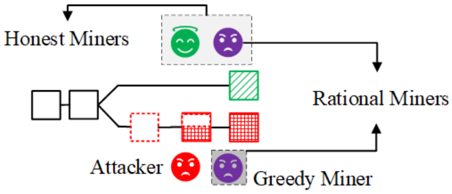


Fig. 2. **Miner's role in PSM scenario.** *Attacker*: a colluding minority pool that conducts the PSM attack. *Rational Miners*: Miners that chooses the mining strategy that benefits them most. *Greedy Miners*: miners working in attacker's private branch. *Honest Miners*: miners working in the original public branch.

Attacker:
: A miner or a colluding minority pool that follows the PSM strategy. It can preserve blocks it mined, form a private branch, and share partial block data with rational miners like [21]. Besides, it also has access to the smart contract to share the proof of block possession and other needed data with rational miners.

Rational miners:
: A minority group of miners that will not take the initiative to launch an attack but may choose the most profitable way to mine. Realizing the attacker launched a partial block sharing related attack like PSM, rational miners can choose their optimal strategies (i.e., mining on which branch) to get a higher reward.

Greedy miners:
: Part of rational miners that choose to work on the attacker's private branch.

Honest miners:
: Miners working in the public branch when no race happens, including part of rational miners and all other non-rational miners. When rational miners choose to behave honestly, from a third-party viewer's point of view, it has no behavior that contradicts the blockchain protocol.

Note that all the greedy miners are rational, but the converse does not always hold since some rational miners may regard mining on the original branch as their optimized strategy in some cases.

We denote the rushing ability of the attacker by γ. If the attacker publishes a new block from its private branch racing with other miners, γ is the expected ratio of honest miners that adopt the attacker's block. γ mainly depends on the miner's network condition. In previous works, γ is commonly considered as $\frac{1}{2}$ [10], [28]. However, some researchers also pose methods that could enable the attacker to get larger γ[26]. In this paper, we consider $0 < \gamma < 1$ in our analysis.

B. Assumptions

We have the following assumptions that are consistent with other PoW-based mining attacks, such as [10], [23], [12], and [11].

1) Instead of 6.25 BTC, the reward of a new block is normalized to 1. Our analysis gives the expected reward for every participant [2].
2) Each miner/pool's computational power should not be greater than 0.5 to avoid "51% attacks" [3].
3) Rational miners do not trust the attacker and join its private branch unless the attacker can provide an extra method to guarantee the profits of rational miners.
4) Not all the miners in the blockchain system are rational. Though we proposed some measures to address the concern of miners to work in the attacker's private branch, we take the possibility that they cannot allay the concerns of all miners into consideration. Thus we discuss the fraction of rational miners is no more than 50% in our research.
5) Rational miners will not intentionally launch any attack. The assumption of profit-driven miners is also made in [9], [10], [2].

IV. Partial Selfish Mining Attack

In PSM, an attacker follows the partial block sharing strategy and shares the partial block information with rational miners. The partial block has some data covered, e.g., *nonce* and part of arbitrary bytes in the coinbase transaction. Miners can mine after it to get a new block. The hidden data can be recovered by others, spending considerable mining power. We denote the hidden data as *secret*.

The generic PSM can firstly withhold the new block, then release the partial block, and finally broadcast the complete block. If an attacker does not release the partial block nor complete block in PSM, it becomes the selfish mining. The attacker can also bypass the first two steps and directly publish the complete block and it is the honest mining. Or the attacker can bypass the first step (when the selfish mining is not the dominant strategy as discussed in Section VIII) but continue the second and third steps, which has not been discussed previously and is the focus of this section.

A. Attack Overview

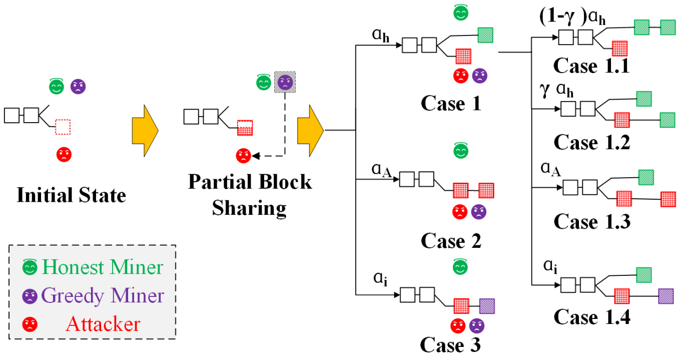


Fig. 3. **Workflow of PSM strategy.** Instead of publishing a new block, the attacker shares partial block data with other miners to attract them to join its private branch. Three possible cases of finding another new block after the announcement of the partial block. Case1: By honest miners; Case 2: By the attacker; Case 3: By greedy miners.

The workflow of PSM is shown in Figure 3. In PSM, when the attacker finds a block, it keeps the block private instead of immediately releasing it. In the meantime, the attacker releases the partial block together with the proof of block possession . With these released data, rational miners could mine on the private branch. To assure the profits of greedy miners, the attacker will also announce a smart contract that allows the rational miner to get the attacker's collateral if it does not release the secret as it promised.

After the releasement of the partial block, three possible cases may happen: the attacker finds the new block on its private branch, honest miners find a new block on the public branch, or greedy miners find a new block on the private branch.

In the first case where the attacker finds a new block in the private branch, the attacker will release the whole private branch and get the two blocks' revenue. Then the system goes back to the single branch state.

In the second case where miners on the public branch find the new block, the attacker releases its private branch and starts a 0-lead racing. In this scenario, the attacker and the greedy miners will mine on the previously private branch, and honest miners will choose to mine on either branch. As defined earlier, γ honest miners will work on the private branch, and $1-\gamma$ honest miners will mine on the public branch.

In the third case, the greedy miners who participate in the private branch find the new block. Like the first case, the attacker will immediately release the whole private branch to get the revenue of all the blocks on the private branch. Then the blockchain goes back to the single branch state.

B. PSM Reward

We use the following parameters in our analysis.

- γ: the ratio of honest miners choosing attacker's branch when race occurs.
- α_i: mining power of greedy miners.
- α_R: mining power of all miners but the attacker.
- α_A: mining power of the attacker.
- α_h: mining power of honest miners.
- R^n_m: revenue of miner m in case n.

Figure 3 shows the possible cases after the attacker release the partial block data. When the attacker first finds a new block, it will publish the message that it has found the new block, together with the proof and the partial block data. In this state, four possible cases may happen.

Case 1: An honest miner finds the new block. Then four possible sub-cases may happen:

Case 1.1: An honest miner finds the new block after the honest miner's branch. In this case, the honest miner will get 2 block rewards. The possibility is $(1-\gamma)\alpha_h$. The expected profits of each participant can be represented as: $R^A_{1.1} = R^i_{1.1} = 0, R^h_{1.1} = 2\alpha_h \times (1-\gamma)\alpha_h$.

Case 1.2: An honest miner finds the new block after the attacker's branch. In this case, the attacker will get 1 block reward and the honest miner will get 1 block reward. The possibility is $\gamma\alpha_h$. The expected profits of each participant can be represented as: $R^A_{1.2} = \alpha_h \times \gamma\alpha_h, R^i_{1.2} = 0, R^h_{1.2} = 2\alpha_h \times (1-\gamma)\alpha_h$.

Case 1.3: The attacker finds the new block. In this case, the attacker will get 2 block rewards. The possibility is α_A. The expected profits of each participant can be represented as:$R^A_{1.3} = 2\alpha_h \times \alpha_A$, and $R^i_{1.3} = R^h_{1.3} = 0$.

Case 1.4: A greedy miner finds the new block. In this case, the attacker will get 1 block reward, and the greedy miner who finds the new block will get 1 block reward. The possibility is α_i. We can represent the expected profits of each participant as follows: $R^A_{1.4} = R^i_{1.4} = \alpha_h \times \alpha_i, R^h_{1.4} = 0$.

Case 2: The attacker finds the block. The attacker will release the two partial blocks to the public chain immediately. In this case, the adversary will get 2 block rewards. we can represent the expected profits of each participant as follows: $R^A_2 = 2\alpha_A, R^i_2 = R^h_2 = 0$.

Case 3: A greedy miner finds the block. The attacker will release two partial blocks to the public chain immediately. In this case, the adversary will get 1 block reward, and the greedy miner will get 1 block reward. The expected profits of each participant can be represented as: $R^A_3 = R^i_3 = \alpha_i, \quad R^h_3 = 0$.

We can derive the attacker's expected profits as:

$$\begin{aligned} R^P_A =& R^A_{1.2} + R^A_{1.3} + R^A_{1.4} + R^A_2 + R^A_3 \\ =& \gamma\alpha_h^2 + 2\alpha_A\alpha_h + \alpha_i\alpha_h + 2\alpha_A + \alpha_i. \end{aligned} \tag{1}$$

C. Rational Miners' Profits in PSM

To attract rational miners to work in its private branch, the attacker needs not only to assure the rational miners can get more rewards than honest mining but also to make a solid promise to guarantee that the attacker's cost of breaking the promise is unbearably high. In this section, based on our assumption in Section III, we analyze the rational miner's strategy space and their profits.

Assuming in a bitcoin-like blockchain network, for the rational miner k, it will know the α_A and γ from the attacker's message. It is not easy to verify the correctness of γ, but rational miners could calculate its profits with different γ and make their decision from the worst case ($\gamma = 0$). Miner k will choose the most profitable branch to work. We are interested in which strategies can maximize the miner's profit with different mining power given γ and α_A.

Assuming that all the rational miners have made their decision except the miner k. When following the greedy mining strategy, the revenue of miner k is:

$$R^G_k = R^i_{1.4} + R^i_3 = (1+\alpha_h)\alpha_k. \tag{2}$$

When mining honestly, the possible cases that minker k could get rewards is shown in figure 4, and the rational miner k's revenue is:

$$\begin{aligned} R^H_k &= R^k_A + R^k_B + R^k_C + R^k_D \\ &= 2\alpha_k^2 + (1-\gamma)\alpha_h\alpha_k + \alpha_h\alpha_k. \end{aligned} \tag{3}$$

Here, we first consider a specific case: an attacker with the computational power of 0.1 who execute the PSM attack

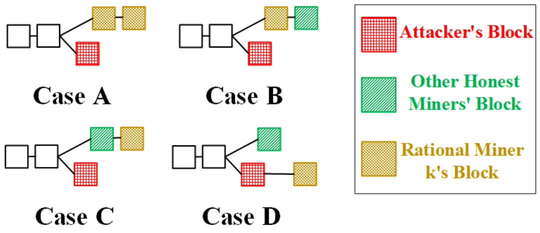


Fig. 4. Cases that rational miners can get revenue when mining honestly. Case A: miner k finds two blocks on the public branch. Case B: miner k finds one block on the public branch, then other honest miners find a block on the public branch. Case C: other honest miners find a new block on the public branch, then miner k finds a new block on the public branch. Case D: another honest miner finds a new block on the public branch, then miner k find a new block on the attacker's branch.

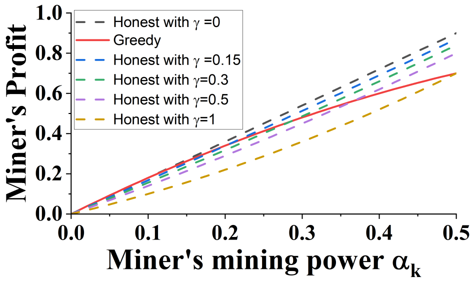


Fig. 5. **Revenue comparisons for a miner to choose greedy or honest mining strategies when attacker's mining power** $\alpha_A = 0.1$**.** Note that the rushing ability γ has no impact on the revenue of greedy strategy.

against the blockchain. For different α_k, the expected profits of being greedy and honest are shown in Figure 5.

To further evaluate the profits of both strategies, we calculate the relative extra reward (RER) between following greedy and honest strategies.

The expected RER can be expressed as:

$$RER_k^{s_1,s_2} = \frac{R_k^{s_1} - R_k^{s_2}}{R_k^{s_2}}, \tag{4}$$

where s_1 and s_2 indicate different strategies, and k represents miner k, and $R_k^{s_1}$ presents the reward of k when adopting s_1 strategy. The RER of the miner k is:

$$RER_k^{G,H} = \frac{1 - 2\alpha_k - (1-\gamma)\alpha_h}{2\alpha_k + (2-\gamma)\alpha_h}. \tag{5}$$

$RER_k^{G,H} > 0$ means being greedy is more profitable for the miners, and a negative value means the miner suffers a loss when choosing the greedy strategy.

With $\alpha_h = 1 - \alpha_A - \alpha_i - \alpha_k$, when $RER_k^{G,H} > 0$, we have:

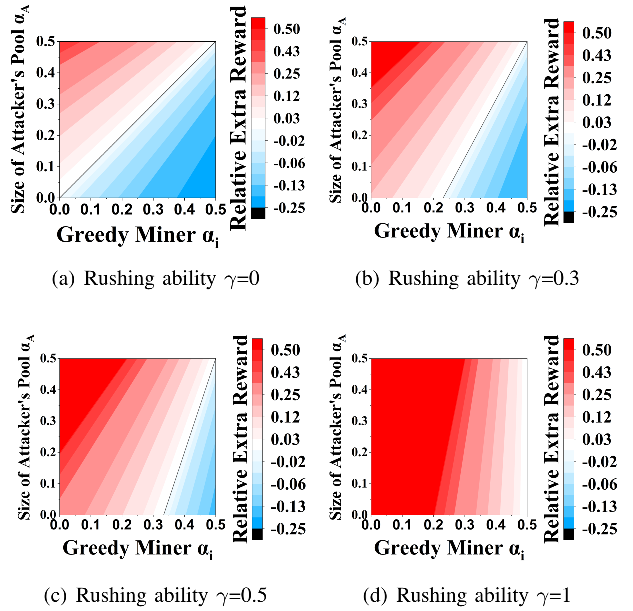
$$\begin{aligned} 1 - 2\alpha_k - (1-\gamma)\alpha_h &> 0 \\ \alpha_k &< \frac{1 - (1-\gamma)(1-\alpha_A-\alpha_i)}{1+\gamma} \end{aligned} \tag{6}$$

In theory, there will be multiple rational miners in a blockchain network. But in practice, miner k cannot tell how many rational miners there are in the network. Therefore, when judging the profit, the rational miner should consider the worst case that there is only one rational miner in the whole network, $\alpha_i = 0$.

γ \ α_A	0.1	0.2	0.3	0.4
0	-5.00(-5.00)	0.02(0)	6.25(6.25)	14.30(14.29)
0.25	7.03(7.04)	12.48(12.50)	19.33(19.3)	27.95(28.0)
0.5	22.56(22.56)	28.47(28.47)	35.96(36.00)	45.34(45.45)
0.75	43.46(43.4)	50.04(50.0)	58.19(58.144)	68.36(68.42)
1	72.80(72.73)	79.98(80)	88.79(88.89)	100.02(100.0)

TABLE I. **Monte Carlo simulation result of rational miner k's relative extra reward** ($RER_k^{G,H}$). Miner k's mining power $\alpha_k = 0.2$. The values $x(y)$ indicates the miner's RERs in simulation and theoretical analysis respectively.

If let the the $\alpha_i = 0$, from Equation (6), we can see that the miner's RER mainly depends on γ and α_A. In the worst case, when $\gamma = 0$ and $\alpha_i = 0$, miner k will get more rewards in the attacker's private branch only if $\alpha_k < \alpha_A$. The quantitative simulation result is shown in Figure 6.



(a) Rushing ability γ=0

(b) Rushing ability γ=0.3

(c) Rushing ability γ=0.5

(d) Rushing ability γ=1

Fig. 6. **Rational miner's relative extra reward between being greedy and honest with an PSM attacker**($RER_k^{G,H}$). The solid line represents no extra reward. For miner k, when $\gamma = 0$, working on the private branch is more profitable only when $\alpha_k < \alpha_A$. Working on the private branch is always more profitable when $\gamma > 0.5$.

To verify the theoretical result, we simulate the RER of a greedy miner with a mining power of 0.2, using a Monte Carlo method over 10^9 rounds, with an upper bound of 10^{-4} error. The Monte Carlo simulation results are in Table I. The result is the same as our expectation.

V. Feasibility of PSM

In this section, we describe the mechanism to convince rational miners that it is profitable to work on attacker's private branch in detail. To convince the rational miners that the attacker will follow the rules it announced, the attacker needs to provide proof of block possession and an profit protection mechanism that assures the rational miners can get the revenue the attacker promised.

A. Partial Block Publication Method and Proof of Block Possession

Specifically, to provide a proof of block possession, the attacker writes some random information r to the coinbase of the new block to provide sufficient randomness and uses r and the $nonce$ of the block as the witness. Other parts of the block header besides $nonce$ and r, denoted as b, can be used as the public statement, including information such as the Merkle root of transactions in the block, and the hash h of the block which satisfies the difficulty requirement. After this, the attacker needs to prove that:

1) it knows $nonce$ and r such that $H(b, nonce, r) = h$; and
2) the $nonce$ and b are well-formed, i.e., $nonce$ is an 32-bit integer.

To this end, the attacker needs to generate a proof of block possession $\pi_b \xleftarrow{\text{R}} \mathsf{Prove}((b, h), (nonce, r))$ to prove the following relation:

$$H(b, nonce, r) = h \ \wedge \ 0 \leq nonce \leq 2^{32} - 1 \tag{7}$$

is satisfied. The attacker then makes the tuple (π_b, b, h) publicly available on a dedicated website. In this way, other miners can calculate:

$$\mathsf{Verify}((b, h), \pi_b) \stackrel{?}{=} 1 \tag{8}$$

to verify whether the proof π_b holds. If the above verification passes, the other miners can be sure that the attacker holds a specific $nonce$ and r, which enables the hash h of the new block to satisfy the difficulty requirement.

If the attacker does not want to share the block information with every miner, it can also share the partial block data b with a specific miner. For this purpose, it can take advantage of the zero-knowledge contingent payment (ZKCP) protocol [5]. We propose the partial block sharing strategy in Appendix **??**.

Before the exchange starts, the attacker can deploy a smart contract as an arbiter. Then, the attacker and the miner achieve a fair exchange of the partial block data through a simple interaction:

1) The attacker generates a random key k and encrypts the b value with k, i.e., $\hat{b} = \mathsf{Enc}(b, k)$. Next, it computes the hash of k $h_k \leftarrow H(k)$ and generates a proof:

   $$\pi_e \xleftarrow{\text{R}} \mathsf{Prove}((\hat{b}, h, h_k), (b, k, nonce, r)) \tag{9}$$

   to prove the following relation:

   $$\begin{aligned} H(b, nonce, r) = h \ \wedge \ 0 \leq \text{nonce} \leq 2^{32} - 1 \\ \wedge \ \hat{b} = \mathsf{Enc}(b, k) \wedge h_k = H(k) \end{aligned} \tag{10}$$

   is satisfied. Finally, it sends the tuple $(\pi_e, \hat{b}, h, h_k)$ to the miner.
2) After receiving the tuple $(\pi_e, \hat{b}, h, h_k)$ from the attacker, the miner can verify whether π_e is valid by computing $b \leftarrow \mathsf{Verify}((\hat{b}, h, h_k), \pi_e)$. If $b = 1$, the miner deposits to the arbiter contract the payment agreed upon by both parties beforehand, as well as the hash h_k.

	Block Possession	Block Exchange
Proving time	2.77 s	10.71 s
Verification time	19 ms	40 ms
Proof size	517 Byte	612 Byte

TABLE II. PERFORMANCE OF ZERO-KNOWLEDGE PROOFS.

3) The attacker checks whether the h_k provided by the miner is valid, and the payment deposited is as previously agreed. If so, the attacker sends the key k to the arbiter contract.
4) The arbiter contract verifies that $h_k = H(k)$ holds. If it is the case, the contract transfers the payment to the attacker, otherwise it returns the payment to the miner. Since the k disclosed to the contract will also be available to the miner at the same time, the miner can decrypt $\hat{b}$ by $b \leftarrow \mathsf{Dec}(\hat{b}, k)$.

Any miner as a buyer cannot obtain any information about b without completing the payment. Meanwhile, any attacker acting as a seller cannot cheat the payment by submitting the wrong b. In this way, the attacker is able to sell the partial block data b to a specific miner and gain revenue.

To evaluate the performance of proof generation and verification, a computer running Ubuntu 22.04 with a 3.50 GHz Intel i9-11900k CPU and 32 GB of RAM was used. We constructed the above zero-knowledge proofs based on libsnark and implemented the proof of block possessionand ZKCP-based block information exchange using about 3100 lines of C++ code.

Table II shows the performance of generating and verifying these zero-knowledge proofs.

As can be seen from the results, it takes only 2-3 seconds to generate a proof of block possession. The process of generating such proofs is all at once. Once generated, these proofs can be made public along with the statement, and anyone can verify the correctness of the declared relations by evaluating the proofs. The time required to verify such succinct proofs is in the millisecond range and is therefore very efficient.

B. PSM-DoS Attack and The Promise of Secret Publication

It should be noted that the above mechanism can only prove to other miners that the attacker has indeed mined a block that satisfies the requirement. But there is no guarantee that the attacker will share the secret in the future. By potentially withholding the reserved block, the attacker could waste the greedy miner's mining power and may cause a DoS attack on the public branch. Since this new attack has to launch with PSM, we call it PSM-DoS attack.

The workflow of PSM-DoS attack is shown in Figure 7. First, the attacker distributes the partial block data to all the miners and attracts greedy miners to join the attacker's private branch. In the meantime, the attacker leaves the private branch and puts all the mining power back into the public branch. Then, once greedy miners on the private branch find the new block, the attacker refuses to release the first block. In this case, the public chain will not accept the new block because its previous block is not published. Thus, miners who try to follow the attacker will suffer losses if the attacker consistently fails to disclose the full block to other miners.

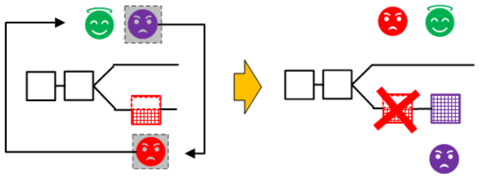


Fig. 7. **Workflow of PSM-DoS Attack.** After attracting the greedy miners to work on its private branch, attacker discard the secret and goes back to work on the public branch.

Here, the PSM attacker needs to apply two countermeasures to address the greedy miners' potential concern about PSM-DoS attacks.

First, the attacker needs to select the number of hidden bytes carefully. When a PSM-DoS attack happens, greedy miners can calculate the hidden bytes within an acceptable time. As we have demonstrated in Section V-A, when the attacker shares the partial block, miners receive the full block without the *nonce* and several bytes of coinbase transaction information. Assuming that the attacker hides n bytes of data, greedy miners need to calculate 2^{4n} times of hash to get the hidden data.

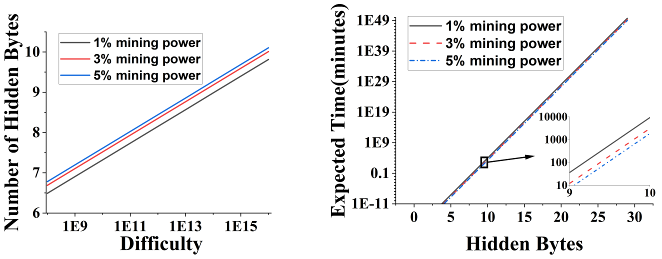
Take Bitcoin as an example. According to [22], with difficulty D, we can approximate the hash power of all the miners in the network as $D \times \frac{2^{32}}{600}$ H/s. Assuming that attacker hides b bytes of data so that fr percent of miners need to calculate for a duration of T_c, then we have

$$T_c = \frac{2^{8b}}{fr \times D \times \frac{2^{32}}{600}}, \quad b = \log_{2^8}(T_c \times fr \times D \times \frac{2^{32}}{600}). \tag{11}$$

Figure 8(a) shows when $T_c = 10$ minutes, the number of bytes the attacker need to hide. According to [7], the Hashrate of the Bitcoin network is $1.55371 \times 10^{16} H/s$. The difficulty of bitcoin $D_{btc} \approx 31.25 \times 10^{12}$, with Equation (11), we can get hiding 9 bytes of data can assure the miner with 1% mining power calculate the hidden bytes for approximately 10 minutes.

With the difficult value of the Bitcoin network, we further evaluate the expected time to get the hidden data with $D = 31.25 \times 10^{12}$. The result is shown in Figure 8(b).

To further address the potential concerns of miners and provide some incentive, the attacker can create a smart contract beforehand and deposit a certain amount of collateral in it, along with the cryptographic commitment of the block data. By doing so, the attacker effectively guarantees that it will redeem the collateral at some agreed point in the future by submitting a secret to the contract that satisfies the requirement and opens the commitment. Otherwise, this collateral will be transferred to the greedy miner who finds the new block. On the other hand, when the collateral value is less than the attacker's cost to deceive other miners, the miner has every incentive to provide an invalid block to deceive other miners. To solve the problem, the collateral value is required to be much larger than the cost of mining for the specified period. Specifically, the flow of this smart contract is as follows.



(a) Number of bytes the attacker need to hide when $T_C = 10min$.

(b) The expected time to get the hidden data with $D = 31.25 \times 10^{12}$.

Fig. 8. **Evaluation result about the number of hidden bytes.**

1) The attacker pre-collateralizes the contract with some coins in the contract with a value equivalent to the proceeds of mining n blocks.
2) A greedy miner discovers a subsequent new block of the attacker's partial block and submits the information about the new block to the smart contract.
3) The smart contract verifies the new block. If the verification passes, the contract opens a challenge period within which the attacker should disclose all information about the partial block. If the attacker discloses the full information of the partial block within the challenge period, it can redeem all the collateral it has previously deposited. Suppose the attacker fails to disclose the block information within the challenge period or discloses an incorrect block. In that case, the contract transfers the attacker's collateral to the greedy miner who finds the new block.

Assuming the challenge period lasts for a duration T_C, the miner will get the revenue, and the system will surely go back to a single branch state after T_C. If the attacker refuses to publish the full block, the attacker will lose both the partial block revenue and the collateral. The attacker can get more profits by launching PSM-DoS only if it could get more than $n + 1$ blocks within T_C.

According to [14], the possibility of finding a new block within duration T can be expressed as:

$$R(T) = \alpha_e \times (1 - (1 - p_e)^{\frac{T}{T_{avg}}}), \tag{12}$$

where the T_{avg} is the average block generation time in the blockchain. In bitcoin $T_{avg} = 10$ minutes. $p_e = 64\%$, which means all miners in the network have a possibility of 64% to generate a new block within 10 minutes.

Theoretically, T_C could be much less than the average block generation time. If the attacker can provide large enough collateral with a low enough T_C, then launching PSM-DoS attack is economically not worthwhile for the attacker.

Though our proposed method could address most of the concerns of rational miners, we agree that it is not realistic to assume all the miners are rational in practice. In the following, we discuss the attacker's gain with different ratios of the rational miners.

VI. PSM Analysis and Comparisons

When calculating the revenue, the attacker can get α_A, γ, and the mining power distribution in the blockchain network. It can also roughly estimate the number of rational miners by the number of applications for partial block data, but it cannot know the fraction of rational miners in the entire network. Thus, the attacker needs to estimate its reward with a different fraction of rational miners.

In this section, we use numeric analysis to evaluate the profitability of the PSM strategy. Then we simulate the gains of the attacker using different mining strategies with different α_A, α_i, and γ with the Monte Carlo simulator.

A. Comparison with Honest Mining

1) Quantitative Analysis: We mathematically analyze the revenue of PSM against honest mining. As stated before, the attacker's computational power is α_A. If the attacker chooses to follow the honest mining strategy, it will surely get 1 block revenue, and the possibility of getting the second block revenue is α_A. Overall, the attacker's expected profit when following honest mining is:

$$R_A^H = 1 + \alpha_A. \tag{13}$$

If the attacker chooses the PSM attack, it could attract rational miners with α_i mining power, and the total mining power of honest miners is $\alpha_h = 1 - \alpha_A - \alpha_i$. For the attacker, the expected RER between PSM and honest mining can be expressed as follows:

$$\begin{aligned} RER^{P,H}_{Attacker} &= \frac{R_A^P - R_A^H}{R_A^H} \\ &= \frac{\gamma\alpha_h^2 + 2\alpha_A\alpha_h + \alpha_i\alpha_h + 2\alpha_A + \alpha_i}{1 + \alpha_A} - 1. \end{aligned} \tag{14}$$

In Figure 9, we show the numerous simulation result of RER between the PSM strategy and the honest mining strategy. The higher the γ is, the more likely the attackers can get more rewards than honest mining. When $\gamma = 1$, PSM can surely get more rewards than honest mining. If it can attract enough rational miners, the PSM attacker with enough mining power can still get more rewards than honest mining when $\gamma = 0$.

2) Simulation Results: To further verify the accuracy of our quantitative results, we implement a Monte Carlo simulator in java to verify our theoretical analysis. We simulate an attacker with $\alpha_A = 0.2$ and run the simulator over 10^9 rounds. The upper bond for error is 10^{-4}. The result is shown in Table III. The attacker's RER is almost the same as expected.

γ \ α_i	0.1	0.2	0.3	0.4
0	-29.167(-29.17)	-20.0(-20.0)	-12.5(-12.5)	-6.67(-6.67)
0.25	-18.96(-18.96)	-12.5(-12.5)	-7.29(-7.29)	-3.33(-3.33)
0.5	-8.75(-8.75)	-5.0(-5.0)	-2.08(-2.08)	0.0(0.0)
0.75	1.46(1.46)	2.5(2.5)	3.13(3.12)	3.33(3.33)
1	11.67(11.67)	10.0(10.0)	8.33(8.33)	6.67(6.67)

TABLE III. **Monte Carlo simulation result of attacker's relative extra reward in PSM against honest mining.** The attacker's mining power $\alpha_A = 0.2$. The values $x(y)$ indicates the attacker's RERs in simulations and theoretical analysis respectively.

B. Comparison with Selfish Mining

1) Quantitative Analysis: If the attacker with one private block chooses to follow the selfish mining strategy, then two cases may happen:

(1) Other miners find a new block, then the attacker releases the private block and starts the race (possibility α_R). In this case, the attacker can get the revenue of 1 block if other miners find the new block on its private branch (possibility $\gamma\alpha_R$) or get the revenue of 2 blocks if it finds the new block on its private branch (possibility α_A).

(2) The attacker finds the new block on its private branch and extends its leading to two or more blocks (probability α_A). The attacker's profit is determined as follows:

According to [10], if the attacker's lead is 2, others find a new block (probability α_R). Then the attacker can release the private branch and get revenue from 2 blocks. If the attacker's lead is more than 2, then every time other miners find a new block, the attacker releases one block until the lead is 2. Assuming the attacker finds $n-1$ blocks and extends the attacker's lead to n. Denote the possibility of the attacker keeping a lead of n blocks as P_n, then we have $P_n = (\frac{\alpha_A}{\alpha_R})^{n-1}P_1$. Since we only consider the case after the attacker finds one block, in our scenario, $P_1 = 1$. The overall revenue of this state is:

$$\sum_{n=3}^{\infty} P_n\alpha_R + 2P_2\alpha_R = \frac{\alpha_A^2}{1-2\alpha_A} + 2\alpha_A. \tag{15}$$

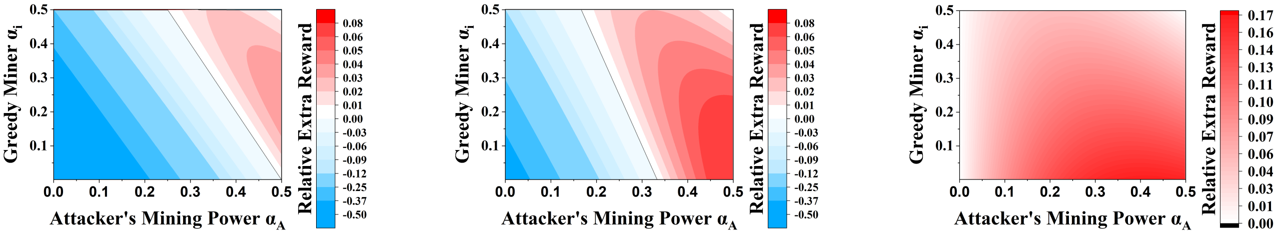
Overall, if choosing the selfish mining, the attacker's reward is:

$$R_A^S = \alpha_R(\gamma\alpha_R + 2\alpha_A) + \frac{\alpha_A^2}{1-2\alpha_A} + 2\alpha_A, \tag{16}$$

and the expected RER between PSM and selfish mining is

$$RER_A^{P,S} = \frac{\gamma\alpha_h^2 + 2\alpha_A\alpha_h + \alpha_i\alpha_h + 2\alpha_A + \alpha_i}{\alpha_R(\gamma\alpha_R + 2\alpha_A) + \frac{\alpha_A^2}{1-2\alpha_A} + 2\alpha_A} - 1. \tag{17}$$

In Figure 10, we show the numerous simulation result of the attacker's RER between following PSM and selfish mining. The PSM attacker can get a higher reward than the selfish miner when its mining power is relatively small. When the attacker's mining power is large enough, the possibility of

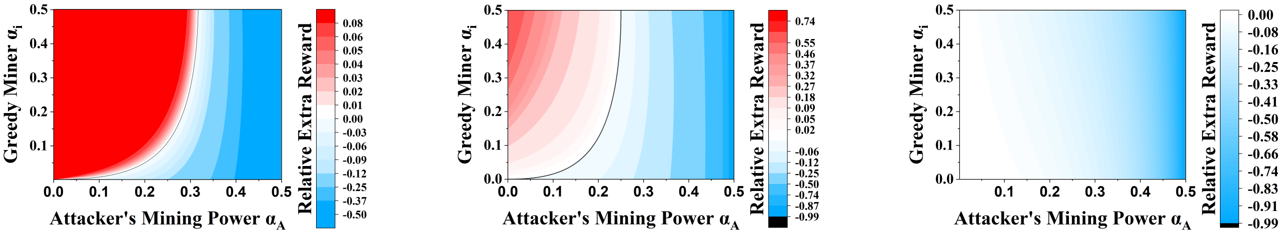


(a) Rushing ability γ=0

(b) Rushing ability γ=0.5

(c) Rushing ability γ=1

Fig. 9. **Attacker's relative extra reward between PSM and honest mining ($RER_A^{P,H}$).** The solid line represents no extra reward.



(a) Rushing ability γ=0

(b) Rushing ability γ=0.5

(c) Rushing ability γ=1

Fig. 10. **Attacker's relative extra reward between following PSM and selfish mining ($RER_A^{P,S}$).** The solid line represents no extra reward.

γ \ α_i	0.1	0.2	0.3	0.4
0	8.07(8.05)	22.04(22.03)	33.51(33.47)	42.35(42.37)
0.25	2.74(2.73)	10.93(10.92)	17.52(17.52)	22.54(22.54)
0.5	-1.06(-1.05)	3.01(3.01)	6.2(6.17)	8.42(8.43)
0.75	-3.88(-3.88)	-2.9(-2.89)	-2.3(-2.3)	-2.11(-2.11)
1	-6.08(-6.07)	-7.48(-7.48)	-8.88(-8.88)	-10.29(-10.28)

TABLE IV. **MONTE CARLO SIMULATION RESULT OF ATTACKER'S RELATIVE EXTRA REWARD BETWEEN PSM AND SELFISH MINING.** THE ATTACKER'S MINING POWER $\alpha_A = 0.2$. THE VALUES $x(y)$ INDICATES THE ATTACKER'S RERS IN SIMULATION AND THEORETICAL ANALYSIS RESPECTIVELY.

finding more than one block on its private branch becomes non-negligible. Thus, the revenue of selfish mining is higher than PSM. We propose an advanced PSM strategy to address this issue in Section VII.

2) Simulation Results: To further verify the accuracy of our quantitative results, assuming the attacker with a computation power of 0.2, we compare the Monte Carlo simulation result of the RER between PSM and selfish mining with our evaluation result. We run the java-based simulator over 10^9 rounds. The upper bond for error is 10^{-4}. The Monte Carlo simulation result is shown in Table IV. The attacker's RER is the same as expected.

VII. Advanced PSM Strategy

The PSM strategy is not likely to outperform selfish mining when the attacker's mining power is large enough. That is because, for the attacker, with the increment of its mining power, the possibility of finding more than one block on its private branch becomes non-negligible. To address this issue and further increase the profits of PSM for the attacker, in this section, we proposed an optimized PSM strategy named Advanced PSM (A-PSM).

A. Attack Overview

When miners find new blocks, the A-PSM attacker follows the selfish-mining-like strategy to publish the partial-released block instead of simply releasing the partial blocks' secrets. The attacker will keep the secret private until the lead of the private branch is no more than 2 blocks. Greedy miners can immediately release the block it finds, but it will not be recognized as a valid block until the secret of the prior partial block is released.

Specifically, if the honest miner finds a new block, two possible cases may happen: if the lead of the private branch is 2, the attacker will release all the partial blocks, and the system goes back to the single branch state. If the private branch's lead is more than 2, then the attacker and rational miners will continue working on its private branch until the lead is 2.

To avoid the cases where the length of the private branch grows faster than public branch, we extend our assumption that in the A-PSM scenario, the mining power of the attacker together with rational miners is no more than 50%.

We also extend the assumption that the attacker can promise that it will release the secret based on the length of both the private and the public branches. This assumption is reasonable because when launching mining attacks, attackers are motivated to maximize their revenue, and the secret computation mechanism in Section V further assures the malicious behavior is economically not worthwhile for the attacker. To further address the rational miner's concern, the attacker can also have the latest block height of the public branch via decentralized oracle [35] or reported by a greedy miner.

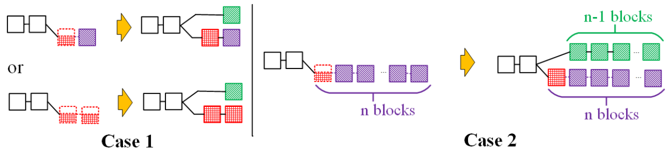


Fig. 11. **Partial block sharing with A-PSM strategy.** It assures the lower bound of attacker's revenue is almost the same as selfish mining.

B. A-PSM Reward

After the attacker shares the partial block, two possible cases may happen:

Case 1: Honest miners find the new block, then the attacker publishes the secret of the partial block and starts racing (probability α_h). Then there are four possible sub-cases:

- Honest miners find the new block on the public branch (probability $(1-\gamma)\times\alpha_h$). Attacker and greedy miners get the revenue of 0 blocks. Honest miners get the revenue of 2 blocks.
- Honest miners find the new block on the private branch (probability $\gamma\times\alpha_h$). The attacker gets the revenue of 1 block. Honest miners get the revenue of 1 block.
- Attacker finds the new block on the private branch (probability α_A). The attacker gets the revenue of 2 blocks.
- Greedy miners find the new block on the private branch (probability α_i). The attacker gets the revenue of 1 block, and the greedy miner gets the revenue of 1 block.

Case 2: Miners on the attacker's private branch (including greedy miners and the attacker) find the new block on the private branch and extends their leading to two or more blocks (probability $\alpha_A+\alpha_i$). The attacker and greedy miners' expected profits are determined as follows:

If the private branch's lead is 2, honest miners find a new block (probability α_h). Then the attacker releases all the partial blocks. In this case, the attacker will get the revenue from 1 block for sure, and the possibility of getting another block's revenue is $\frac{\alpha_A}{\alpha_A+\alpha_i}$. The rational miner's expected revenue is $\frac{\alpha_i}{\alpha_A+\alpha_i}$. If the private branch's lead is more than 2, then the attacker and the greedy miners keep mining on the private branch until the lead is 2. Assuming the miners on the private branch find $n-1$ blocks and extend the private branch's lead to n (probability α_A^{n-1}). According to [10], denote the possibility of the private branch keeping a lead of n blocks as P_n, then we have $P_n=(\frac{\alpha_A+\alpha_i}{\alpha_h})^{n-1}P_1$. Since we only consider the case after the attacker finds one block, in our scenario, $P_1=1$. The overall expected profit of the attacker in this state is:

$$\begin{aligned}&\frac{\alpha_A}{\alpha_A+\alpha_i}\sum_{n=3}^{\infty}P_n\alpha_h+(1+\frac{\alpha_A}{\alpha_A+\alpha_i})P_2\alpha_h\\=&\frac{\alpha_A}{\alpha_A+\alpha_i}\frac{(\alpha_A+\alpha_i)^2}{1-2(\alpha_A+\alpha_i)}+2\alpha_A+\alpha_i,\end{aligned}\tag{18}$$

and the rational miner's expected profit in this state is:

$$\begin{aligned}&\frac{\alpha_i}{\alpha_A+\alpha_i}\sum_{n=3}^{\infty}P_n\alpha_h+\frac{\alpha_i}{\alpha_A+\alpha_i}P_2\alpha_h\\=&\frac{\alpha_i}{\alpha_A+\alpha_i}\frac{(\alpha_A+\alpha_i)^2}{1-2(\alpha_A+\alpha_i)}+\alpha_i,\end{aligned}\tag{19}$$

Overall, if choosing the A-PSM strategy, the attacker's expected profit is:

$$\begin{aligned}R_A^{AP}&=\alpha_h(\gamma\alpha_h+2\alpha_A+\alpha_i)+\\&\frac{\alpha_A}{\alpha_A+\alpha_i}(\frac{(\alpha_A+\alpha_i)^2}{1-2(\alpha_A+\alpha_i)})+2\alpha_A+\alpha_i,\end{aligned}\tag{20}$$

the greedy miner's expected profit is:

$$R_i^{AP}=\alpha_h\alpha_i+\frac{\alpha_A}{\alpha_A+\alpha_i}(\frac{(\alpha_A+\alpha_i)^2}{1-2(\alpha_A+\alpha_i)})+\alpha_i.\tag{21}$$

C. Profit Analysis

Rational Miners' profits Analysis: For rational miners, when following the honest mining strategy, the expected profits are shown in Equation (3) (See Section IV-C). The RER between following the A-PSM and honest mining is:

$$RER_i^{AP,H}=\frac{\alpha_h\alpha_i+\frac{\alpha_i}{\alpha_A+\alpha_i}(\frac{(\alpha_A+\alpha_i)^2}{1-2(\alpha_A+\alpha_i)})+\alpha_i}{(2-\gamma)\alpha_R\alpha_i}-1.\tag{22}$$

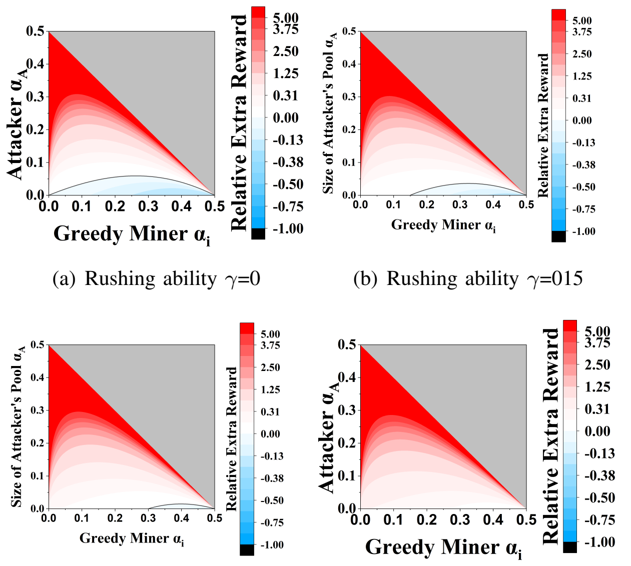
In Figure 12 we show the numerous simulation result of the miner's RER between being greedy and honest with an A-PSM attacker. The reward of being greedy is always greater than that of the honest strategy when $\gamma>0.5$.

Attacker's Profit Analysis: For the attacker, when following the honest mining strategy, the revenue is shown in Equation (13). The RER between following A-PSM and honest strategy is:

$$\begin{aligned}&RER_A^{AP,H}=\\&\frac{\alpha_h(\gamma\alpha_h+2\alpha_A+\alpha_i)+\frac{\alpha_i}{\alpha_A+\alpha_i}(\frac{(\alpha_A+\alpha_i)^2}{1-2(\alpha_A+\alpha_i)})+2\alpha_A+\alpha_i}{(2-\gamma)\alpha_h\alpha_i}-1.\end{aligned}\tag{23}$$

And the reward of following the selfish mining strategy is shown in Equation (16). The RER between following A-PSM and selfish mining strategy is:

$$\begin{aligned}&RER_A^{AP,S}=\\&\frac{\alpha_h(\gamma\alpha_h+2\alpha_A+\alpha_i)+\frac{\alpha_i}{\alpha_A+\alpha_i}(\frac{(\alpha_A+\alpha_i)^2}{1-2(\alpha_A+\alpha_i)})+2\alpha_A+\alpha_i}{\alpha_R(\gamma\alpha_R+2\alpha_A)+\frac{\alpha_A^2}{1-2\alpha_A}+2\alpha_A}-1.\end{aligned}\tag{24}$$



(a) Rushing ability γ=0

(b) Rushing ability γ=015

(c) Rushing ability γ=0.3

(d) Rushing ability γ=0.5

Fig. 12. **Rational miner's relative extra reward between being greedy and honest with an A-PSM attacker**($RER_i^{AP,H}$). The solid line represents no extra reward.

γ \ α_i	0	0.1	0.2	0.3
0	-64.31(-64.32)	-47.88(-47.88)	-31.37(-31.36)	-9.09(-9.09)
0.25	-45.91(-45.91)	-33.33(-33.33)	-20.23(-20.23)	-0.91(-0.91)
0.5	-27.5(-27.5)	-18.78(-18.79)	-9.09(-9.09)	7.28(7.27)
0.75	-9.1(-9.09)	-4.23(-4.24)	2.05(2.05)	15.45(15.45)
1	9.32(9.32)	10.3(10.3)	13.18(13.18)	23.64(23.64)

TABLE V. MONTE CARLO SIMULATION RESULT OF ATTACKER'S RELATIVE EXTRA REWARD BETWEEN A-PSM AND HONEST MINING. THE ATTACKER'S MINING POWER $\alpha_A = 0.1$. THE VALUES $x(y)$ INDICATES THE ATTACKER'S RERS IN SIMULATION AND THEORETICAL ANALYSIS RESPECTIVELY.

In Figure 13(a) and 13(b), we show the simulation result of the attacker's RER between following A-PSM and honest mining strategy when $\gamma = 0$ and 1 simultaneously. And the result of the attacker's RER between A-PSM and selfish mining is shown in Figure 13(c) and 13(d) simultaneously.

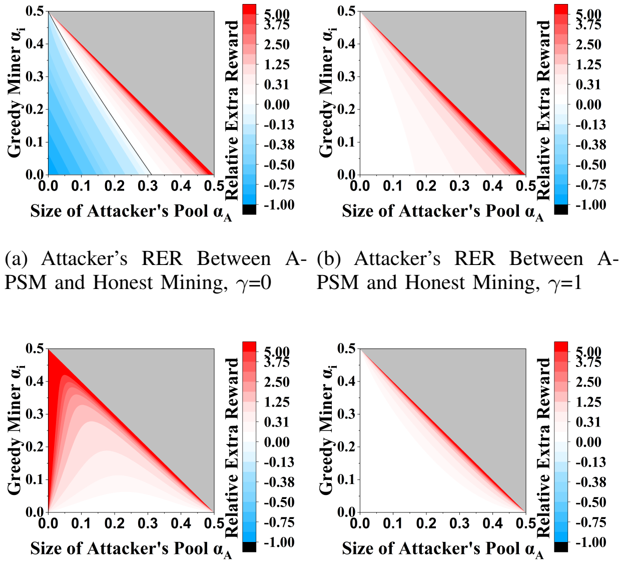
We compare the Monte Carlo simulation result of the A-PSM attack with honest mining and selfish mining profits for the attacker with a computation power of 0.1 over 10^9 rounds simultaneously. The upper bond for error is 10^{-4}. The comparison result with honest mining is shown in Table V, and the comparison result with selfish mining is shown in Table VI.

VIII. DOMINANT MINING STRATEGY

In an ideal model with known α_A and γ, we can get the "*Dominant Mining Strategy*" whose profit is higher than others. In this section, we analyze the conditions to make PSM and A-PSM to be the dominant mining strategy, comparing them with selfish mining and honest mining based on Equations (1), (13), (16), and (20). Note that A-PSM strategy requires rational miners to trust the attacker more than PSM. Here we make a stricter assumption for A-PSM strategy, i.e., $\alpha_A + \alpha_i \leq 0.5$.

γ \ α_i	0	0.1	0.2	0.3
0	-0.01(0.0)	46.04(46.07)	92.39(92.36)	154.8(154.78)
0.25	0.02(0.0)	23.25(23.25)	47.47(47.48)	83.17(83.19)
0.5	-0.04(0.0)	12.01(12.02)	25.39(25.39)	48.02(47.96)
0.75	0.0(0.0)	5.32(5.33)	12.26(12.25)	27.0(27.0)
1	0.01(0.0)	0.9(0.9)	3.54(3.53)	13.09(13.1)

TABLE VI. MONTE CARLO SIMULATION RESULT OF ATTACKER'S RELATIVE EXTRA REWARD BETWEEN A-PSM AND SELFISH MINING. THE ATTACKER'S MINING POWER $\alpha_A = 0.1$. THE VALUES $x(y)$ INDICATES THE ATTACKER'S RERS IN SIMULATION AND THEORETICAL ANALYSIS RESPECTIVELY.



(a) Attacker's RER Between A-PSM and Honest Mining, γ=0

(b) Attacker's RER Between A-PSM and Honest Mining, γ=1

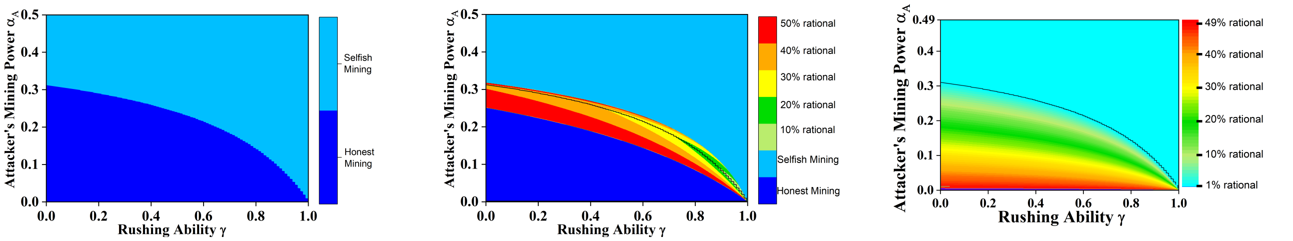
(c) Attacker's RER Between A-PSM and Selfish Mining, γ=0

(d) Attacker's RER Between A-PSM and Selfish Mining, γ=1

Fig. 13. **Attacker's relative extra reward.** The solid line represents no extra reward.

Figure 14 shows the simulation results of dominant mining strategy under different conditions. Specifically, Figure 14(a) shows the dominant strategy between selfish mining and honest mining in variant network settings. When the mining power of a selfish miner is larger than $1/3$, it can always get more profits than honest mining.

In Figure 14(b), we show in what conditions PSM can outperform both honest mining and selfish mining. For example, when the attacker's mining power $\alpha_A = 0.09$, and $\gamma = 0.9$, the attacker with rational miner controls only 10% of mining power can be the dominant mining strategy. When greedy miners control 50% of mining power, an attacker with 25% of mining power can outperform selfish and honest mining even if $\gamma = 0.$ 9 For the A-PSM strategy, Figure 14(c) shows the amount of rational mining power the attacker needed to outperform both selfish and honest mining in Figure 14(a). As we can see from Figure 14(c), A-PSM can outperform selfish mining even if the greedy miners only control less than 1% of mining power. The fraction of rational miners' mining power needed increases with the decrement of the attacker's mining power. When attacker's mining power is less than 1%, the amount of rational miner's mining power needed approaches



(a) Dominant strategy between selfish mining and honest mining.

(b) Dominant strategy among selfish mining, honest mining, and PSM with different fraction of rational miners.

(c) The amount of mining power needed for A-PSM to be the dominant strategy.

Fig. 14. **Dominant strategies for different α_A and γ values.** We also redraw the borderline between selfish and honest mining in (a) in (b) and (c) for comparisons.

49%. But the overall mining power of greedy miners and the attacker is always no more than 50%.

In the reality, a blockchain network is dynamic. α_A, α_i and γ could change dynamically. An attacker can use network measurement [27], [26] or historical information, e.g., its mining power ratio in the past 1 hour, to infer current network status and select the attack strategy accordingly.

IX. Discussion

A. Multiple Attackers

Here, we consider a model with two attackers.

The rational miners can choose to work on one of the attacker's private branches or continue working on the public branch.

First, we consider the case that two attackers release the partial block simultaneously. Assuming there is an attacker A with mining power α_A, and rushing ability γ_A. Attacker B with mining power α_B and rushing ability γ_B. Rational miner i with mining power α_i considers that the rest of the miners are honest, which means $\alpha_h = 1 - \alpha_A - \alpha_B - \alpha_i$.

For PSM, the relative extra reward between choosing attacker A and Honest mining is $\frac{2-\alpha_A-\alpha_B-\alpha_k}{(1-\alpha_A-\alpha_B)(2-\gamma_A-\gamma_B)} - 1$. And the RER between mining in attacker A and B's branch is always 0. That means the variance of mining power between different attackers has no impact on rational miners. Joining either one of them could assure its profits.

For A-PSM, the RER between choosing attacker A and B is $\frac{\alpha_A-\alpha_B}{4\alpha_i^2+(4\alpha_B+4\alpha_A-4)\alpha_i+(4\alpha_A-2)\alpha_B-2\alpha_A+1}$, it is always more beneficial for rational miners to work with the attacker with larger mining power.

Then we consider one of the attackers to find the new block first. Assuming attacker A finds the new block at time t_a, the attacker B finds the new block at time t_b, duration $T_d = t_b - t_a > 0$. Since the rational miner trust both attackers equally, joining the attacker A's private branch means the rational miner can have $\alpha_h R(T_d)\alpha_i + \alpha_i R(T_d)$ more expected revenue during the period T_d. Note that the definition of $R(T)$ is given in Equation (12). Thus, greedy miners will tend to join the attacker A's private branch first to assure higher profits for both PSM and A-PSM strategies. If the rational miners do not find the new block during the period T_d, with the PSM strategy, the greedy miners have no motivation to switch the working branch. For the A-PSM strategy, the greedy miners need to re-evaluate their profits and work with the attacker with higher mining power.

B. Mitigation

One of the straightforward ways to prevent PSM attacks is to forbid the broadcasting of partial block data in the target blockchain network. But as we have stated before, the partial block data do not need to be shared through the target network, so it would not be easy to forbid the broadcast of partial block data. However, it is possible that honest miners declare the block invalid. It is not easy to prevent rational miners from sharing partial block data with others. Thus, some honest miners can know the hash value of the attacker's private block. These honest miners refuse to mine on the attacker's branch when race occurs. The attacker's rushing ability γ is reduced in this case.

As demonstrated in section V, we denote the hidden data as *secret*. Theoretically, other miners could calculate a valid secret and announce it after getting the partial block data, especially for those honest miners with relatively sizeable computational power.

Assuming that the attacker publishes a partial block at time t_0, the honest miner calculates the secret at t_1. Let $T = t_1 - t_0$. The expected profits of greedy miners can be expressed as $\alpha_h R(T)\alpha_i + \alpha_i R(T)$. Let the greedy miner's mining power is 0.1. When $T \to 0$, the greedy miners' expected extra revenue approaches 0.

If an honest miner conducts such a countermeasure, then none of the greedy miners will take the risk of getting negligible profits. When $\alpha_i = 0$, for the attacker following the A-PSM strategy, its revenue downgrades to selfish mining. The revenue following the PSM strategy is less than selfish mining (see Section VI-B for details).

Another possible defense method against PSM and A-PSM attacks is to reduce the efficiency of partial block dissemination. Honest miners can drop the partial block when receiving it. Counter-attackers can manage to announce fake information to mislead rational miners.

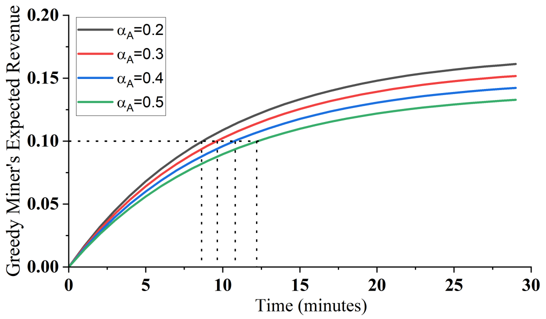


Fig. 15. **The greedy miners expected revenue changes with allowed time in attacker's private branch.** When $T \to 0$, none of the greedy miners will take the risk to take the negligible profits.

X. Conclusions

In this paper, we propose a new mining attack called Partial Selfish Mining (PSM). Based on the idea of partial block sharing, PSM attack allows both attacker and greedy miners to earn an unfair share of reward. PSM is also feasible to launch because it has two mechanisms to guarantee greedy miners' profit such that they will follow attacker's private branch. The proposed Advanced PSM (A-PSM) can further improve attacker's profit to be no less than selfish mining, which makes it a profitable alternative to current selfish mining in mining-related attacks. To mitigate the partial selfish mining, we discussed some possible countermeasures. However, a practical solution remains to be open.

References

[1] L. Aumayr, P. Moreno-Sanchez, A. Kate, and M. Maffei, "Blitz: Secure multi-hop payments without two-phase commits," in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 4043–4060.

[2] C. Bendiksen, S. Gibbons, and E. Lim, "The bitcoin mining network-trends, marginal creation cost, electricity consumption & sources," CoinShares Research, vol. 21, pp. 3–19, 2018.

[3] D. Bradbury, "The problem with bitcoin," Computer Fraud & Security, vol. 2013, no. 11, pp. 5–8, 2013.

[4] L. Breidenbach, P. Daian, F. Tramèr, and A. Juels, "Enter the hydra: Towards principled bug bounties and exploit-resistant smart contracts," in 27th USENIX Security Symposium (USENIX Security 18), 2018, pp. 1335–1352.

[5] M. Campanelli, R. Gennaro, S. Goldfeder, and L. Nizzardo, "Zero-knowledge contingent payments revisited: Attacks and payments for services," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 229–243.

[6] CCN, "Japanese cryptocurrency monacoin hit by selfish mining attack," https://www.ccn.com/japanese-cryptocurrency-monacoin-hit-by-selfish-mining-attack/.

[7] Coinwarz, "Bitcoin hashrate chart," https://www.coinwarz.com/mining/Bitcoin/hashrate-chart.

[8] Etherscan, "How many bitcoins are there and will they ever run out?" https://capitalcounselor.com/how-many-Bitcoins-are-there/, february 23, 2022.

[9] I. Eyal, "The miner's dilemma," in 2015 IEEE Symposium on Security and Privacy. IEEE, 2015, pp. 89–103.

[10] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in International Conference on Financial Cryptography and Data Security. Springer, 2014, pp. 436–454.

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

[12] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 3–16.

[13] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in 24th USENIX Security Symposium (USENIX Security 15), 2015, pp. 129–144.

[14] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in Proceedings of the 2012 ACM SIGSAC Conference on Computer and Communications Security, 2012, pp. 906–917.

[15] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in 25th USENIX Security Symposium (USENIX Security 16), 2016, pp. 279–296.

[16] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (FAW) attacks on bitcoin," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 195–209.

[17] A. Lewis-Pye and T. Roughgarden, "How does blockchain security dictate blockchain implementation?" in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 1006–1019.

[18] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden markov decision process," International Journal of Intelligent Systems, vol. 36, no. 7, pp. 3596–3612, 2021.

[19] K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in International Conference on Financial Cryptography and Data Security. Springer, 2017, pp. 264–279.

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

[21] M. Mirkin, Y. Ji, J. Pang, A. Klages-Mundt, I. Eyal, and A. Juels, "BDoS: Blockchain denial-of-service," in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 601–619.

[22] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Decentralized Business Review, p. 21260, 2008.

[23] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016, pp. 305–320.

[24] K. A. Negy, P. R. Rizun, and E. G. Sirer, "Selfish mining re-examined," in International Conference on Financial Cryptography and Data Security. Springer, 2020, pp. 61–78.

[25] obscuren, "[release/1.3.4] core: Added new td strategy which mitigate the risk for selfish mining," https://github.com/ethereum/go-ethereum/commit/bcf565730b1816304947021080981245d084a930.

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

[27] M. Saad, S. Chen, and D. Mohaisen, "Syncattack: Double-spending in bitcoin without mining power," in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 1668–1685.

[28] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in International Conference on Financial Cryptography and Data Security. Springer, 2016, pp. 515–532.

[29] C. F. Torres, R. Camino et al., "Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain," in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 1343–1359.

[30] I. Tsabary, M. Yechieli, A. Manuskin, and I. Eyal, "MAD-HTLC: Because HTLC is crazy-cheap to attack," in 2021 IEEE Symposium on Security and Privacy (S&P). IEEE, 2021, pp. 1230–1248.

[31] X. Wang, V. V. Muppirala, L. Yang, S. Kannan, and P. Viswanath, "Securing parallel-chain protocols under variable mining power," in

Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 1700–1721.

[32] G. Wood et al., “Ethereum: A secure decentralised generalised transaction ledger,” Ethereum project yellow paper, vol. 151, no. 2014, pp. 1–32, 2014.

[33] S. Wu, Y. Chen, M. Li, X. Luo, Z. Liu, and L. Liu, “Survive and thrive: A stochastic game for DDoS attacks in bitcoin mining pools,” IEEE/ACM Transactions on Networking, vol. 28, no. 2, pp. 874–887, 2020.

[34] Y. Wu, X. Xu, L. Qian, B. Ji, Z. Shi, and W. Jia, “Revenue-sharing based computation-resource allocation for mobile blockchain,” in IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops. IEEE, 2020, pp. 56–61.

[35] F. Zhang, D. Maram, H. Malvai, S. Goldfeder, and A. Juels, “DECO: Liberating web data using decentralized oracles for TLS,” in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 1919–1938.

[36] M. Zhang, X. Zhang, Y. Zhang, and Z. Lin, “Txspector: Uncovering attacks in ethereum from transactions,” in 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 2775–2792.