The Impact Analysis of Multiple Miners and Propagation Delay on Selfish Mining

Qing Xia[†‡], Wensheng Dou[†‡], Tong Xi[§‡], Jing Zeng[†], Fengjun Zhang[†¶], Jun Wei[†‡], Geng Liang[†¶]
[†]Institute of Software, Chinese Academy of Sciences, Beijing, China
[‡]University of Chinese Academy of Sciences, China
[¶]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[§]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[†]{xiaqing2018, wensheng, zengjing, fengjun, weijun, lianggeng}@iscas.ac.cn, [§]xitong@iie.ac.cn

Abstract—**Bitcoin has emerged as a popular decentralized cryptocurrency and attracted much attention from the public. Bitcoin embodies the Nakamoto consensus to reach an agreement about its blockchain ledger. However, the Nakamoto consensus can suffer from selfish mining attacks. Existing studies on selfish mining usually assume that the total mining power is divided into two parts (i.e., honest and selfish), and ignore propagation delay among miners. The assumptions cannot reflect real-world scenarios, in which multiple miners generate blocks at a fixed interval and propagate them with certain delay. Therefore, it is unknown how the practical factors, i.e., multiple miners and propagation delay, can affect selfish mining.**

In this paper, we explore the impact of multiple miners and propagation delay on selfish mining. First, we propose a new selfish mining strategy that can handle these factors. Second, we design a simulation approach to analyze the performance of the new selfish mining strategy. From our empirical study we observe many interesting findings that can be utilized in combating selfish mining. For example, the blockchain system with a higher orphan rate is more vulnerable to the selfish mining attack.

Index Terms—**Blockchain, consensus protocol, selfish mining, propagation delay**

I. Introduction

Blockchain is a novel ledger-sharing technology that is used as the core component in Bitcoin [1]. In blockchain, each participant maintains a continuously-growing ledger, which consists of a list of blocks. In the ledger, each block contains a cryptographic hash pointer to its previous block. Since any modification to a block will invalidate all its subsequent blocks, blockchain can provide immutability and traceability. Therefore, blockchain is regarded as a promising technology for building multi-party trusted distributed systems. Nowadays, blockchain has been widely applied in various domains, e.g., government management, cross-border remittances and supply chain tracing.

Nakamoto consensus [1] is one of the most widely-used protocols to guarantee the consistency and security of blockchain. Nakamoto consensus regards the longest chain as the main chain. Honest miners try to solve a cryptographic hash puzzle after the most recent block in their local longest chain. Once successful, honest miners broadcast the newly-generated block immediately to extend the longest chain. When multiple blocks refer to the same parent block, i.e., fork blocks, are generated, the branches with equal length occur. In this case, honest miners work on the first arrived branch temporarily until one branch becomes the longest. Only the longest chain block is accepted, and other fork blocks are discarded as orphan blocks. The blockchain system provides a fixed amount of cryptocurrency as reward to the miner of an accepted block.

In blockchain, the possibility to generate a block is proportional to a miner's mining power. Therefore, a miner's reward is generally considered to be proportional to its mining power. However, a miner can adopt the selfish mining strategy [2] to obtain a higher reward than its fair share. Instead of broadcasting a newly-generated block immediately, the selfish miner withholds the new block first and reveals it later. By intentionally creating equal branches with the hidden blocks, the selfish miner forces honest miners to waste their mining power on the stale longest branch and hence improves its reward. Existing study shows that a selfish miner with more than 33% of the total mining power can gain a higher reward [2], which significantly destroys confidence in the blockchain.

The original selfish mining strategy [2] and its variants [3]–[7] usually adopt the Markov chain model to analyze the performance of selfish mining. In this model, they divide the total mining power into two parts, i.e., honest and selfish, and ignore dynamically-changing propagation delay among miners. The model involves three assumptions. First, all miners hold the same blockchain view at the same time. Second, a fixed fraction of honest power always mines on the selfish branch when there is an equal branch. Third, it is impossible that multiple miners can generate multiple branches with their fork blocks. Specifically, the branches can only occur when the selfish miner creates them intentionally. These assumptions are required for the numerical simulation based on the Markov chain model, but do not hold in the real-world blockchain system. Therefore, the Markov-based strategies [2]–[7] cannot work in the real-world scenarios, and it is still unknown how the practical factors, i.e., multiple miners and propagation delay, affect the performance of selfish mining.

In this paper, we first propose a new selfish mining strategy that can handle the practical factors. Second, we design a simulation approach by analyzing the blockchain execution process to simulate the blockchain scenarios. We then apply the new strategy into the simulation system and obtain many interesting findings from our empirical study. The main find-

TABLE I
THE STUDIES RELATED TO SELFISH MINING EVALUATION.

Related work	Factors	Findings		
		Same	New	Diff
Chang [8]	multiple honest miners	-	1, 3	2
Liu et al. [6], Bai et al. [7]	multiple selfish miners	4	-	-
Göbel et al. [9]	propagation delay	6, 9	5, 7, 8	-

ings are shown as follows. (1) A miner with sufficient mining power has an inherent mining advantage even under honest mining. For example, in the 10-miner system, the honest miner with 45% of the mining power can obtain 46.83% of the total reward. (2) Selfish mining performs better in the blockchain system with more miners and a larger propagation delay. For example, the profit threshold for launching selfish mining attacks decreases from 31% to 29% when the number of miners increases from 5 to 100. (3) The blockchain system with a higher orphan rate is more vulnerable to selfish mining. For example, the profit threshold is reduced to 21% when the system orphan rate reaches to 40.37%. These findings are useful for combating selfish mining attacks.

We summarize the main contributions as follows.

- We propose a new selfish mining strategy to handle blockchain scenarios with multiple miners and propagation delay.
- We propose a simulation approach to simulate the real-world blockchain scenarios.
- We evaluate the performance of the selfish mining strategy on the simulation system and obtain many interesting findings from our empirical study.

II. Related Work

In this section, we discuss related work that is close to our findings.

Table I compares the studies on selfish mining evaluation. Chang [8] considered the influence of multiple honest miners, whose main finding is contrary to our finding 2. The reason has been discussed in Section VI-B. Besides, finding 1 and 3 are our new findings. Liu et al. [6] and Bai et al. [7] considered multiple selfish miners, whose main finding is similar to our finding 4. Göbel et al. [9] investigated how propagation delay affect selfish miner's network advantage, whose main findings are similar to finding 6 and 9. Besides, finding 5, 7, 8 are our new findings. Different from Göbel et al. [9], we also consider different block intervals in the RPD parameters.

Some research efforts have been put on exploring the new selfish mining strategies. Nayak et al. [3] proposed stubborn mining, in which the miner insists on its private selfish branch despite losing the lead advantage. Sapirshtein et al. [5] optimized selfish mining in different parameter space. Negy et al. [4] proposed intermittent selfish mining when facing difficulty adjustment. Niu and Feng [10], Ritz and Zugenmaier [11] evaluated the impact of uncle reward in Ethereum on selfish mining, and found Ethereum is more vulnerable to selfish mining than Bitcoin. However, these strategies cannot work in the real-world blockchain scenarios with multiple miners and propagation delay. In this paper, our proposed new strategy can handle the real-world situations. We can also improve these new strategies to make them work in the real-world scenarios, and analyze the impact of practical factors. We leave them in the further work.

$$SHA\text{-}256(ParentHash \parallel Metadata \parallel \boldsymbol{Nonce}) \leq Target$$

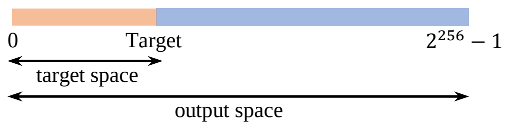


Fig. 1. Proof of Work (PoW) in Blockchain.

III. Background

In this section, we briefly introduce Nakamoto consensus and the selfish mining strategy.

A. Nakamoto Consensus

Nakamoto consensus [1] is one of the most widely-used protocols to reach the agreement on the blockchain ledger. Nakamoto consensus utilizes Proof of Work (PoW) to determine who can generate the new block. A new block must contain a valid solution to the cryptographic hash puzzle, i.e., PoW puzzle. Miners compete against each other to find the valid solution. The PoW puzzle is shown as Fig. 1. $ParentHash$ denotes the hash pointer of the parent block. $Metadata$ includes the necessary information of the block, e.g., version, timestamp, Merkle root of the transaction list. $Nonce$ denotes a random number. $Target$ is an integer from 0 to $2^{256}-1$, representing the puzzle's difficulty. $Target$ is dynamically adjusted to ensure that blocks are generated at a fixed interval, e.g., 10 minutes in Bitcoin. Output space contains all possible results of the SHA-256 hash function, while target space only contains results below the $Target$. A valid PoW puzzle solution means a $Nonce$ satisfies the inequation in Fig. 1, i.e., the block's hash value calculated by the SHA-256 function falls into the target space.

Since multiple miners compete on solving the same PoW puzzle independently, they may generate fork blocks that refer to the same parent block. Thus, the branches occur. To guarantee consistency, Nakamoto consensus regards the longest chain as the main chain. Blocks in the longest chain are valid and accepted. Blocks outside the longest chain are discarded as orphan blocks, whose transactions are returned to miners' transaction pool and wait for re-processing. When facing branches, honest miners work on the first arrived one temporarily until the longest chain is determined.

Due to the collision-resistance property of hash functions [12], a miner's best choice to find the valid PoW solution is to randomly try $Nonce$ in Fig. 1. Therefore, the probability of finding the PoW solution is proportional to the miner's mining power. To compensate for miners' computing cost, the blockchain system provides a fixed amount of cryptocurrency

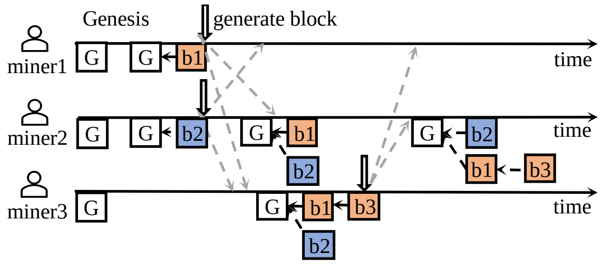


Fig. 2. Blockchain execution process in the system with three miners.

to the miners of the main chain blocks. While orphan blocks are not rewarded.

Fig. 2 shows a simplified blockchain execution process with three miners. Initially, the blockchain is initialized with only the genesis block. Once a miner, e.g., $miner_1$, generates a new block $b1$ by solving the PoW puzzle, it immediately broadcasts $b1$ through the peer-to-peer (P2P) network. It will take some time, i.e., propagation delay, for other miners to receive the block. Once the miner receives a new block, it will update its blockchain and work on the longest chain. Since $miner_2$ has generated the block $b2$ before receiving $b1$, a branch occurs. After block $b3$ is generated, the branch problem is solved and block list (G, b1, b3) is regarded as the main chain.

In blockchain, three factors can affect its execution process.

- **Block generation time**: The time required for a miner to generate a block.
- **Block propagation delay**: The time to propagate a block to another miner.
- **Mining strategy**: A miner's actions when generating and receiving blocks.

As discussed earlier, block generation time is proportional to a miner's mining power and is affected by the difficulty of the PoW puzzle. Block propagation delay can be affected by multiple factors, e.g., network topology, propagation mechanism, and block size [13]. For the mining strategy, honest miners follow the Nakamoto consensus mentioned before. However, a miner can adopt different strategies, e.g., selfish mining [2].

B. Selfish Mining Strategy

Selfish mining was proposed by Eyal and Sirer [2]. In selfish mining, instead of releasing newly-generated blocks immediately, a selfish miner can intentionally withhold its new blocks, and works on its private branch. In the same time, honest miners work on the stale public branch and waste their mining power. Thus, the selfish miner can obtain higher revenue.

To analyze the performance of selfish mining, original selfish mining (SM) [2] and its variants [3]–[7] usually divide the total mining power into two parts, i.e., honest and selfish. Thus, there are only two mining groups in the analysis model. We use α to denote the fraction of mining power controlled by the selfish miner, and β to denote the power controlled by the honest miner. We assume the selfish miner's network advantage γ as the fraction of honest mining power that works on the selfish branch when two branches are equal.

TABLE II
THE SELFISH MINER'S ACTIONS.

$lead$	**Generate block $b1$**	**Receive block $b2$**
0	withhold $b1$	add $b2$ to blockchain
0'	release $b1$	add $b2$ to blockchain
1	withhold $b1$	release its hidden block
2	withhold $b1$	release its two hidden blocks
≥ 3	withhold $b1$	release its first hidden block

The selfish miner takes different actions according to its lead advantage $lead$, which is described as follows.

$$lead = len(private\ branch) - len(public\ branch) \quad (1)$$

Table II summarizes the selfish miner's detailed actions in SM. Note that, both $lead = 0$ and $lead = 0'$ denotes the selfish miner has no private block. The difference is that at $lead = 0$, there is a unique longest chain shared by all miners. And at $lead = 0'$, there are two branches in the public chain that are created by the honest miner and the selfish miner. In this case, γ fraction of honest power will mine on the selfish miner's branch. When $lead = 0$, the selfish miner withholds its new block. If it receives the honest miner's new block, it will update its blockchain and keep up-to-date. When $lead = 0'$, there are two branches, so the selfish miner always releases its new block, and tries to become the longest chain. If it receives the honest miner's new block, it will update its blockchain.

When $lead \geq 1$, the selfish miner continues to withhold its new block. At $lead = 1$, if it receives the honest miner's new block, it immediately releases its hidden block, and creates a branch, thus $lead$ becomes 0'. So, a fraction of honest miners ($\gamma \cdot \beta$) will work on the selfish miner's branch, and only $(1-\gamma) \cdot \beta$ of honest power works on the honest branch. At $lead = 2$, if it receives a new block, it immediately releases two hidden blocks. So, its private chain becomes the longest, and $lead$ becomes 0. When $lead \geq 3$, if it receives a new block, it will release its first hidden block. In this case, the selfish miner can always hold the longest chain.

When there is no honest power mines on the selfish branch, i.e., $\gamma = 0$, a miner requires at least 33% of mining power to launch the selfish mining attack [2]. The required mining power further decreases as γ increases.

IV. Selfish Mining Strategy With Multiple Miners and Propagation Delay

The real-world blockchain system usually contains multiple miners. For example, in Bitcoin, there are ten large mining pools accounted for more than 98% of mining power and lots of other miners [14]. Once a miner broadcasts its new block, other miners may receive the block at different timings due to the propagation delay, which is dynamically-changing as the network environment changes. Since miners may hold different blockchain views at the same time, the blockchain forks occasionally [15].

However, existing studies [2]–[7] usually adopt the Markov chain model to analyze selfish mining, whose key assumptions do not hold in the real-world blockchain scenarios.

TABLE III
THE SELFISH MINER'S ACTIONS IN REAL-WORLD SCENARIOS.

$lead$	Generate block $b1$	Receive new block(s)
0	withhold $b1$	add block(s) to blockchain
0'	release $b1$	add block(s) to blockchain
1	withhold $b1$	$lead = 1$, keep its hidden block
		$lead = 1'$, release its hidden block
		$lead \leq 0$, release its hidden block
2	withhold $b1$	$lead = 2$, keep its two hidden blocks
		$lead \leq 1$, release its two hidden blocks
$N (N \geq 3)$	withhold $b1$	$2 \leq lead \leq N$, release its first $(N - lead)$ hidden blocks
		$lead \leq 1$, release its N hidden blocks

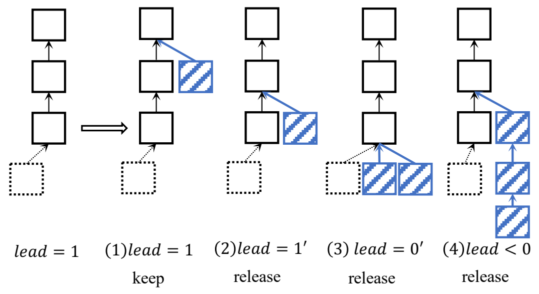


Fig. 3. Actions taken by the selfish miner at $lead = 1$. Blocks with the solid lines, dotted lines represent public blocks and selfish miner's hidden blocks, respectively. Blocks with the blue diagonal lines represent the received blocks.

- **Assumption 1**. All miners hold the same blockchain view at the same time. In the system with the propagation delay, miners may have different blockchain views as discussed earlier.
- **Assumption 2**. A fixed fraction of the honest power, i.e., γ, always works on the selfish branch when there is an equal branch. In the blockchain system, γ is dynamically-changing and unknown.
- **Assumption 3**. It is impossible that multiple miners can generate multiple branches, and the branches can only occur when the selfish miner creates them intentionally. However, due to propagation delay among multiple miners, branches among all miners can occur.

To overcome the above limitations, we propose an improved selfish mining strategy (ISM), the first practical mining strategy that can work in real-world blockchain scenarios. Table III shows the selfish miner's actions in ISM, according to its lead advantage $lead$. Different from SM in Table II, the selfish miner must handle the stale block and multiple blocks due to the delay. We describe the details of ISM as follows.

When $lead = 0$ or $lead = 0'$, the selfish miner has no hidden block and hence takes the same action as in SM discussed in Section III-B. Specifically, When $lead = 0$, the selfish miner withholds the new block. When $lead = 0'$, the selfish miner releases its new block. When $lead = 0$ or $lead = 0'$, it updates the blockchain after receiving the new block.

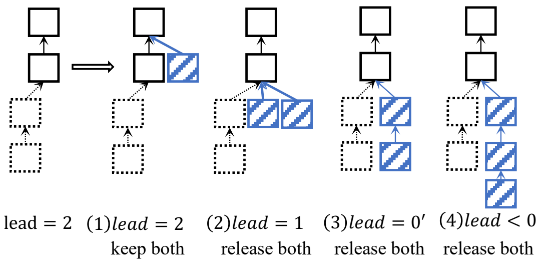


Fig. 4. Actions taken by the selfish miner at $lead = 2$.

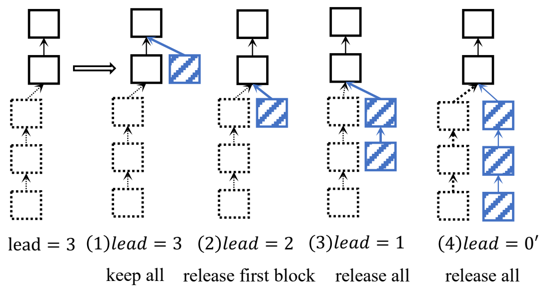


Fig. 5. Actions taken by the selfish miner at $lead = 3$.

When $lead = 1$, the selfish miner continues to withhold the new block and extends its private chain. Fig. 3 shows the cases when the selfish miner receives blocks. In case 1, the selfish miner receives a stale block, and still has the lead advantage. Therefore, it still keeps the hidden block. In case 2, the selfish miner receives a new block with the same height as the hidden block's parent block. This case is denoted as $lead = 1'$, in which the selfish miner has one lead block but the public blockchain forks. To become the longest chain, the selfish miner releases its hidden block. In case 3 and 4, the selfish miner loses its lead advantage. Thus it releases the hidden block, and keeps up-to-date with the public blockchain.

When $lead = 2$, the selfish miner still withholds its new block. Fig. 4 shows the cases when the selfish miner receives blocks. In case 1, the selfish miner maintains the lead advantage, e.g., receiving a new block with the same height as its parent block. In this case, it keeps its hidden blocks. In case 2, the selfish miner receives new blocks with the same height as its first hidden block, and its lead decreases to 1. In this case, it releases both hidden blocks and tries to become the longest chain. In case 3 and 4, the selfish miner loses its lead advantage, i.e., $lead \leq 0$. Thus, it releases both hidden blocks and keeps up-to-date.

When $lead = 3$, the selfish miner still withholds its new block. Fig. 5 shows the cases when the selfish miner receives blocks. In case 1, the selfish miner receives a new block with the same height as its parent block. In this case, it keeps the hidden blocks. In case 2, the selfish miner receives a new block with the same height as its first hidden block, and its lead decreases to 2. In this case, it releases the first hidden blocks to catch the public branch. In case 3, the selfish miner receives two blocks with the same height as its first two hidden blocks, and its lead decreases to 1. In this case, it releases all hidden blocks and tries to become the longest chain. In case 4 and even worse case, the selfish miner loses its lead advantage, i.e., $lead \leq 0$. Thus, it releases all hidden blocks, and keeps up-to-date.

Without loss of generality, we can infer the selfish miner's actions at $lead = N (N \geq 3)$ with the analysis at $lead = 3$. When $lead = N$, the selfish miner continues to withhold its new blocks. After receiving blocks from other miners, if the selfish miner's lead decreases but at least 2, i.e., $2 \leq lead \leq N$, it releases the first $N - lead$ hidden blocks. However, if the selfish miner's lead decreases to less than 2, i.e., $lead \leq 1$, it releases all hidden blocks.

V. Blockchain Simulation

To evaluate the performance of ISM, we propose a blockchain simulation approach considering multiple miners and propagation delay. As discussed in Section III-A, three factors can affect the blockchain execution process, i.e., block generation time, propagation delay and mining strategy. In this section, we first discuss how we simulate block generation time and propagation delay. Then, we explain how to implement the simulation approach.

A. Block Generation Time

The previous study [16] has demonstrated that the block generation time is proportional to a miner's mining power, and is affected by the difficulty of the PoW puzzle. As Equation (2) shows, a miner's block generation time t_i follows the shifted geometric distribution with the probability p_i.

$$t_i \sim Geo(p_i);\ p_i = \epsilon \cdot h_i \tag{2}$$

In Equation (2), ϵ denotes the success probability of a single hash operation on the PoW puzzle that can be regarded as a Bernoulli trial. The value of ϵ changes with the difficulty of the PoW puzzle. h_i denotes the number of hash operations that a miner can perform within a second. Since miners' hash rate h_i is hard to be controlled, the equation is not available for the blockchain simulation. Therefore, we make the further derivation.

First, we introduce an intermediate variable H denoting the total mining power. Hence, the probability p_i can be transformed into Equation (3).

$$p_i = \epsilon \cdot H \cdot r_i \tag{3}$$

In Equation (3), r_i denotes the fraction of mining power controlled by the miner, i.e., h_i/H. Since ϵ is the success probability of a single hash operation, $1/\epsilon$ represents the expected hash operations until finding a PoW solution. Assuming that blocks are generated at a fixed interval G, the total mining power H equals to $\frac{1/\epsilon}{G}$ [17]. Therefore, the probability p_i can be transformed into Equation (4).

$$p_i = \frac{r_i}{G} \tag{4}$$

Equation (4) shows that the probability p_i in the shifted geometric distribution is the ratio of miner's mining power ratio to the block interval. The two parameters can be set flexibly to simulate different blockchain systems.

B. Block Propagation Delay

Blockchain systems usually adopt the gossip protocol for communication as Bitcoin [18]. Decker and Wattenhofer found that the propagation delay in the Bitcoin system fits the exponential curve with the median time of 12.6 seconds, and the median time changes with the block size [13]. Thus, different blockchain systems may have different median time for propagation delay [19]. In our simulation system, we divide the continuous time into discrete one, i.e., seconds. So, the exponential distribution is discretized into the shifted geometric distribution as follows, in which miners can take varied time to receive the same block.

$$delay \sim Geo(medianTime) \tag{5}$$

C. Simulation Algorithm

In the simulation system, each miner runs the blockchain simulation algorithm and all miners form a distributed blockchain system. In the following, we describe the simulation algorithm for the selfish miner, as shown in Algorithm 1. Note that, the simulation algorithm for the honest miner is similar to Algorithm 1, but uses different mining strategy.

At first, the selfish miner's blockchain is initialized with only the $genesis$ block (Line 1). The miner works on the $genesis$ block, i.e., $curChain = genesis$ (Line 4). Its next block generation time $genTime$ is calculated by the shifted geometric distribution randomly discussed in Section V-A (Line 5). The miner keeps generating and receiving new blocks until $totalBlock$ blocks are included in its blockchain (Line 6−23). For each iteration, it checks whether the new block can be generated based on $genTime$ (Line 7). If yes, it generates a new block, and adds into the blockchain (Line 8−9). Based on ISM, the selfish miner can decide whether the new block can be released to other miners or not (Line 10−13). If the miner decides to withhold the block, it puts the block into $hiddenBlocks$ and does not release. Otherwise, it broadcasts the block immediately, which will be added into other miners' $incomingBlocks$. The propagation delay is calculated by the shifted geometric distribution as discussed in Section V-B (Line 25−28).

The selfish miner further checks its $incomingBlocks$. If a block in $incomingBlocks$ can be received, the selfish miner adds the block into its blockchain (Line 14−17). Then, based on ISM, the miner decides which hidden blocks should be released to compete with the incoming block (Line 18−20). Finally, the miner checks whether the chain it works on, i.e., $curChain$, has changed (Line 21). If yes, the miner updates $curChain$ to start a new iteration (Line 22−23).

Algorithm 1: Blockchain simulation algorithm for a selfish miner.

```
1  $blockchain \leftarrow genesis$
2  $incomingBlocks \leftarrow NULL$
3  $hiddenBlocks \leftarrow NULL$
4  $curChain \leftarrow genesis$
5  $genTime \leftarrow curTime() + random(Geo(r_i/G))$
6  while $blockchain.size() < totalBlock$ do
7      if $curTime() \geq genTime$ then
8          $block \leftarrow generateBlock(curChain)$
9          $blockchain.add(block)$
10         if $shouldBeReleased(blockchain, block)$ then
11             broadcast($block$)
12         else
13             $hiddenBlocks.add(block)$
14     for $block \in incomingBlocks$ do
15         if $curTime() \geq block.incomingTime$ then
16             $incomingBlocks.remove(block)$
17             $blockchain.add(block)$
18     for $block \in hiddenBlocks$ do
19         if $shouldBeReleased(blockchain, block)$ then
20             broadcast($block$)
21     if $curChain \neq blockchain.getCurChain()$ then
22         $curChain \leftarrow blockchain.getCurChain()$
23         $genTime \leftarrow$
             $curTime() + random(Geo(r_i/G))$
24
25 Function broadcast($block$) do
26     for $miner \in otherMiners$ do
27         $block.incomingTime \leftarrow$
             $curTime() + random(Geo(medianTime))$
28         $miner.incomingBlocks.add(block)$
```

VI. Empirical Study on Selfish Mining

We evaluate the improved selfish mining strategy (ISM) and study the following three research questions.

RQ1: *How can multiple miners affect selfish mining?*

RQ2: *How can propagation delay affect selfish mining?*

RQ3: *How can orphan rate affect selfish mining?*

We choose different settings for the block generation time and propagation delay, and describe the performance measurement of selfish mining (Section VI-A). To answer RQ1, we evaluate the performance of ISM on the simulation systems with multiple honest miners and selfish miners. (Section VI-B). To answer RQ2, we evaluate the performance of ISM on the simulation systems with different propagation delays (Section VI-C). To answer RQ3, we analyze the impact of orphan rate on ISM (Section VI-D).

TABLE IV
Relative propagation delay.

	Bitcoin	Other settings	
propagation delay (second)	12.6	15	6
block interval (second)	600	300	60
ratio	1/48	1/20	1/10

A. Experimental Setup

The blockchain simulation algorithm (Algorithm 1) involves three parameters, i.e., propagation delay[1], block interval and miners' mining power ratios. In our experiment, we use the *relative propagation delay*, i.e., the ratio of propagation delay to block interval, to replace these two parameters, which denotes the relative synchronization speed among miners.

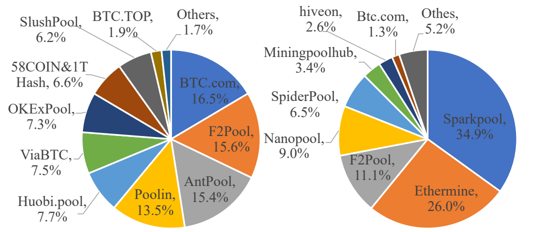
Relative propagation delay. Table IV shows the setting for relative propagation delay (RPD) in our simulation system. In the Bitcoin simulation system, we set the block interval to 600 seconds as that in the real system [1], and the propagation delay to 12.6 seconds based on the experimental result [13]. Thus, the RPD of Bitcoin simulation system is 1/48. To improve throughput, recent blockchain systems tend to reduce the block interval and increase the block size. For example, Litecoin [20] reduces its block interval to 150 seconds, and Dogecoin [21] reduces to 60 seconds. Besides, multiple Bitcoin's fork blockchains, e.g., Bitcoin Cash [22], Bitcoin SV [23] increases their block size, which also results in a larger propagation delay [13]. Therefore, we pay more attention to the blockchain system with a larger RPD, i.e., 1/20 and 1/10.

Since the selected ratio parameters come from existing studies [13] and the trend of blockchain systems [24], [25]. Therefore, these parameters can reflect the real-world situations. The propagation delay may change dynamically with changes in block size and network environment. Our experiment does not take the dynamic change of the propagation delay into consideration. We believe that the influence of its dynamic change can be negligible to our findings.

Mining power distribution. To understand the mining power distribution, we investigate two popular PoW-based blockchain systems, i.e., Bitcoin and Ethereum [14]. As Fig. 6 shows, the top 10 mining pools of Bitcoin and Ethereum account for 98% and 96% of the total mining power, respectively. The power distribution in Ethereum is more centralized than that in Bitcoin, i.e., the top 5 mining pools of Bitcoin and Ethereum account for 69% and 88% of the total mining power.

In the centralized mining pool, a pool manager assigns tasks to miners, and miners submit their solutions. The miners stay connected with the pool manager through specifically-designed protocols, e.g., Stratum [26], which is much faster than the Gossip protocol in the blockchain network. Therefore, we regard a mining pool as a virtual miner. To reflect real-world blockchain systems, we regard 10 miners (mining pools), as the general situation, and 5 or 20 miners as the more centralized or decentralized situation. Without loss of

[1]In the following, propagation delay refers to the meantime of propagation delay.



(a) Bitcoin power distribution (b) Ethereum power distribution

Fig. 6. Mining power distribution in Bitcoin and Ethereum.

generality, we also consider 100 miners. For clarity, the selfish miner is denoted as *Alice* who owns *Alice* α fraction of the total mining power, ranging from 1% to 50% [2]. And the remaining miners equally share the rest mining power.

Experimental measurement. In each simulation, all miners generate $totalBlocks$ blocks, and finally $validBlocks$ blocks will be included in the longest chain. Specifically, we set $totalBlocks$ as 1,000 and iterate each simulation 200 times to avoid potential randomness. Thus, the system orphan rate can be calculated as follows.

$$orphanRate = 1 - validBlocks/totalBlocks \tag{6}$$

Suppose that a miner m_i generates $Blocks_i$ blocks, in which $validBlocks_i$ blocks are included in the longest chain. Its mining revenue and individual orphan rate are calculated as follows.

$$revenue_{m_i} = validBlocks_i/validBlocks \tag{7}$$

$$orphanRate_{m_i} = 1 - validBlocks_i/Blocks_i \tag{8}$$

Equation (9) shows the selfish miner's profitability, which is the revenue difference between selfish mining and honest mining. We use *profit threshold* to denote the minimum mining power required for a miner to make its profitability positive.

$$profit_{ISM} = revenue_{SM} - revenue_{HM} \tag{9}$$

B. RQ1: How can multiple miners affect selfish mining?

In this experiment, we first change the number of honest miners to observe its impact on honest mining and selfish mining. Then, we change the number of selfish miners to observe its impact on selfish mining.

Alice's revenue in the honest mode. Fig. 7(a) shows Alice's honest mining revenue in the blockchain system with the same RPD (1/10) but different honest miners (5, 10, 20, 100). More miners means a more decentralized system. Compared with its mining power ratio (black line), Alice earns more than its power ratio with sufficient mining power. We call the difference between the honest mining revenue and the power ratio as miner's inherent advantage. Fig. 7(b) shows Alice's inherent advantage, which increases with more miners. For example, with 45% of mining power, Alice's mining revenue is 46.50%, 46.83%, 47.20% and 47.30% with 5, 10, 20 and 100 miners, respectively. The revenue is impressive considering the market capitalization of blockchain systems. At the time of writing, a 1% increase in mining revenue means an extra 297,000 USD per day in Bitcoin system. As the number of miners increases, the number of branches also increases. Hence, the strong miner can gain more as it can extends its branch faster.

[2]When α exceeds 50%, the miner can subvert the blockchain history.

Finding 1: *A miner with sufficient mining power has an inherent advantage in the honest mode, especially in a decentralized environment.*

Implication 1. The inherent advantage partially explains the evolution of mining power centralization in the PoW-based blockchain systems [14]. Joining a large pool, miners can obtain a higher revenue. Therefore, the mining power centralization is inevitable without external supervision.

Alice's revenue in the selfish mode. Fig. 7(c) shows Alice's selfish mining revenue in the system with the same RPD (1/10) but different honest miners (5 and 100). It shows that Alice's selfish mining revenue increases and the profit threshold decreases in the more decentralized environment. For example, its profit threshold is 31% and 29% with 5 and 100 miners, respectively.

We notice that the observation is different from the existing study [8], which shows that the number of honest miners has no impact on the performance of selfish mining. The reason is that the theoretical analysis taken by [8] ignores the possibility of branches among these honest miners. However, the possibility can not be ignored when there are a large number of honest miners and propagation delays.

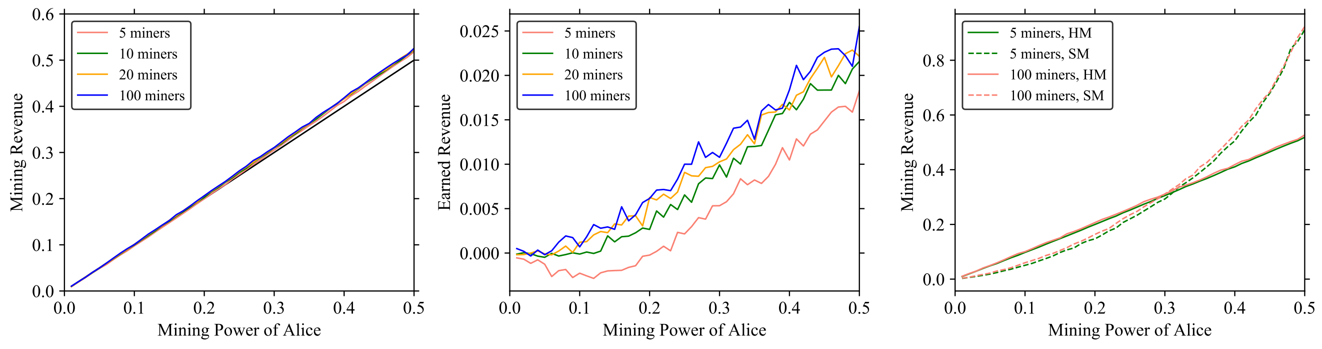
Finding 2: *Selfish mining performs better in the decentralized environment with more honest miners.*

Comparison of honest mining and selfish mining. In selfish mining, the selfish miner is more likely to succeed with a longer private chain. Intuitively, this can cause that multiple consecutive blocks in the longest chain are generated by the same miner. Fig. 8 compares the distribution of consecutive blocks in different mining modes when Alice owns 35% mining power. Compared with honest mining, Alice's probability of generating more than three consecutive blocks increases, which means it holds a longer private chain more often.

Finding 3: *Selfish mining results in more consecutive blocks generated by the selfish miner.*

Implication 2. The finding 3 indicates that we can combat selfish mining through adjusting the mining reward mechanism, e.g., when the same miner generates multiple blocks continuously, the block reward decreases.

Equation (10) is a simple improved mining reward mechanism, in which $blockReward$ denotes the fixed reward for the main chain block, and $Conti$ denotes the number of consecutive blocks. The preliminary analysis shows that the



(a) Honest mining revenue (b) Honest mining earned revenue (c) Selfish mining revenue

Fig. 7. Alice's mining revenue with different honest miners.

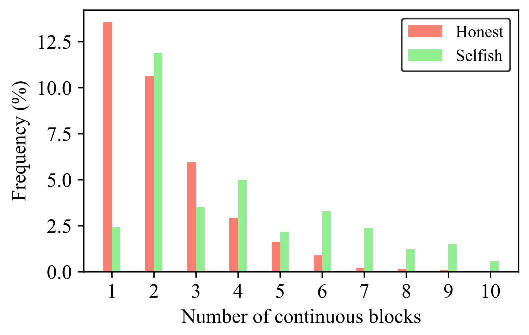


Fig. 8. The distribution of continuous blocks in different mining modes with the same RPD (1/10) and miners (10).

mechanism can effectively reduce the selfish miner's revenue. Under the same condition as in Finding 3, Alice's revenue decreases by 14% in the selfish mode. Even if a miner has nearly 50% of mining power, it cannot benefit from selfish mining. The mechanism can have some impact on the honest miners' absolute revenue, e.g., Alice's honest revenue decreases by 1.5%. However, the honest miners' revenue share increases without selfish mining. The implementation of this new mechanism is technically feasible since miners' addresses are available in the blockchain.

$$\begin{cases} blockReward(Conti = 1, ..., 4) \\ \dfrac{blockReward}{Conti - 2}(Conti \geq 5) \end{cases} \tag{10}$$

Alice's revenue in the selfish mode with multiple selfish miners. We change the number of selfish miners from 0 to 4 in the 5-miner system. Specifically, when there are M selfish miners, the miners from $miner_1$ to $miner_M$ act selfish. We exclude the situation of 5 selfish miners because selfish mining cannot work without honest miners. As mentioned before, $miner_1$, i.e., Alice, owns α fraction of mining power, ranging from 1% to 50%, and other miners equally share the rest power. Therefore, we compute the mining revenue of Alice, and the unit mining revenue, i.e., mining revenue divided by mining power, of $miner_2$ and $miner_5$.

Fig. 9(a) shows Alice's mining revenue, which raises significantly with more selfish miners. For example, when there are 1, 2, 3 and 4 selfish miners, its revenue is 29.55%, 33.65%, 39.88% and 58.02% with 30% of mining power.

Fig. 9(b), (c) shows the unit mining revenue of $miner_2$ and $miner_5$. The line with symbol star(*) means the miner acts honest in this case. It shows that both the small honest miner and small selfish miner lose their revenue when there is a large selfish miner with more mining power. For example, when Alice owns 30% of mining power, i.e., $miner_2$ and $miner_5$ own 17.5% of mining power, $miner_2$' unit mining revenue is 0.76, 0.77 and 0.60, and $miner_5$' unit revenue is 1.01, 0.94 and 0.60 corresponding to 2, 3 and 4 selfish miners. Besides, it also shows the small selfish miner ($miner_2$) loses more than the small honest miner($miner_5$). The observation indicates that the smaller miner's best choice is to be honest.

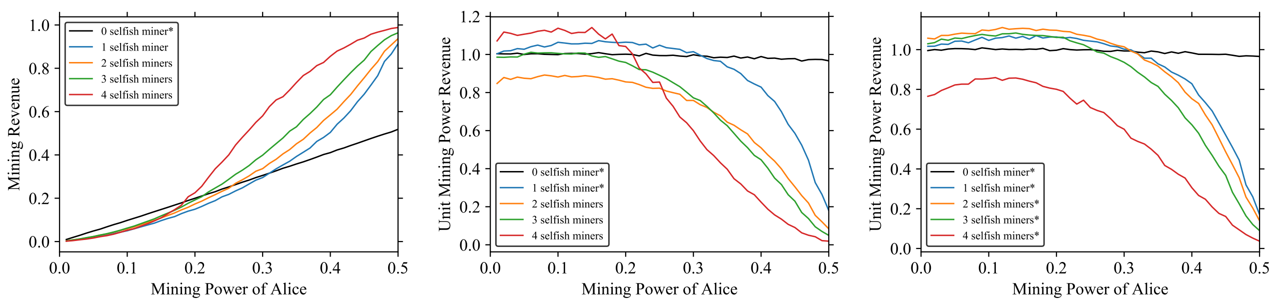
Finding 4: *When there are multiple selfish miners, the large selfish miner with more mining power can benefit from selfish mining, while other smaller miners cannot.*

C. RQ2: How can propagation delay affect selfish mining?

Alice's revenue in the honest mode. Fig. 10(a) shows Alice's honest mining revenue in the 10-miner system with different relative propagation delays (RPDs) and Fig. 10(b) shows its earned revenue. We can see that Alice obtains the similar inherent mining advantage, which becomes greater with a larger RPD. For example, with 45% of mining power, Alice's mining revenue is 45.37%, 46.00% and 46.83% corresponding to the RPD of 1/48, 1/20 and 1/10. A larger delay leads to a greater inconsistency among miners' blockchain views, which benefits the strong miner who can extends its blockchain faster. Therefore, it indicates that a faster communication mechanism contributes to the fairness of the mining process.

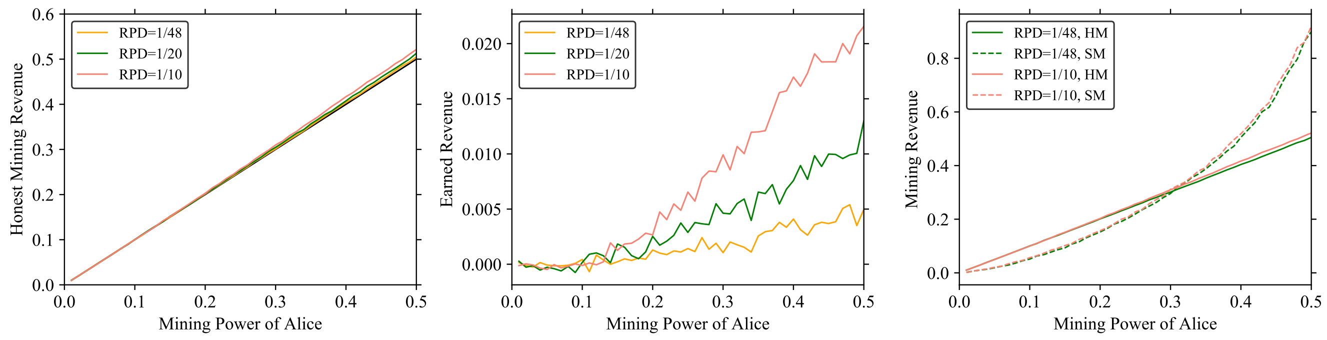
Finding 5: *A miner with sufficient mining power has an inherent advantage in honest mining, especially in the system with a larger propagation delay.*

Alice's revenue in the selfish mode. Fig. 10(c) shows Alice's selfish mining revenue in the 10-miner system with different RPDs, i.e., 1/48 and 1/10. We can see that Alice's selfish mining revenue also increases with a larger RPD. For



(a) Mining revenue of Alice (b) Unit mining power revenue of miner2 (c) Unit mining power revenue of miner5

Fig. 9. Miners' mining revenue in the blockchain system with multiple selfish miners. Star(*) denotes the miner acts honest.



(a) Honest mining revenue (b) Honest mining earned revenue (c) Selfish mining revenue

Fig. 10. Alice's mining revenue with different relative propagation delays (RPDs).

example, with 30% of mining power, Alice's selfish mining revenue is 29.60% and 31.12% respectively.

Finding 6: *Selfish mining performs better in the system with a larger propagation delay.*

D. RQ3: How can orphan rate affect selfish mining?

Fig. 11(a) shows the system orphan rates of different blockchain systems without selfish mining, which increase with a larger RPD and more miners. Since the orphan rate measures the number of discarded blocks, it reflects the consensus efficiency of the blockchain system. More miners or a larger RPD increases the cost of reaching consensus, thus increasing the orphan rate.

Fig. 11(b) shows the system and miners' orphan rates under honest mining with different RPDs, which explains Alice's inherent advantage. With more mining power, Alice's orphan rate decreases, while other miners' orphan rates increase. The reason is that, with the propagation delay, when the blockchain forks, the longest rule favors the larger miner who can extend its blockchain faster. With more frequent forks, the larger miner's inherent advantage becomes more obvious.

Finding 7: *The large miner's inherent advantage comes from its lower orphan rate.*

Fig. 11(c) shows the profit threshold of selfish mining in different blockchain systems. The X-axis represents the system's orphan rate without selfish mining, and Y-axis represents the selfish miner's profit threshold under selfish mining. With a larger orphan rate, selfish mining performs better and the profit threshold decreases, e.g., in the 10-miner system, the profit threshold decreases from 31% to 21% when the orphan rate increases from 1.67% to 40.37%. This indicates selfish mining performs better in a slow synchronization system.

Finding 8: *Profit threshold of selfish mining decreases in the system with a higher orphan rate.*

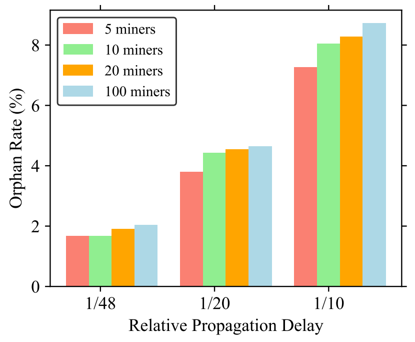
We also investigate how the orphan rate changes with selfish mining. Fig. 12 shows the orphan rates under different mining strategies. In selfish mining, as Alice's mining power increases, the system and other honest miners' orphan rates raises significantly, while Alice's orphan rate decreases. With 40% of mining power, Alice's orphan rate is 12.23%, while other honest miners' increase to 41.76%.

Finding 9: *Selfish mining can have a great impact on the system and miners' orphan rates.*

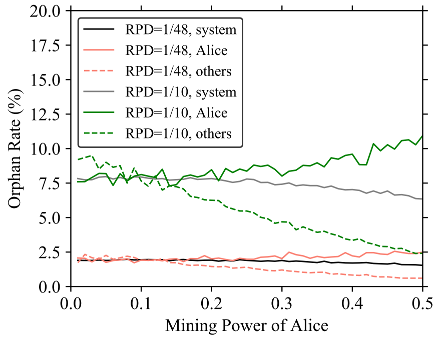
Implication 3. This finding indicates that selfish mining can be detected by monitoring the system orphan rate, and the selfish miner can also be identified by its orphan rate.

VII. Conclusion

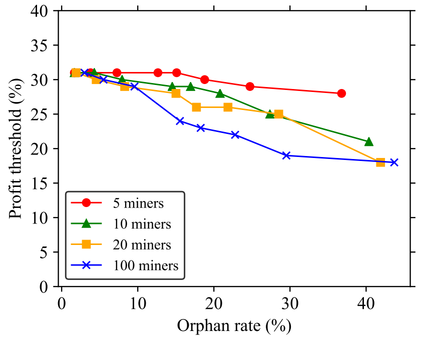
Blockchain serves as an immutable ledger, and has been used to build trusts among participants in distributed systems.



(a) System Orphan rate of different blockchain systems



(b) System and miners' orphan rate under honest mining



(c) Profit threshold in different blockchain systems

Fig. 11. The orphan rates of different blockchain systems.

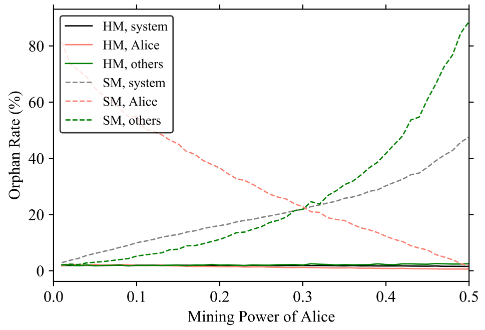


Fig. 12. System and miners' orphan rates in different mining modes with the same RPD (1/48) and miners (10).

Selfish mining can affect the fairness of blockchain systems. In this paper, we propose a new selfish mining strategy that can work in the real-world blockchain scenarios. We then evaluate the new strategy based on the simulation system. The empirical results show many interesting findings, which can be used to combat selfish mining.

Acknowledgement

This work is supported by the National Key Research and Development Program of China under Grant No. 2020YFC1523203, the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA20080200) and Blockchain Technology and Application Joint Laboratory, Guiyang Academy of Information Technology (Institute of Software Chinese Academy of Sciences Guiyang Branch).

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer online cash system," Tech. Rep., 2008.
[2] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography*, 2014.
[3] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *Proceedings of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 305–320.
[4] K. A. Negy, P. Rizun, and E. G. Sirer, "Selfish mining re-examined," 2020.
[5] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proceedings of International Conference on Financial Cryptography and Data Security*, 2016, pp. 515–532.
[6] H. Liu, N. Ruan, R. Du, and W. Jia, "On the strategy and behavior of bitcoin mining with n-attackers," in *Proceedings of Asia Conference on Computer and Communications Security (CCS)*, 2018, pp. 357–368.
[7] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong, "A deep dive into blockchain selfish mining," in *Proceedings of IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.
[8] D. Chang, "Revenue generation strategy through selfish mining focusing multiple pools of honest miners," Ph.D. dissertation, Indraprastha Institute of Information Technology New Delhi, 2019.
[9] J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," *Performance Evaluation*, vol. 104, pp. 23–41, 2016.
[10] J. Niu and C. Feng, "Selfish mining in Ethereum," *Proceedings of IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1306–1316, 2019.
[11] F. Ritz and A. Zugenmaier, "The impact of uncle rewards on selfish mining in Ethereum," *Proceedings of IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 50–57, 2018.
[12] A. M. Antonopoulos, *Mastering bitcoin: Programming the open blockchain*, 2017.
[13] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proceedings of IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2013, pp. 1–10.
[14] (2020) Mining power distribution of pow-based blockchain systems. [Online]. Available: https://blockchair.com/bitcoin/charts/hashrate-distribution
[15] (2020) Bitcoin orphan blocks. [Online]. Available: https://bitcoin-chain.com/block_explorer/orphaned
[16] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proceedings of ACM conference on Computer and Communications Security (CCS)*, 2012, pp. 906–917.
[17] (2020) Bitcoin difficulty. [Online]. Available: https://en.bitcoin.it/wiki/Difficulty
[18] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, "A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–32, 2020.
[19] M. Alharby and A. van Moorsel, "Blocksim: A simulation framework for blockchain systems," *SIGMETRICS Perform. Evaluation Rev.*, vol. 46, pp. 135–138, 2019.
[20] (2020) Litecoin block generation. [Online]. Available: https://bitcointalk.org/index.php?topic=47417.0
[21] (2020) Dogecoin block generation. [Online]. Available: https://github.com/dogecoin/dogecoin
[22] (2020) Bitcoin cash. [Online]. Available: https://en.wikipedia.org/wiki/Bitcoin_Cash
[23] (2020) Bitcoin sv. [Online]. Available: https://bitcoinsv.io/2018/10/21/bitcoin-sv-version-0-1-goes-live/
[24] 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 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 3–16.
[25] (2020) Block size limit controversy. [Online]. Available: https://en.bitcoin.it/wiki/Block_size_limit_controversy
[26] (2020) Stratum mining protocol. [Online]. Available: https://en.bitcoinwiki.org/wiki/Stratum_mining_protocol