Evaluating Fork after Withholding (FAW) Attack in Bitcoin

Runkai Yang
18112049@bjtu.edu.cn
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation
Beijing Jiaotong University
Beijing, China

Xiaolin Chang
xlchang@bjtu.edu.cn
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation
Beijing Jiaotong University
Beijing, China

Jelena Mišić
jmisic@ryerson.ca
Department of Computer Science
Ryerson University
Toronto, Canada

Vojislav Mišić
vmisic@ryerson.ca
Department of Computer Science
Ryerson University
Toronto, Canada

Haoran Zhu
21112051 @bjtu.edu.cn
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation
Beijing Jiaotong University
Beijing, China

ABSTRACT

Fork after withholding (FAW) attack is an easy-to-conduct attack in the Bitcoin system and it is hard to be detected than some attacks like selfish mining and selfholding attacks. The previous studies about FAW attack made some strong assumptions, such as no propagation delay in the network.

This paper aims to quantitatively examine the profitability of FAW attack in Bitcoin system with block propagation delay. We first establish a novel analytic model, which can analyze FAW attack in the Bitcoin system. Then we apply the model to design metric formulas for the Bitcoin system. These formulas can be used to evaluate the miner profitability (in terms of miner reward) and the impact of FAW attack on system throughput (in terms of transactions per second). We make a comparison of FAW attack and other attacks (including selfish mining and selfholding attacks). Experimental results reveal that FAW adversaries can get more rewards in the network with propagation delay than without delay. The results of the comparison of selfish mining and FAW attacks show that adversaries with large computational power can conduct selfish mining or selfholding attack to get more rewards, but they can conduct FAW attack to profit more when their computational power is small. Our work can be used to analyze Bitcoin-like blockchain systems and help design and evaluate security mechanisms.

CCS CONCEPTS

•Security and privacy ~ Systems security ~ Distributed systems security •Theory of computation ~ Models of computation ~ Probabilistic computation

KEYWORDS

Bitcoin, FAW attack, Quantitative analysis, Selfish mining, Selfholding attack

CF '21, May 17–19, 2022, Torino, Italy

ACM ISBN 978-1-4503-9338-6/22/05…$15.00
https://doi.org/10.1145/3528416.3530248

ACM Reference format:

Runkai Yang, Xiaolin Chang, Jelena Mišić, Vojislav Mišić and Haoran Zhu. 2022. Evaluating Fork after Withholding (FAW) Attack in Bitcoin. In *Proceedings of the 19th ACM International Conference on Computing Frontiers (CF'22)*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3528416.3530248

1. INTRODUCTION

Bitcoin [1] is the most representative and valuable Proof-of-Work (PoW)-based blockchain system, where miners create blocks to get rewards. Bitcoin uses a reward mechanism to impel miners to take effort to generate valid blocks and record transactions in the blockchain. Block propagation delay in the Bitcoin network can cause forks, called unintentional forks in this paper. Forks can also occur due to malicious miners who launch attacks such as double-spending [2], selfish mining [3], and fork after withholding (FAW) [4]. This type of fork is named intentional. Once a fork occurs, miners usually choose a branch for mining by using the longest-chain protocol, which selects the longest branch as the main chain (namely, the branch with the most blocks). There are two types of blocks, regular blocks, which are in the main chain, and stale blocks, which are not in the main chain.

Bitcoin is subject to various attacks, among which FAW attack is easier to be conducted than the two classic attacks, double-spending and selfish mining attacks. The reason is that FAW attackers with any computational power can earn an unfair profit (double-spending attack needs more than half of total computational power and selfish mining needs about 25%) [4]. Compared to these two classic attacks in Bitcoin, FAW attack is hard to be detected [4]. Thus, it is necessary to quantitatively analyze FAW attack and evaluate the profitability of attackers by computing their rewards because attackers conduct FAW attack to get more rewards. In addition, FAW attack can lower the throughput of the system, which is a collateral effect of the main goal of making more profit from malicious behavior in mining. The prior analytical modeling works [4][5] on FAW attack assumed there was no propagation delay. Additionally, the prior analytical models of the other Bitcoin attacks like selfish mining like [3] are hard, if not impossible, to be used to analyze FAW attack, because the adversary behaviors of different types of

attacks are different.

This paper analyzes FAW attack by developing a novel stochastic model. The differences of our work from [4][5] include: ① We consider more features of FAW attack in the Bitcoin system, like the occurrence of unintentional forks and the situation where pool mines blocks on its own block (if it has) when forks occur. ② We consider the effort of infiltration miners (defined in Table I) and derive the relationship between $p_{VO,V}$ and other system parameters (see Eq. (1)). ③ We evaluate system throughput in terms of transactions per second (TPS).

There are some similar attacks with FAW attack, such as selfholding and selfish mining attacks. Note that FAW and these attacks have several common features, including: ① They both attack against Bitcoin. ② Some or all of the attackers withhold blocks and publish them maliciously. ③ Attackers behave maliciously to get more rewards. These motivate us to make a quantitative comparison of them.

This paper makes the following contributions.

- We design a Markov model to describe the blockchain dynamic evolution in the Bitcoin system under FAW attack. The model enables the computation of miner rewards. To the best of our knowledge, we are the first to model FAW attack in a Bitcoin network with unintentional forks.
- We propose formulas for computing the profitability of different types of miners (in terms of their rewards) in the Bitcoin system. Moreover, we compute the Bitcoin system throughput under FAW attack. To the best of our knowledge, we are the first to derive formulas to calculate these metrics.
- By conducting experiments, we investigate the impact of FAW attack on Bitcoin. Moreover, we compare FAW with some similar attacks, and evaluate the impact of FAW attack on Bitcoin throughput. To the best of our knowledge, this is the first time that FAW and these attacks are quantitatively compared.

These quantitative analyses not only enable honest miners to detect whether they are being attacked by observing their reward and the system throughput but also help pool managers design countermeasures, like more secure in-pool reward mechanisms, to prevent FAW attack. We leave the applications for future work.

The organization of the rest paper is as follows. Background and related work can be found in Section 2 and Section 3, respectively. In Section 4, we describe the system we consider, the proposed model and formulas. In Section 5, we give the experiment results. Section 6 concludes the paper.

2. BACKGROUND

In this section, we first present mining blocks in Bitcoin and mining pools. Then, we introduce some classic attacks in Bitcoin systems, including FAW, selfish mining and selfholding attacks.

2.1. PoW and Mining blocks in Bitcoin

In Bitcoin, transactions are recorded in blocks, and all blocks form a chain structure. Each block in Bitcoin has a block header and a block body. The block body contains a set of transactions. The block header includes several fields, like a hash of the previous block, a nonce value, the difficulty value, and so on. This structure ensures that the transactions in Bitcoin are hard to be changed.

Bitcoin uses PoW to achieve the system consensus. PoW consensus is a widely-used consensus in blockchain-based cryptocurrency systems. The principle of PoW consensus is asymmetry: it is difficult to find a PoW solution, but it is easy to verify the solution. The process of finding a PoW solution is compared to golden mining, and the participant in Bitcoin are named miners.

To produce a new block in Bitcoin, miners first fill in all fields in the block header except the nonce value. And then, miners have to find a nonce value to satisfy the mining difficulty. Namely, the hash value of the block header is lower than the difficulty value. This nonce value is named FPoW (namely, Full PoW solution). Namely, once an FPoW is founded by a miner, the miner generates a new block in Bitcoin. The height of a block refers to the number of blocks preceding the block. The mining process is a competition between miners, and a round denotes that blocks are generated at a block height. Once a round ends, the next round starts. In a perfect network without attacks, only one block is generated in a round, but there may be more than one block generated when the network is imperfect or some attackers in the system.

With the increasing number of miners and the development of mining machines, the total computational power in Bitcoin increases rapidly. To keep the average time of generating a block stable (about 10 minutes), the difficulty to find an FPoW is adjusted (by changing the difficulty value), and this mechanism is named the difficulty adjustment mechanism. Some attacks, such as FAW, selfish mining, and selfholding attacks, prolong the block generation time, resulting in mining more easily and making attackers get more mining rewards.

2.2. Mining pools

In recent years, the total computational power of the Bitcoin system has been extremely large [6]. Thus, it may take several years for an individual miner to generate a block. To avoid this situation, miners form pools to mine blocks collaboratively. Besides in-pool miners, a pool manager is in the pool to assign tasks to in-pool miners. Different from mining individually, in-pool miners are asked to find a solution, which is much easier than finding an FPoW, named a Partial PoW (PPoW) solution. In short, generating an FPoW means generating a block, but most PPoWs cannot satisfy the difficulty requirement (no blocks are generated). Usually, in-pool miners are rewarded by the pool manager, and their rewards depend on how many PPoWs they submit.

2.3. Fork after withholding (FAW) attack

FAW attackers consist of two parts. One part of attackers mine blocks in a pool innocently (denoted as malicious pool, MP). The others (denoted as infiltration miners, IMs) infiltrate into an honest pool (denoted as the victim pool, VP) and mining blocks maliciously. Once a PPoW is found, honest in-pool miners submit it to the pool manager. IMs withhold FPoWs (namely, blocks) and only submit them when other honest pools (OPs) generate blocks. However, the manager of a VP does not know the existence of attackers in the pool since IMs submit PPoWs normally. By conducting FAW attack, attackers lower the rewards of VP (due to withholding blocks) and trick honest pools into creating forks (by publishing blocks strategically). Namely, the rate of generating regular blocks is decreased, and then the difficulty of mining blocks is easier. Thus, attackers can get more rewards. TABLE I summarizes the behaviors of honest miners, innocent miners and IMs.

TABLE I. Behaviors of each type of miners

Miner type	Miner behavior
Honest miner	Submit all PPoWs, including FPoWs.
Innocent miner	Form a pool and mines blocks honestly.
Infiltration miner	Submit all PPoWs and do not submit FPoWs until other honest pools generate a block.

2.4. Selfish mining attack

Selfish mining attackers (namely, selfish miners) always form a pool and centralize computational power to generate blocks. Different from honest mining strategy, in some cases, selfish pool generates a new block but not publishes the block. With attackers adopting selfish mining strategy, honest miners are tricked into mining stale blocks and lose mining rewards. As honest miners produce more stale blocks than usual, the block generation time prolongs, and then the mining difficulty becomes easier. Thus, the selfish pool can get more rewards more easily. The behaviors of the selfish pool are given in TABLE II.

TABLE II. Behaviors of selfish miners

The difference length of branches	If honest miners generate a block	If the selfish pool generates a block
0 (There is no fork)	Mine on the new block	Conceal the block
0 (There is a fork with equal-length branches)	Mine on the new block	Publish the block
1	Publish a block	Conceal the block
2	Publish two blocks	Conceal the block
>2	Publish a block	Conceal the block

2.5. Selfholding attack

Selfholding attack can be regarded as the variant attack of block withholding (BWH) attacks. Thus, we introduce BWH attack first. In BWH attack, some attackers infiltrate into an honest pool and pretend to mine in it. These attackers withhold FPoWs but submit PPoWs to the manager. The rest of attackers form a pool and mine blocks innocently. Namely, the rest of attackers in the pool adopt honest mining strategy. In BWH attack, attackers can get rewards not only from the innocent mining pool but also from infiltration mining. In selfholding attack, some selfholding attackers form a mining pool and mine blocks selfishly. The others infiltrate into victim pools and do not submit FPoWs to the pool manager.

3. Related Work

As the market cap of Bitcoin increases rapidly and the blockchain is widely used, some research has been conducted to study blockchain security [7]-[10]. This section focuses on FAW attack in Bitcoin and some related attacks in Bitcoin.

FAW attack is a way of attacking to get more undeserved rewards. Based on BWH attack [11], FAW attack changes some strategies and makes attackers get about 56% extra reward than BWH attack [4]. Ke *et al*. [5] compared the mining rewards between FAW attack and BWH attack. However, they did not consider unintentional forks and only focused on the mining reward. In fact, the system considered in [4][5] is a special case of our system. Different from [4][5], we also investigate some detailed system features as follows:

① We consider the network delay so that unintentional forks can occur.

② In our system, VP mines blocks on its own block (if it has) when forks occur. This is because in the real world when a fork is created, a pool must mine blocks on its own block in order to make the block be a regular block.

③ The authors of [4][5] ignored the computational power of infiltration miners when they computed the probabilities of MP and VP in generating blocks. Actually, infiltration miners also contribute computational power and create blocks and affect the probabilities. In this paper, when we analyze the rewards of miners, we also consider the effort of IMs in detail.

④ When an intentional fork is created, the authors in [4][5] defined a variable to denote the probability ($p_{VO,V}$) that the block is produced by VP. However, our paper derives the relationship between $p_{VO,V}$ and the other system parameters (see Eq. 1) to investigate FAW attack more effectively.

Recently, some variants of the original FAW attack were proposed in [12][13], and these works assumed that there are no unintentional forks. We leave for future work the extension of our model developed in this paper to study FAW variants. To defend FAW attack and other withholding attacks, researchers proposed that they can be alleviated by modifying the pool reward mechanism [14]-[16]. Our work can help design and evaluate the countermeasures to resist FAW attack.

Selfish mining is another type of attack against the Bitcoin system to get more rewards. Researchers have evaluated selfish mining using Markov models or simulation experiments [3],[17]-[22]. Selfholding attack was proposed by [23], and Yang *et al.* [24] studied the selfholding attack in an imperfect network where exists unintentional forks due to block propagation delay. But these existing models about selfish mining and selfholding attacks cannot be used to analyze FAW attack due to that the behaviors of these attackers and FAW attackers are different. For example, when a system is under FAW attack, a part of attackers make VP creates forks intentionally. This does not occur in the blockchain under selfish mining. It is noticed that these two attacks share some common features with FAW attack, and we compare these three attacks in this paper.

4. System Description and System Model

This section first presents the Bitcoin system to be studied and then describes the developed model and formulas. TABLE III gives the notations to be used in the following.

4.1. System description

In the system to be studied, all miners are divided into honest and malicious miners (attackers). Each miner works in mining pools. There are three types of pools, an MP, a VP and a large number of OPs. Two types of miners exist in a VP, honest miners and IMs. We use Fig. 1 to illustrate the pools and miners in the system, where there is one MP, one VP and three OPs. Each pool contains multiple miners.

Although an MP is created by attackers, it employs the honest mining strategy. Thus, all pools employ the honest mining strategy, and the details are as follows: ① Every pool chooses the longest branch to mine blocks. ② The pool mines blocks on its own block if it finds forking (with equal-length branches) occurrence and one of the branches is created by itself. Otherwise, it selects a branch randomly. ③ Pools can create unintentional forks on account of the bad network connection delay and/or block propagation.

The computational power of a pool in the Bitcoin system is a key metric of the pool. We define the total computational power

of attackers is C_A. τ is defined to denote the fraction of the computational power for infiltration mining in attackers' total power. Namely, $C_{IM} = C_A\tau$ and $C_{MP} = C_A(1-\tau)$. The computational power of honest miners in VP is C_{VP}. Since attackers join this pool and do infiltration mining, the total computational power of the VP is $C_{VP} + C_{IM}$. The computational power of OPs is C_O. Here, $C_T = C_A + C_{VP} + C_O$.

We make the following system assumptions.

- At most two blocks can be created simultaneously by honest pools unintentionally. It is because the probability of the occurrence of a three-branch unintentional fork can be ignored in Bitcoin [25].
- All miners are in-pool miners. It is because that few blocks are generated by individual miners in Bitcoin.
- The total computational power remains unchanged.
- There are no other attacks except FAW attack in the system and only one group of attackers.
- Once blocks are generated by the VP or OPs, attackers can find the blocks at once. Thus, MP does not create blocks with other pools at the same time.
- Although IMs can withhold more than one block, they only publish one block in order to avoid being detected.

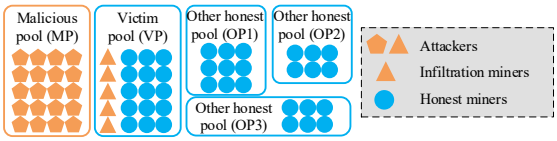


Fig. 1.Types of pools and miners in the system.

TABLE III. Definition of notations

Notation	Definition
C_T	Total computational power in the system.
C_{MP}, C_{IM}	Computational power of MP and IMs, respectively.
C_A	Computational power of all attackers.
C_{VP}, C_{OP}	Computational power of honest miners in the VP and OPs, respectively.
F	F=1 denotes IMs withhold blocks, and F=0 denotes IMs do not withhold.
n	The number of chains with the same length.
r_1, r_2	The rates of all miners in honest pools generating a block and two blocks in a block interval, respectively.
r_M , r_{VP}, r_{IM}, r_{OP}	The rates of MP, VP (honest miners), IMs and OPs generating a block, respectively.
r_{VI}, r_{VO}, r_{IO}, r_{OO}	The rates of generating two blocks at the same time by VP and IMs (VI), VP and OPs (VO), IMs and OPs (IO), and two OPs (OO), respectively.
τ	The fraction of the computational power of IMs in total attackers' power.
θ	The probability that pools create an unintentional fork.
$\pi_{(F,n)}$	The steady-state probability of state (F,n).
$r_M{}^r$,$r_V{}^r$, $r_O{}^r$	The rewards of MP, VP and OPs, respectively.
r_A,r_{VP}	The rewards of attackers (A) and honest miners in VP (VP), respectively.
r_{total}	The total rewards of all miners.
rel_r_A, rel_r_V, rel_r_O	The relative reward of attackers, honest miners in VP and honest miners in OPs, respectively.
TPS	Transactions per second.

4.2. The system model

Both propagation delay and FAW attackers can cause forks in the system. We define n to denote the number of branches in Bitcoin. If IMs withhold blocks, intentional forks can occur after OPs generate blocks, and we use F to denote whether IMs withhold blocks. Thus, (F,n) is defined to denote the state of the system. According to the assumptions, $F \in \{0,1\}$ and $n \in \{1,2,3\}$. Thus, there are six states in total. Namely, $(0,1)$, $(0,2)$, $(0,3)$, $(1,1)$, $(1,2)$ and $(1,3)$. Note that, $n=3$ denotes that there are three branches. Two branches are created by OPs unintentionally, and one branch is created by IMs. We use $(0,2)$ and $(1,3)$ as examples to further explain the system states in Fig. 2. $(0,2)$ denotes that the number of branches is two, and IMs do not withhold blocks. $(1,3)$ denotes that there are three branches and IMs withhold blocks. It is assumed that the block generation time is exponentially distributed as in [3].

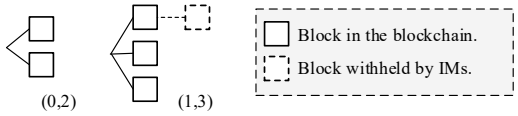


Fig. 2. Illustration of system states (0,2) and (1,3).

For a pool, the rate of the creation of a block is proportional to its computational power. Without loss of generality, we normalize the block generation rate as 1. Similar to the total system computational power. The rate of MP generating a block is $r_M = (1-\tau)C_A$. Let r_2 denote the rate that an unintentional fork occurs, and $r_2 = \theta$. The total rate of miners in VP and OPs generating a block is $r_1 = 1-2\theta-r_M$. r_i, r_V and r_O are proportional to the computational power of IMs, VP and OPs. The assumptions suggest that the rate of two blocks being found at the same time can be computed by $r_2 \cdot P_{2-x}$. Here x denotes miner type and $x \in \{VO, IO, OO, VI\}$. For example, P_{2-VO} denotes the probability that the two blocks are generated at the same time by honest miners in VP and OPs, and $P_{2-VO} = C_{VP} \cdot C_{OP} / (1-C_{VP}) + C_{OP} \cdot C_{VP} / (1-C_{OP})$. By solving the global balance equations based on the transition diagram, we can get the steady-state probabilities of the six states.

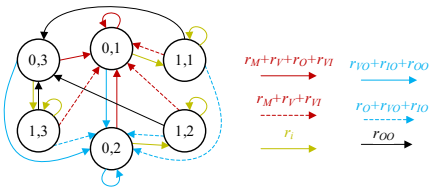


Fig. 3. State-transition diagram.

4.3. Metric formulas

As mentioned before, FAW attack leads to Bitcoin mining difficulty adjustment, and the relative reward represents the rewards that miners can get after mining difficulty adjustment. In this section, we first calculate the relative reward of each type of miner. Then, we evaluate system throughput by computing TPS of Bitcoin. The probabilities that MP, IMs, VP and OPs generate a block are denoted by p_M, p_I, p_V and p_O, respectively. The probabilities of generating two blocks at the same time are p_{VI}

(the two blocks are found by VP and IMs), p_{VO} (the two blocks are found by VP and OPs), p_{IO} (the two blocks are found by OPs and IMs) and p_{OO} (the two blocks are found by OPs). Similar to calculating the rate of generating block, we can get these probabilities.

4.3.1. Relative reward of miners

Relative reward is defined as the proportion of the miner rewards in all miner rewards, and it represents the profitability of miners. This section first calculates the rewards of different types of pools. Then, we compute the rewards of attackers and honest miners. Last, we can get the relative reward of miners. Before calculating the rewards, we first propose *Lemma 1*, which enables the computation of rewards later.

Lemma 1. Assume there are two blocks at the same height, which are created by VP and OPs, respectively. $p_{VO,V}$ denotes the probability that the block created by VP is a regular block, and $p_{VO,O}$ denotes the probability that the block created by OPs is a regular block. Similarly, when there is a fork with three blocks (two of them are created by OPs, and the rest is created by VP), $p_{VOO,V}$ and $p_{VOO,O}$ are defined to denote the probabilities that the blocks created by VP or OPs are regular blocks, respectively. There, $p_{VO,V}$ and $p_{VOO,V}$ can be computed by Eqs. (1) and (2). In addition, $p_{VO,O}=1-p_{VO,V}$ and $p_{VOO,O}=1-p_{VOO,V}$.

$$\begin{aligned} p_{VO,V} &= \frac{p_M+p_O+p_{IO}+p_{VO}}{2}+p_V \\ &+\frac{p_{IO}+p_{VO}+p_{OO}}{2}p_{VO,V}+p_{VI}+\frac{p_{OO}}{4} \\ &+p_I\left(p_O\frac{1+p_{VO,V}}{2}+p_{OO}\frac{1+p_{VOO,V}}{2}\right) \end{aligned} \tag{1}$$

$$\begin{aligned} p_{VOO,V} &= \frac{p_M+p_O+p_{IO}+p_{VO}}{3}+p_V+p_{VI} \\ &+\frac{p_{OO}}{9}+\left(\frac{p_{IO}+p_{VO}}{3}+\frac{4p_{OO}}{9}\right)p_{VO,V} \\ &+p_I\left(p_O\frac{1+2p_{VO,V}}{3}+p_{OO}\frac{1+2p_{VOO,V}}{3}\right) \end{aligned} \tag{2}$$

Proposition 1. The rewards of MP (denoted as r_M^r) are p_M.

Proposition 2. The rewards of VP (denoted as r_V^r) are given in Eq. (3).

$$\begin{aligned} r_V^r &= p_V+p_{VI}+\left(p_{VO}+p_{IO}\right)p_{VO,V} \\ &+p_I\left(p_Op_{VO,V}+p_{OO}p_{VOO,V}\right) \end{aligned} \tag{3}$$

Proposition 3. Eq. (4) can calculate the rewards of OPs (denoted as r_O^r).

$$\begin{aligned} r_O^r &= p_O\left(\sum_{n=1}^{3}\left(\pi_{(0,n)}\right)+\sum_{n=1}^{3}\left(p_{VO,O}\pi_{(1,n)}\right)\right) \\ &+p_{OO}\left(\sum_{n=1}^{3}\left(\pi_{(0,n)}\right)+\sum_{n=1}^{3}\left(p_{VOO,O}\pi_{(1,n)}\right)\right) \\ &+\left(p_{VO}+p_{IO}\right)p_{VO,O} \end{aligned} \tag{4}$$

For attackers, they can get rewards from two parts, MP and IMs. The in-pool rewards for IMs are proportionate to the computational power of IMs. Thus, we can compute the attacker rewards by the formula of $r_M^r+r_V^rC_{IM}/(C_{IM}+C_{VP})$. The rewards of honest miners in VP (denoted by r_{VP}) can be calculated by the formula of $r_V^rC_{VP}/(C_{IM}+C_{VP})$. r_{total} is defined to denote the total rewards of all miners and then we get r_{total} by Eq. (5).

$$r_{total}=r_M^r+r_V^r+r_O^r \tag{5}$$

The relative reward of attackers (denoted by rel_r_A) is r_A/r_{total}. Similarly, the relative reward of honest miners in VP and that of miners in OPs can be calculated.

The relative extra reward is defined in [4], which can be computed by $(rel\_r_A-C_A)/C_A$. The relative extra reward denotes the proportion of extra rewards to the rewards which attackers can get if they conduct honest mining.

4.3.2. Transactions per second

Bitcoin accounts the transactions only in regular blocks. Each block includes a constant number of transactions, and TPS can be evaluated by the regular block generation rate. As one reward means that a regular block is generated, we can calculate the regular block generation rate by the total rewards (by Eq. (5)), namely, $r_M^r+r_V^r+r_O^r$. To clearly observe the impact of FAW attack, we normalize TPS of Bitcoin as 1 when there is no fork or attackers. Thus, the system TPS can be calculated by $r_M^r+r_V^r+r_O^r$.

5. EXPERIMENT RESULTS

In this section, we first show that our model and formulas are approximately accurate in Section 5.1. Note that varying features of Bitcoin like network delay, miner (pool) number and block size can all change the unintentional fork probability (abbreviated as fork probability in the rest of the paper). Thus, it is reasonable to use fork probability to denote a scenario. Moreover, fork probability affects the profitability of FAW attackers. Thus, we quantitatively analyze the attacker profitability by varying fork probability in Section 5.2 rather than varying these features. As mentioned before, selfish mining, selfholding and FAW attacks have many similar features, and therefore, we compare them in Section 5.3. FAW attack also lowers system throughput. Thus, we evaluate system throughput in Section 5.4.

By using MAPLE [26], we can get the steady-state probabilities of states and calculate the metrics. VP is set as 20% of total computational power because the largest pool in Bitcoin is about 20% of total computational power [27]. As $\tau\in[0,1]$, we increase τ from 0 to 1 to observe the optimal τ under different parameters to make attackers get the maximized relative reward, and then we show the results when τ is optimal. The common fork probability in real Bitcoin is about 0.5% [28], and we set the forking probability as 0.5%, 1% and 2% to observe the impact of fork probability on FAW attack.

5.1. Verification of the model and formulas

We develop a simulation approach in C++ language. By comparing the relative reward of FAW attackers obtained from numerical and simulation experiments, we verify the proposed model and formulas. In each simulation experiment, a Bitcoin

blockchain with 100 million regular blocks is created. We show the results of numerical and simulation experiment results where fork probabilities are 0.5% and 2%, respectively. Fig. 4 shows the results, where "num" and "sim" denote numerical and simulation results, respectively. "rel_rA", "rel_rV" and "rel_rO" denote the relative reward of attackers, honest miners in VP and honest miners in OPs, respectively. We observe that numerical and simulation results are very close, which can verify that our model and formulas are approximately accurate. Note that the model of Kown *et al.* [4] is a special case of ours by letting $p_{VO,V} = 0.5$ and the fork probability as 0. Our formulas under these settings can produce the same results as in [4] (as shown in Fig. 5), which suggests the approximate accuracy of our work too.

By observing Fig. 4-a) and Fig. 4-b), we can find that: ① With the computational power of attackers increasing, attackers can get more relative reward, and the relative reward of honest miners is lowered. This is because attackers make honest miners create intentional forks, resulting in honest miners generating fewer regular blocks and getting fewer rewards. However, as MP always generates regular blocks (IMs do not submit blocks when MP generates a block. Therefore, there is no intentional fork when MP produces a block), the rewards for MP do not change. ② The relative reward of honest mines in OPs decreases significantly but the relative reward of honest miners in VP decreases little, with the increasing computational power of attackers. This is because that VP is a relatively large pool in our system, and VP always mines on its own branch when there is a fork. Thus, the probability that VP's block becomes a regular block is relatively large when a fork occurs.

5.2. Evaluation of the profitability of FAW attackers

Fig. 6 shows the results, where "FAW-0%", "FAW-0.5%", "FAW-1%" and "FAW-2%" denote the results of the fork probabilities of 0%, 0.5%, 1% and 2%, respectively.

By observing Fig. 6, we find that attackers conducting FAW attack can get more relative reward than adopting honest mining strategy. Attackers can get more relative reward in the Bitcoin network with propagation delay. Additionally, the higher the fork probability, the more the relative reward that FAW attackers can get. This is because with the increasing forking occurrence more computational power of honest miners is used to create stale blocks. Namely, the total rewards decrease. However, MP can always generate regular blocks, and its rewards do not change. Relative reward is the proportion of the miner rewards in the total rewards. Thus, with the increasing fork probability, attackers gain more relative reward.

5.3. Comparison of selfholding, selfish mining and FAW attacks

These three types of attacks are compared from the perspective of relative reward. Fig. 7 shows the results, where "FAW", "selfholding" and "selfish mining" denote the results of FAW, selfholding and selfish mining attacks. There are analytical models about selfish mining attack, among which only the authors in [21] considered the same scenario as in our paper, and we implemented the work of [21] to compute the relative of selfish miners. For the same reason, we reproduce the work in [24] to calculate the relative reward of attackers who conduct selfholding attack. To make a clear comparison, we set the fork probabilities of the systems as 0.5% and 2%.

Different from FAW attack, selfholding attackers and selfish miners cannot get more relative reward without enough computational power (more than about 25% of total computational power). Namely, for the attackers with small computational power they can get unfair relative reward by conducting FAW attack. But for the attackers with large computational power, conducting selfish mining makes them gain more undeserved rewards. With the increasing attackers' computational power (more than 25%), attackers get more relative reward by conduct selfholding or selfish mining attacks than FAW attack. This is because, in selfish mining, all attackers behave maliciously and it can increase relative reward obviously. However, in FAW attack, only a part of attackers behave maliciously, and attackers can get less unfair rewards.

5.4. Impact of FAW attack on Bitcoin system throughput

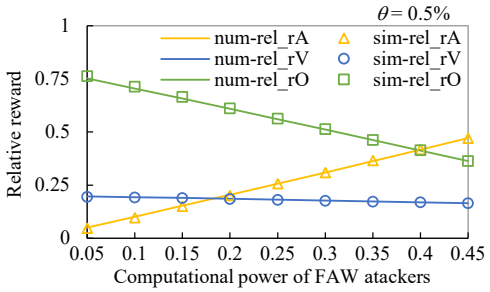
Fig. 8 gives the throughput under different fork probabilities. We find that TPS decreases with the increasing computational power of FAW attackers. This is because that more FAW attackers' computational power means more computational power of IMs, leading to less effective computational power to produce blocks in the Bitcoin system. There are some sharp drops in the figure (for example, when FAW attackers' computational power increases from 0.35 to 0.4 and $\theta = 0.5\%$). The reason is that FAW attackers increase τ (leading to the computational power of IMs increasing) to get the maximum rewards, resulting in less effective computational power and less TPS. When we compare the TPS of the system which is under different attacks, we can find that the FAW attack affects system TPS less than selfholding and selfish mining attacks. This is because that the stale blocks created by FAW attackers are fewer than the other types of attackers. By comparing Fig. 8-a) and Fig. 8-b), we find that fork probability also affects system TPS. When the computational power of attackers is small, fork probability is the main factor affecting system TPS.

6. CONCLUSION

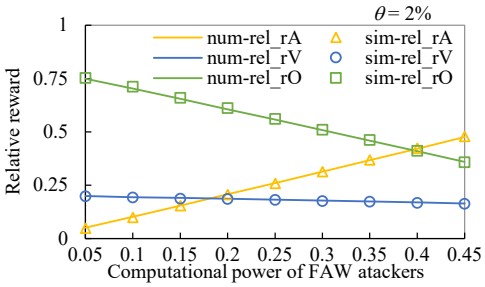
This paper explores an analytical modeling approach to quantitatively study FAW attack in the Bitcoin system. Numerical results show that FAW attackers can get more relative reward in Bitcoin with forks than in a network without forks. We also make a comparison of FAW attack with selfholding and selfish mining attack in terms of attacker revenue and throughput. We find that when adversaries have small computational power, they can conduct FAW attack to benefit more than conducting selfholding or selfish mining attack. But when they have large computational power, they can benefit more by conducting selfholding or selfish mining attack. In terms of blockchain throughput (TPS), conducting selfholding or selfish mining attacks can degrade TPS more than conducting FAW attack.

ACKNOWLEDGMENTS

The work of R. Yang, X. Chang and H. Zhu was supported by Beijing Municipal Natural Science Foundation (No. M22037). The work of J. Mišić and V. Mišić was supported by Natural Science and Engineering Research Council (NSERC) of Canada.



a) Fork probability is 0.5%.



b) Fork probability is 2%.

Fig. 4. Verify the relative reward by simulation experiments.

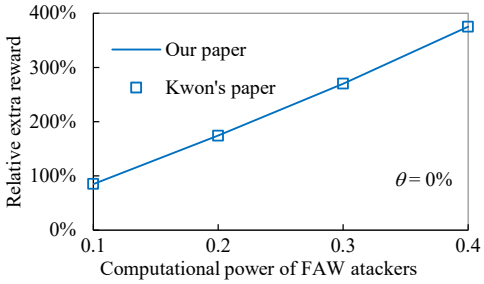


Fig. 5. Verify the model and formulas by comparing with the existing work.

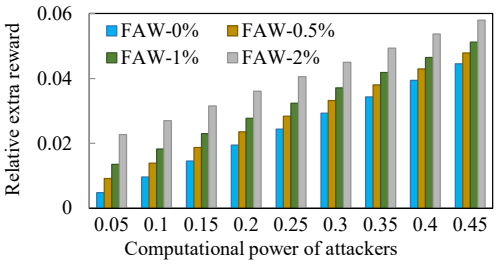
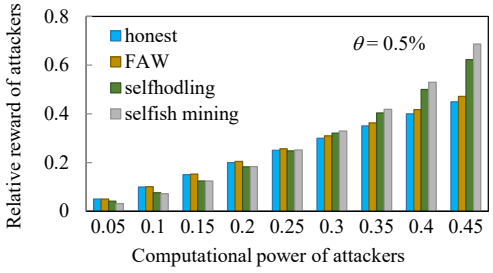
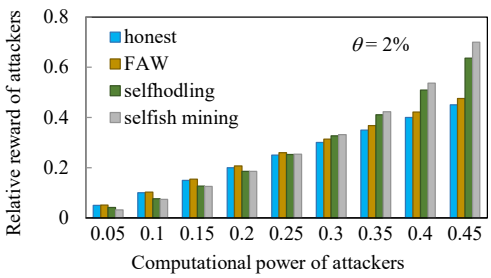


Fig. 6. The relative extra reward of FAW attackers over fork probability.

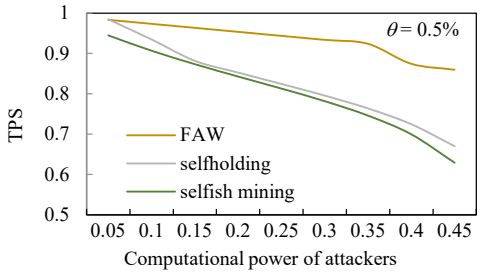


a) Fork probability is 0.5%.

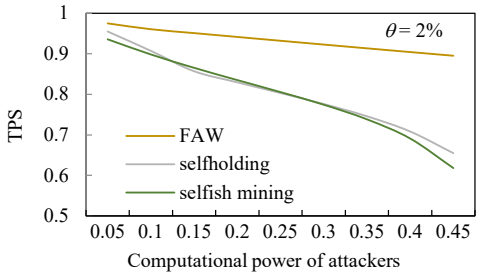


b) Fork probability is 2%.

Fig. 7. Compare with selfholding, selfish mining and FAW attacks.



a) Fork probability is 0.5%.



b) Fork probability is 2%.

Fig. 8. TPS of Bitcoin system under FAW, selfholding and selfish mining attacks.

REFERENCES

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review. 2008 Oct 31:21260.

[2] Karame GO, Androulaki E, Capkun S. Double-spending fast payments in bitcoin. In Proceedings of the 2012 ACM conference on Computer and communications security 2012 Oct 16 (pp. 906-917).

[3] Eyal I, Sirer EG. Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security 2014 Mar 3 (pp. 436-454). Springer, Berlin, Heidelberg.

[4] Kwon Y, Kim D, Son Y, Vasserman E, Kim Y. Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security 2017 Oct 30 (pp. 195-209).

[5] Ke J, Jiang H, Song X, Zhao S, Wang H, Xu Q. Analysis on the block reward of fork after withholding (FAW). In International Conference on Network and System Security 2018 Aug 27 (pp. 16-31). Springer, Cham.

[6] https://tokenview.com/en/minePoolList. Accessed, Aug, 2021.

[7] Zaghloul E, Li T, Mutka MW, Ren J. Bitcoin and blockchain: Security and privacy. IEEE Internet of Things Journal. 2020 Jun 22;7(10):10288-313.

[8] Zhu H, Yang R, Mišić J, Mišić VB, Chang X. How Does FAW Attack Impact an Imperfect PoW Blockchain: A Simulation-based Approach. In 2022 IEEE International Conference on Communications (ICC) 2022 May 15 (pp. 1-6). IEEE.

[9] Saad M, Spaulding J, Njilla L, Kamhoua C, Shetty S, Nyang D, Mohaisen D. Exploring the attack surface of blockchain: A comprehensive survey. IEEE Communications Surveys & Tutorials. 2020 Mar 2;22(3):1977-2008.

[10] Li X, Jiang P, Chen T, Luo X, Wen Q. A survey on the security of blockchain systems. Future Generation Computer Systems. 2020 Jun 1;107:841-53.

[11] Rosenfeld M. Analysis of bitcoin pooled mining reward systems. arXiv preprint arXiv:1112.4980. 2011 Dec 21.

[12] Chang SY, Park Y, Wuthier S, Chen CW. Uncle-block attack: Blockchain mining threat beyond block withholding for rational and uncooperative miners. In International Conference on Applied Cryptography and Network Security 2019 Jun 5 (pp. 241-258). Springer, Cham.

[13] Gao S, Li Z, Peng Z, Xiao B. 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 Nov 6 (pp. 833-850).

[14] Sarker A, Wuthier S, Chang SY. Anti-withholding reward system to secure blockchain mining pools. In 2019 Crypto Valley Conference on Blockchain Technology (CVCBT) 2019 Jun 24 (pp. 43-46). IEEE.

[15] Lee S, Kim S. Countering block withholding attack efficiently. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) 2019 Apr 29 (pp. 330-335). IEEE.

[16] Chang SY, Park Y. Silent timestamping for blockchain mining pool security. In 2019 International Conference on Computing, Networking and Communications (ICNC) 2019 Feb 18 (pp. 1-5). IEEE.

[17] Göbel J, Keeler HP, Krzesinski AE, Taylor PG. Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. Performance Evaluation. 2016 Oct 1;104:23-41.

[18] Carlsten M, Kalodner H, Weinberg SM, Narayanan A. On the instability of bitcoin without the block reward. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security 2016 Oct 24 (pp. 154-167).

[19] Yang R, Chang X, Mišić J, Mišić V.B., Kang H. Quantitative Comparison of Two Chain-Selection Protocols under Selfish Mining Attack. IEEE Transactions on Network and Service Management. (Early Access).

[20] Bai Q, Zhou X, Wang X, Xu Y, Wang X, Kong Q. A deep dive into blockchain selfish mining. In 2019 IEEE International Conference on Communications (ICC) 2019 May 20 (pp. 1-6). IEEE.

[21] Yang R, Chang X, Mišić J, Mišić VB. Assessing blockchain selfish mining in an imperfect network: Honest and selfish miner views. Computers & Security. 2020 Oct 1;97:101956.

[22] Kang H, Chang X, Yang R, Mišić J, Mišić VB. Understanding Selfish Mining in Imperfect Bitcoin and Ethereum Networks with Extended Forks. IEEE Transactions on Network and Service Management. 2021 Apr 15.

[23] Dong X, Wu F, Faree A, Guo D, Shen Y, Ma J. Selfholding: A combined attack model using selfish mining with block withholding attack. Computers & Security. 2019 Nov 1;87:101584.

[24] Yang R, Chang X, Misic J, Misic VB, Kang H. On Selfholding Attack Impact on Imperfect PoW Blockchain Networks. IEEE Transactions on Network Science and Engineering. 2021 Aug 10.

[25] Mišić J, Mišić VB, Chang X. On Ledger Inconsistency Time in Bitcoin's Blockchain Delivery Network. In 2019 IEEE Global Communications Conference (GLOBECOM) 2019 Dec 9 (pp. 1-6). IEEE.

[26] https://www.maplesoft.com/. Assessed, July. 2021.

[27] https://btc.com/stats/pool?pool_mode=year. Accessed, August 2021.

[28] https://www.blockchain.com/charts/n-orphaned-blocks. Access: July, 2021.