A SURVEY: REWARD DISTRIBUTION MECHANISMS AND WITHHOLDING ATTACKS IN BITCOIN POOL MINING

SAIDE ZHU\textsuperscript{1}, WEI LI\textsuperscript{1}, HONG LI\textsuperscript{2}
CHUNQIANG HU\textsuperscript{3} AND ZHIPENG CAI\textsuperscript{1}

\textsuperscript{1}Department of Computer Science
Georgia State University, Atlanta, GA, USA
\textsuperscript{2}Institute of Information Engineering
Chinese Academy of Sciences, Beijing, China
\textsuperscript{3}School of Big Data & Software Engineering
Chongqing University, Chongqing, China

(Communicated by Jiguo Yu)

Abstract. The past three years have seen the rapid increase of Bitcoin difficulty, which has led to a substantial variance in solo mining. As a result, miners tend to join a large open pool to get a more stable reward. Nowadays, mining pools take up over 98\% of Bitcoins total computation power. In a sense, this is a manifestation of Bitcoin that tends to be centralized. Thus, researchers have shown an increased interest in pool mining payoff and security. The purpose of this paper is to review and summarize recent research in Bitcoin pool mining system. We first introduce several common reward distribution schemes, and analyze their advantages and disadvantages with some improvement mechanisms; In the second section, to address pool security problems, we examined the practical utility of some existing and potential attack strategies. To study those malicious attack in details, several defense methods are collected. Finally, we make an outlook on Bitcoin future.

1. Introduction. Bitcoin is the first decentralized digital currency in the world. Nakamoto consensus, which is treated as one of core Bitcoin innovations, utilizes a challenging computational puzzle to determine the owner of next state block [31]. In the Bitcoin network, all the users need to participant in transaction validation, and the validated transactions form a public ledger that can prevent users from making a double spending of their Bitcoins [3, 21]. The users that participate in approving and verifying the correctness of Bitcoin transactions are called miners. The process of verifying transactions is also the process of mining Bitcoins. Specifically, the miner who successfully calculates the result can get an amount of 12.5 Bitcoins (BTC) as the block Reward in currently real Bitcoin market [52].

Mining pool is a kind of the mining infrastructure for most digital P2P cryptocurrencies, such as Bitcoin, Ethereum [44], Litecoin [54], etc. The existence of mining pools provides more mining options for miners. In the Bitcoin network, the significance of mining pools is to enhance the stability of Bitcoin mining process and stabilize the expected revenues/rewards of miners. In other words, by forming
a mining pool, it is more efficient to enhance Bitcoin mining success rate as well as to increase miners’ expected revenue, benefiting both the Bitcoin network and the miners.

Meanwhile, with the formation of mining pools, the methods of distributing block rewards among miners become a critical issue in pool mining. In the real Bitcoin market, some reward distribution mechanisms are widely adopted to attract miners and to improve revenue of mining pools, including proportional mechanism, slush’s mechanism, geometric mechanism, double geometric mechanism, pay-per-share mechanism, and pay-per-last-n-share mechanism. However, prior studies show that some existing reward distribution mechanisms are vulnerable to miners’ malicious behaviors, which can increase the malicious miners’ revenue while reducing the honest miners’ revenue [37, 33]. In the past years, there has been an increasing number of works investigating new attack strategies, revenue analysis, and solutions against these attacks [42, 4, 8, 23].

Nevertheless, most existing works analyze Bitcoin reward distribution mechanisms from one or several aspects, there lacks a systematic summary of Bitcoin pool mining strategies, attack strategies, and defense strategies. Therefore, to provide a comprehensive analysis of pool mining in the Bitcoin network, we in this paper aims to summarize and compare various the existing mainstream reward distribution mechanisms, withhold attack, and defense methods in mining pools. Moreover, we discuss some potential directions for the future research on pool mining. Our contributions are summarized as:

- A comprehensive and in-depth analysis of the existing popular reward distribution mechanisms in the Bitcoin network is proposed, which can help miners better choose a suitable allocation method.
- The withholding attack, including selfish mining, BWH attack, etc., is systematically investigated, which illustrates why attackers can enhance received revenue and compare attackers’ utilities in different scenarios.
- To resist the withholding attack, several defense strategies proposed in recent years are summarized.

The organization of this paper is as follows. In Section 2, the basic knowledge in pool mining is briefly introduced. Next, several popular reward distributing mechanisms are described in Section 3. The theoretical analysis of various types of withholding attacks are summarized in Section 4. Moreover, in Section 5, several mainstream defense strategies are presented. Finally, this paper is concluded in Section 6.

2. Preliminary.

2.1. Computational puzzle. In Bitcoin networks, all transactions in a block are stored in the form of a merkle tree, and the root of the tree is stored in the block header. Any miner who wants to generate a new block in the blockchain must figure out a computational puzzle. The computational puzzle in Bitcoin contains a double SHA-256 computation for each block. As shown in Eq. (1), every miner is required to compute an output value (i.e., Nonce) no larger than a threshold (denoted by D) which indicates the difficulty of the Bitcoin network [31].

\[
\text{SHA-256(Merkle + SHA-256(PreBlock + Nonce) \leq D)}
\] (1)
in which Merkle refers to the “merkle root”, and PreBlock indicates the Block ID for the last connected block. With this hash value, we can track all transactions stored in this block.

Technically, in order to get a block reward, a miner has to generate a random nonce value to satisfy the requirement of the computational puzzle. The randomness feature of SHA-256 computation in the puzzle makes it nearly impossible for miners to perform the inverse operation [9]. Even a slight change in the input can result in a completely different output. Hence, the only approach for miners to find the required nonce values is exhaustion, which usually consumes a huge amount of computational power. The difficulty of the computational puzzle is dynamically changed in the Bitcoin network. To be specific, whenever every 2016 blocks are mined, the difficulty in the Bitcoin network is adjusted such that the average mining time per block is regulated to be 10 minutes [51, 10].

2.2. Pool mining. Mining pool is a kind of infrastructure for Bitcoin and other virtual cryptocurrencies, and its significance is to enhance the stability of Bitcoin mining and stabilize miners’ revenues. Over the past three years, there has been a dramatic increase of computational power in the Bitcoin network [30]. The difficulty of the Bitcoin network has been growing fast correspondingly, which causes a great variance of revenues in solo mining. In particular, solo miners could hardly mine a block for years [40]. Thus, most miners choose to gather their computation power together in the form of mining pool to raise the probability of finding a new block. Once a block is successfully mined in the mining pool, the pool manager fairly distributes the reward to the participants corresponding to their expended computation efforts. Hence, in a long-term period, the miners in the mining pool can obtain more stable revenues than performing solo mining [29]. For example, the Bitcoin network hash rate is 5094526985 GH/s on Jun 17, 2017 [51]. Suppose a miner gets a powerful Antminer S9 mining equipment which has a hash rate of 13.5 TH/s. It still takes this miner about 7.2 years to mine a block and get the 12.5 BTC reward in average via solo mining. However, if this miner uses the Antminer S9 to join the mining pool with 15% of the total network computational power, the revenue would be $13.055 per day without considering the pool service fee. That is, miners can receive more rewards by participating in a mining pool.

In a Bitcoin block, the first transaction of the Merkle Tree, which is called a coinbase transaction, is used to record the block reward when a miner successfully mines a new block. In the pool mining scenario, the pool manager points the block reward to his own public key in the coinbase transaction and assigns the pool’s puzzle to the participants. Under this situation, if a malicious miner figures out a full solution and points the block reward to himself in order to swallow the block reward, the change of the transaction causes a difference of the merkle root value in the block header. As a result, the originally found nonce value is no longer a solution to the Bitcoin’s puzzle. In other word, this new block can not pass through the verification process of other miners, indicating that any miner in the mining pool cannot possess the block reward without sharing with others when he finds a full solution. Therefore, a miner must perform solo mining if he wants to obtain a full block reward. This feature ensures that a block reward processed by the pool is only distributed by the pool manager.

2.2.1. Share. Share is the minimum workload as defined in the mining pool. In order to share the reward of BTC, a miner should submit at least one share to the pool
manager, and the amount of the reward he can earn is determined by the distribution mechanism the mining pool adopts. As aforementioned, the computational puzzle requires miners to figure out a value less than the network difficulty.

Specifically, the difficulty level of mining a Bitcoin block can be expressed by the number of leading zeros of the blockhash. For example, for the 500155 block, there are 73 leading zeros, so the difficulty level of this puzzle is $2^{73}$. However, when a miner is mining in a pool, he is given an easier difficulty level which is generated by the pool manager. For instance, he only needs to find out a solution with 50 leading zeros. Each solution that meets the pool difficulty level is called a share. Thus, in the above example, finding a share is $2^{23}$ times easier than finding the original solution. By collecting shares, the pool can verify how much work the participants have done.

2.2.2. *Full Solution.* Once a share meets the difficulty level of Bitcoin puzzle, this share is considered as a full solution. With the full solution submitted by the miner(s), the pool manager can get the 12.5 BTC block reward and then distribute it to all the participants in the pool. In most cases, the expected reward of submitting a full solution is the same as that of submitting a share.

3. **Reward distribution mechanism.** The reward distribution mechanisms determine the approaches to assign rewards to the miners in a mining pool. Currently, ten biggest mining pools which have the largest hash rates possess over 93% of total computational power [42], and many of these pools adopt different distribution mechanisms, such as Pay-Per-Share (PPS), Slush Method, Pay-Per-Last-N-Shares (PPLNS), etc. In the analysis of a reward mechanism, *fairness* is a key factor to be considered. It is expected that each submitted share deserves a similar amount of rewards and the fluctuation of the received reward is not affected by the time factor. In 2016, Schrijvers *et al.* defined that the incentive-compatibility of a reward distribution mechanism means reporting a full solution at once is every miner’s best strategy [37]. Incentive-compatibility is considered as a good evaluation metric to analyze fairness in recent mining pools [13, 22].

In this section, some popular reward distribution mechanisms, including their advantages and disadvantages, are investigated.

3.1. **Proportional mechanism.** The proportional reward mechanism is one of the early popular distribution mechanisms. One key factor in the proportional mechanism is called *round*, where a round is the time interval between two mined blocks as shown in Figure 1.

**Figure 1.** Proportional Mechanism
The idea of this mechanism is to split the reward according to the proportion of shares that a miner submits in the all submitted shares in a round. At the end of each round, if a mining pool can successfully find a block in a round, the mining pool can obtain a reward of $B$, and the pool manager takes a fraction $f$ of reward $B$ as the pool operating fee. Suppose the total number of submitted shares is $N$ and a miner submits $n$ out $N$ shares in this round, then the miner could get an amount of reward \( \frac{n(1-f)B}{N} \) \[33\].

Although the length of any round cannot be predicted, the number of shares already submitted during a round can be counted. Hence, the amount of reward of each share in a round is affected by the time. In other words, the proportional mechanism is not incentive compatible, which may lead to the following problems \[37\].

3.1.1. Delay Submission. The delay submission problem occurs when the proportion of a miner’s submitted shares in all the submitted shares within a round is less than the proportion of the miner’s mining power in the mining pool’s mining power. In this situation, the miner may gain more benefit by holding a full solution.

**Theorem 3.1.** Suppose a miner $p$ with hash power $\alpha$ out of 1 finds a full solution after submitting $n$ out of $N$ submitted shares in a round. Let $U_1$ be the expected reward for immediate submission and $U_2$ be the expected reward for holding one or more shares. If $\alpha > \frac{n}{N}$, we have $U_2 > U_1$.

**Proof.** The block reward is denoted by $B$, and the pool operating fee is represented by $fB$. There exist two strategies for the miner $p$ to choose when he successfully finds a full solution: (i) he submits the full solution immediately; and (ii) he holds one or more shares of the full solution. In the first case, the miner $p$ performs honestly. Thus, he could get the reward corresponding to the proportion of the shares he submits, which is computed in Eq. \(2\).

\[ U_1 = \frac{n}{N} \times (1 - f)B. \] \[2\]

However, if the miner $p$ chooses to hold one or more shares, there is a probability of $\alpha$ that he can successfully find the next share, obtaining an expected reward $\frac{\alpha(n+1)}{N+1} \times B$. Accordingly, the expected reward that $p$ can not find the next share is $\frac{(1-\alpha)n}{N+1} \times B$. Thus, the total expected reward for holding one or more shares is:

\[ U_2 = \left[\frac{\alpha(n+1)}{N+1} + \frac{(1-\alpha)n}{N+1}\right] \times (1 - f)B \]

\[= \frac{\alpha + n}{N+1} \times (1 - f)B. \] \[3\]

Since $\alpha > \frac{n}{N}$, we can get:

\[ U_2 > \left[\frac{n/N + n}{N+1}\right] \times (1 - f)B \]

\[= \frac{n}{N} \times B \]

\[= U_1. \] \[4\]

Thus, with $\alpha > \frac{n}{N}$, the miner $p$ can always receive a larger reward by holding one or more shares. According to this result, an increasing amount of miners may choose to wait to submit a full solution. This phenomenon can lead to a tremendous waste of power consumption. \(\square\)
3.1.2. Pool Hopping. Pool hopping is always viewed as one of the most vulnerable weaknesses of the proportional mechanism [34, 41]. In a mining pool, the expected reward for each share is \( (1-f)B \). The longer a round is, the more shares are submitted. Therefore, as the total number of shares, \( N \), is increased, the value of each share in a longer round is reduced. Such feature of the proportional distribution mechanism results in the pool hopping strategy for rational miners to switch among pools to gain rewards as many as possible. Specifically speaking, the rational miners choose to mine only when the expected reward is high and to leave when it is low. Thus, it is beneficial for a miner to mine within a short round and leaves when the expected reward is reduced. It is shown that the threshold to perform pool hopping strategy is the situation when the number of all submitted shares is 43.5% of the difficulty level [33]. Accordingly, the rational miners would choose to mine in a mining pool when the total number of shares is below this threshold and leave the pool when the total number of shares is higher than this threshold. If the rational miners who choose to adopt hopping strategy can constantly earn the rewards more than the expected reward without hopping, more honest miners lose their deserved profits. Moreover, if all miners choose to leave the mining pool when the number of all submitted shares is 43.5% of the difficulty level, no block can be mined in the pool.

3.2. Slush’s mechanism. Slush mechanism is introduced to prevent pool hopping problem [57]. Compared with proportional mechanism, slush mechanism measures mining power by scoring hash rate instead of simply counting submitted shares. As we discussed above, the value of each share decreases in time under proportional mechanism, i.e., the newer submitted shares are worth more than the older submitted shares in each round. Thus, slush mechanism aims to give more weight to those shares submitted in the late period of each round and to make the expected reward of each share equal. Slush mechanism uses an exponential function to calculate the score of a share \( s \) at time \( t_0 \), i.e.,

\[
C(s, t_0) = e^{\frac{\tau(s)-t_0}{\lambda}},
\]

where \( \lambda \) is a parameter controlling how fast the score of a share declines in time. The lower the \( \lambda \) is, the faster the decline is. Currently, the value of \( \lambda \) is set to be 1200 in real-world slush mining pools [57].

Let \( S_m(t_0) \) be all valid shares submitted by miner \( m \) at time \( t_0 \). For example, if at time \( t_0 \), \( m \) has already submitted 3 shares, then \( S_m(t_0) = \{s_1, s_2, s_3\} \). Define that \( C'_m(t_0) \) is the score of miner \( m \) at time \( t_0 \), which can be calculated by a sum of exponential functions as follows:

\[
C_m(t_0) = \sum_{s \in S_m(t_0)} C(s, t_0) = \sum_{s \in S_m(t_0)} e^{\frac{\tau(s)-t_0}{\lambda}},
\]

Similarly, the score of mining pool \( p \) at time \( t_0 \), denoted by \( C_p(t_0) \), can be represented as the sum of all miners’ scores, i.e.,

\[
C_p(t_0) = \sum_{m \in P} C_m(t_0).
\]

To distribute the reward among pool miners, slush mechanism calculates each miner’s contribution to the pool using the ratio of the miner’s score to the pool score. Accordingly, the reward of miner \( m \) at time \( t_0 \), denoted by \( U_m \), can be
computed via Eq. (8).

\[ U_m(t_0) = \frac{C_m(t_0)}{C_p(t_0)} \times B. \]  

(8)

Despite a notable improvement made by slush mechanism, there are still some drawbacks. One issue is that the stable status of slush mechanism is reached some time after a round starts. Thus, slush mechanism cannot guarantee incentive compatibility.

3.3. Geometric mechanism. Another score-based reward distribution mechanism inspired by slush mechanism is geometric mechanism. The main difference from slush mechanism is that geometric mechanism introduces a novel variable fee \( c \) to address the weakness of slush mechanism. The variable fee is relatively high at the beginning of each round and decreases in time. Thus, with such design, the expected reward of each share can be kept stable in mining pools, indicating that geometric mechanism can achieve incentive compatibility.

Notice that, in other reward distribution mechanisms, the reward assigned among pool minters is \((1 - f)B\). By introducing the variable fee \( c \) paid by miners, the expected reward sent to miners is reduced to \((1-c)\cdot(1-f)B\) in geometric mechanism. Although a pool manager can adjust both the constant and the variable fees in a mining pool, geometric mechanism is still not as attractive as the following Pay-Per-Share (PPS) or Per-Per-Last-N-Shares (PPLNS) mechanisms due to the additional fee.

Thus, by combining both the advantages of PPLNS and geometric mechanisms, a new reward mechanism for mining pool called Double Geometric Method (DGM) was introduced into practice [33].

3.4. Double geometric mechanism. DGM combines the advantages of both PPLNS and geometric mechanism, in which the share value decays geometrically along two directions with one determined by new share discovery and the other determined by new block generation [35].

Besides using the fixed fee and average variable fee in DGM similar to those in geometric mechanism, there is a cross-round leakage parameter in DGM to balance the variance of miners’ revenue and pool manager’s risk. When a new block is found, PPLNS mechanism maintains miners’ scores unchanged, geometric mechanism sets all scores to be 0 and transfer them to the pool manager, while DGM chooses compromise to transfer part of the scores to the manager. When multiple new blocks are mined out, the manager gets a large portion of the scores. Therefore, DGM can reduce the impact on the miners’ reward variance caused by luck. Particularly, when the cross-round leakage parameter is set to be 0, DGM is the geometric mechanism, and when the parameter approaches 1, DGM becomes a variant of PPLNS.

3.5. Pay-Per-Share mechanism. PayPer-Share (PPS) was the most common distribution mechanism used in open pools several years ago. In a PPS system, regardless of how many shares submitted in a round, as long as a miner submits a share, he is immediately rewarded according to his expected contribution. Suppose a miner with a computational power of \( \alpha \) out of 1 in the Bitcoin network, the expected reward of the miner is \( \alpha B \) for each block he finds. That is, in PPS pools, there is no variance in the reward per share and hardly any waiting time for miners to receive their rewards. Such features make it easy for miners to estimate and
verify their exact rewards. Thus, it is attractive for miners to join PPS mining pools.

PPS mechanism is proved to be incentive compatible because there is no difference between holding a solution and publishing it immediately. For a PPS pool, there exists a big possibility that the pool cannot mine any block in a time period, but the pool operator still needs to pay the miner who contribute to the pool. Thus, the operator of a PPS pool should take the risk of mining failure as well as have sufficient deposit to maintain the pool stability, which also explains the reason why the pool operators always set up a relatively high percentage of pool fees in PPS pools. Specifically, a pool operator needs to set up a financial reserve to keep the bankruptcy probability low enough [33].

3.6. Pay-Per-Last-N-Share mechanism. Pay-Per-Last-N-Share (PPLNS) mechanism is a variant version of PPS scheme and is also commonly used in real-world. Instead of paying miners for each submitted share, PPLNS rewards miners according to every $N$ submitted shares.

In PPLNS mechanism, there is no definition of round. Instead, shares fall into slots. In particular, rewards are sent out for every $N$ shares received by a pool manager. Regardless of whether there exist any blocks found in a slot, the pool manager distributes the reward to every share that are submitted within prior $N$ shares. Thus, the expected reward for each share is $(1-f)^{BL}N$, where $L$ is the number of full solutions among the next $N$ shares. As shown in the example of Figure 2, the pool manager uses the rewards got in slot 2 to pay for each share submitted in slot 1, which is like a credit card payment.

![Figure 2. An example of PPLNS Mechanism](image)

Moreover, prior research [33] found that the number of full solutions $L$ follows a standard poison distribution with $\lambda = 1$. Thus, the average number of full solutions is 1 in each slot. Schrijvers et al. proved that PPLNS mechanism is incentive compatible when a proper value of $N$ is selected [37]. Recent work proposed by Zolotavkin et al. used a game theory model to derive the specific conditions when PPLNS method is incentive compatible [50].

3.7. Summary. The objective of all these aforementioned mechanisms is to provide pool miners with a fair and attractive method of reward distribution. From the viewpoint of a miner, he wants to join a mining pool for a stable and high expected reward. On the other hand, from the viewpoint of a pool manager, he intends to attract more miners to enhance his computational power while reducing the risk of bankruptcy at the same time.

Table 1 presents the reward distribution mechanisms in some famous real mining pools. The most popular mechanisms adopted by open pools are PPS and
PPLNS [56]. This two mechanisms can be easily deployed in pools. Also, it is clear for participants in such pools to calculate their expected rewards. To obtain the stable revenue in PPS mechanism, the operators in the PPS pools usually set relatively high fees. Thus, the PPS pools give participants a feeling of stable but not superior. In order to improve this situation, the operators often introduce some unique elements into their allocation to attract more participants. For examples, AntPool uses a variety of distribution mechanisms at the same time for miners to choose, and BTC.com uses full pay per share (FPPS) which not only rewards regular dividends (12.5 BTC for now) but also some of the transaction fees, thus increasing miners’ expected revenue. In future Bitcoin transactions, the part of transaction fee will gradually increase. Therefore, FPPS can well adapt to the future development of Bitcoin pool mining. Slush and Geometric mechanisms are not widely implemented in real Bitcoin systems due to their obvious shortcomings. Some other mechanisms may combine several of above schemes to let participants themselves decide how they want to get reward.

One big issue in the reward distribution system nowadays is security. Existing works have investigated a variety of attack strategies in pool mining, especially in those running PPS and PPLNS mechanisms. In the next section, we will discuss and analyze several major malicious attacks.

### 4. Withholding attack

According to the original Bitcoin white paper, Nakamoto argued that as long as all Bitcoin miners follow the consensus rule, Bitcoin is able to remain stable [28]. However, it is shown the pool miners can get unfair advantages by performing malicious attacks in pool mining, thereby making Bitcoin system not incentive compatible [12, 32, 55, 11].

Bonneau et al. defined the stability of Bitcoin consensus from five aspects including eventual consensus, exponential convergence, liveness, correctness, and fairness [4]. The Bitcoin system remains stable if all these five properties are held. Malicious attacks may harm at least one aspect, thus causing the Bitcoin unstable. For example, an attacker is likely to expect a higher revenue by submitting a share rather than mining honestly. In this paper, we investigate the most traditional and famous attack strategies and analyze how they can harm the Bitcoin system.

Most Bitcoin pools are open to the public, which allows anyone to participate in. Researches have indicated that attackers can directly join an open pool and get extra unfair revenues by never sharing or delaying sharing their proof of work [42, 11]. In this paper, we define such kind of attacks as *Withholding Attack* and propose a systematic analysis of it based on several existing works. To understand why withholding attack can make more profit, here we use a traditional block withholding...
attack (BWH) as an example. Suppose Bob is the owner of a pool with 20% of the total computational power of the Bitcoin network. Consider that the reward for mining a Bitcoin block is 12.5 BTC today, and assume that Bob purchases additional 5% mining equipment (i.e., Bob has additional 5% computational power). He may have two choices: (i) mining honestly, and (ii) infiltrating other pool and withholding the full solution. The expected rewards of adopting these two strategies are compared as follows.

**Case 1: Honest Mining.**

\[
U_h = \frac{25}{105} \times 12.5 = 2.976 \text{ BTC},
\]

where \(U_h\) denotes Bob’s expected reward when mining honestly in the Bitcoin network.

**Case 2: Infiltrating Mining (Withholding Full Solution).**

\[
U_i = \left( \frac{5}{85} \times \frac{80}{100} + \frac{20}{100} \right) \times 12.5 = 3.088 \text{ BTC},
\]

in which \(U_i\) represents Bob’s expected reward when withholding full solution during pool mining process in another pool.

One can see that the expected reward of infiltrating mining is higher than that of honest mining. Thus, a rational miner will choose to perform withholding attacks to get more profit.

Withholding attack can be implemented in two ways, i.e., withholding block and withholding shares. More specifically, the behavior that a miner initially keeps blocks secret after finding them is viewed as withholding block attack, and the behavior that a miner keeps shares secret is treated as withholding share attack. A further analysis is discussed below.

4.1. **Selfish mining.** The selfish mining attack is first introduced by Eyal and Sirer in 2013 [12]. They have shown that Bitcoin POW is not incentive compatible under selfish mining attack, in which colluding miners can get a higher revenue.

The basic idea of selfish mining attack is to cause the honest miners waste their mining power on unnecessary computations. Instead of announcing a block to the network directly, the attacker keeps it secret and tries to find two blocks in a line on its private chain before others find the next one on the public chain. Thus, the attacker intends to orphan the block mined by the honest miners.

There are six scenarios of a simplified selfish mining attack, which is presented in Figure 3. As scenario 1 of Figure 3 shows, when the private branch generated by the attacker is shorter than the public branch, the attacker will directly adopt the block on the main chain, thereby avoiding waste of computational power. However, in scenario 2, if the private branch is ahead of the main chain to get one block (i.e., \(B(i)\)), the attack will withhold the block \(B(i)\) and try to compute the next block, i.e., \(B(i+1)\). During this period, if honest miners can find a block on the public chain as shown in scenario 3, the attacker immediately publishes what he has found. Thus, this scenario leads to a competition on propagation race, which is known as the tie-breaking attack [14]. If the propagation rate of the network is the same for all miners, the attacker would win the race and get the block on public chain orphaned with a probability of 50%. Besides, in scenario 4, if the attacker finds the next block \(B(i+1)\) before the public chain finds \(B(i)\), which means the attacker is leading the main branch two blocks and is likely to withhold these two
blocks until the honest miners find one block. Then, the attacker will publish two blocks together to form the longest chain when $B(i)$ is found on the public branch, where the honest miners in scenarios 5 and 6 waste their computational power on finding the $B(i)$ block.

Based on the aforementioned selfish mining strategies, Eyal and Emin claimed that as long as the attacker possesses more than 33% of the computational power of the Bitcoin network, it is profitable for him to launch selfish mining attack even if he loses every race attack in scenario 3 of Figure 3 [12]. Hence, to prevent a network from being attacked by selfish mining, there should be at least $\frac{2}{3}$ of computational power of miners act honestly. Selfish mining harm the fairness of the Bitcoin network in the following ways. When the mining power of a pool exceeds a threshold, it is profitable for attackers to implement selfish mining strategy, attracting more malicious miners to join the pool and increasing their revenues. This will result in an increase of the pool size. Once the size of the malicious pool exceeds 50% of the total mining power in the Bitcoin network, attackers no longer need to perform selfish mining attack because he can launch the original 51% attack to control the whole Bitcoin network. Several studies have been proposed to help defend the selfish mining attack, which will be covered in the next section [2, 45].

4.2. Witholding share. Besides selfish mining, malicious miners are also able to enhance their rewards by withholding their shares in mining pools. There exist two cases for withholding shares in mining pools.

**Case 1: Delay Full Solution.**

This case happens in some outdated pools such as those adopting proportional mechanism. As long as the proportion of shares submitted by a miner is less than the proportion of mining power he owns, the miner could always increase his received reward by withholding one or more shares to delay full solutions, which is proved in Theorem 3.1. Extending to a wider aspect, if the reward distribution mechanism is not incentive compatible, which means the reward of shares is affected in time...
domain, it is seductive for miners to conduct delay full solution attack. Although most open pools nowadays focus more on incentive compatibility when designing their reward distribution scheme, some small pools may be threatened by such attack.

**Case 2: Withhold Full Solution.**

Withholding full solution, which is the typical form of block withholding attack (BWH), was first defined in 2011. BWH occurs in the case where a malicious miner sends only partial proof of work to the pool manager and discards full proof of work [33]. By taking this strategy, the malicious miner is treated as a regular participant in the mining pool due to the partial shares received by the pool manager. Therefore, the malicious miner could share the reward from the pool without truly contributing to the pool. In the BWH example, it is shown that if a miner gets sufficient computational power in Bitcoin network, he can always infiltrate other pools to expect a higher reward. Even though a pool can use its expected mining power divided by its actual mining power to check the BWH attack, it is hard for pool managers to identify which miners indeed perform BWH attack.

The effect of block withholding attack has always been a debatable topic. Initially, some studies argued that BWH attack is not profitable in practice, and thus if the whole network of mining pools is sufficiently decentralized, pool managers do not need to worry about this attack. The reasons for this conclusion lie in many aspects, such as attacker may suffer from a net loss and honest miners may take counterattack strategies. On the other hand, however, other studies showed that there is always an incentive for attackers to launch BWH attack. BWH attack can be analyzed by using game theory based on various analytic models. In this paper, we mainly discuss three existing analytic models and compare their results.

The first systematic game theory-based analysis of BWH attack in pool mining was proposed by Eyal in 2015 [11], in which the malicious behaviors to perform BWH attack is modeled as “the miner’s dilemma” under two specific scenarios. In the first scenario, only a miner attacks other mining pools as shown in Figure 4. Eyal proved that if there is only one attacker, the attacker can always gain more rewards by implementing BWH attack, thereby indicating that no pool attack is not a Nash Equilibrium. Moreover, a threshold has been calculated for an attacker to get the maximum reward in this scenario.

**Figure 4.** BWH attacker scenario: one attacker

In the second scenario, two miners that are from different pools can attack each other, which is illustrated in Figure 5. If no infiltrating happens, an attacker can always enhance his reward by conducting BWH attack. Thus, the only equilibrium
happens in the situation where neither the two pools can increase their rewards by changing the infiltration rates. However, in the equilibrium situation, both the two attackers’ received awards are less than when they do not launch BWH attack. For example, if DiscusFish and AntPool attack each other by choosing their optimal infiltration rates, these two pools would lose 4% and 10% of their revenues respectively compared with the situation when no BWH attack towards both pools. As a result, if more participants choose to perform BWH attack, there would be a huge waste of computational power in the Bitcoin network.

Later in 2016, Luu et al. conducted a more comprehensive theoretical analysis of BWH attack [27]. In their paper, the authors viewed pool mining as computational power distribution games and investigated the approaches for attackers to rationally allocate their computational power to maximize benefits by performing infiltrative BWH attack. Specifically, the analysis of pool mining in the proposed computational power distribution games can be categorized into four kinds of scenarios. In the first scenario, the entire Bitcoin network is viewed as one big mining pool that can be accessed. An attacker in this scenario can earn more reward by implementing BWH attack, and the received reward can be maximized if the attacker uses half of his computational power to infiltrate others. In the second scenario, the Bitcoin network is composed of multiple mining pools that can be accessed, and attackers can only choose one of them to attack, which is the same as that in [11]. Luu et al. also showed that attacking a large ore pool can obtain more extra rewards than attacking a smaller one. Moreover, there exists a threshold $\beta$ such that attackers can maximize their rewards, in which $\beta$ is the ratio of an attacker’s computational power corresponding to the computational power of the entire Bitcoin network and can be computed as follows [27]:

$$\beta = \frac{-\sqrt{-p'^2(\alpha p' + \alpha - 1)} - \alpha p' + p'}{\alpha(\alpha + p' - 1)},$$

(11)

where $\alpha$ and $p'$ are computational power owned by the attacker and the pool being attacked, respectively. It is worth mentioning that the threshold value of Eq. (11) is slightly different from the threshold given in [11]. Based on our verification, Eq. (11) provides a more accurate value. In the third scenario, an attacker could choose to launch BWH attack towards multiple mining pools at the same time, where the attacker can receive the maximum reward from implementing BWH attack by mining in all the remaining mining pools. The last and the most common scenario considers that the dishonest miners dominated the Bitcoin network. Based on the
analysis in [27], the dishonest miners can still get more rewards by performing BWH attack. Thus, no matter in which cases, implementing BWH can always bring an increase of reward to attackers in the Bitcoin network. Moreover, the authors of [27] also believe that attackers would adopt a mixed strategy for BWH attack, but Prisoners’ dilemmas will occur when most miners adopt BWH attack strategy.

In 2017, Velner et al. innovatively considered a bribe-style BWH attack [42], which is demonstrated in Figure 6. Different from previous infiltrative BWH attacker, the attackers in [42] bribe some miners in the mining pool to withhold full solutions rather than directly pouring their computational power into the mining pools. To be specific, if a victim pool is running the most common distribution mechanism PPS, an attacker can simply gain more reward by conducting BWH attack as long as he possesses at least $\frac{1}{50,000}$ of the computational power in the entire Bitcoin network.

![Figure 6. Bribe-style BWH attack scenario](image)

In the work proposed by Bag et al. in 2017, they studied the BWH attack under the Bitcoin network architecture the same as the above mentioned [1]. However, they observed that BWH attack can indirectly increase the gain of other honest mining pools. Figure 7 shows that an attacker can collude with some mining pools to get a certain fraction of their indirect gained revenues as bonus. In this way, the expected revenue of the attacker could be higher than when conducting classic BWH attack. This type of BWH attack is called sponsored block withholding attack.

![Figure 7. Sponsored BWH Scenario](image)

In current mainstream analysis, BWH attack, which is still a server threat to pool mining in the Bitcoin network, has attracted lots of attentions. Both invading
and bribing BWH attacks can provide attacker with unfair advantages to increase their revenues. However, the attackers still need to prove the computational power of the bribed, infiltrated and sponsored miners. More specifically, on the one hand, the attackers should verify whether a victim pool does find any full solution; on the other hand, it is also necessary to verify that the bribed, infiltrated and sponsored miners do not release blocks to the entire network. Since BWH attack is hard to be detected [11], without these two verification mechanisms, the bribed, infiltrated and sponsored miners would possibly disobey the attackers by releasing full solutions and gain more rewards from their mining pools without the notice of the attackers. Therefore, when an attacker chooses to conduct BWH attack, he needs to consider the cost of various verification mechanisms as well as some counter attacks strategies. This explains why BWH attack is theoretically validated profitable but is not ubiquitous in practice.

4.3. Summary. In this section, we systematically classify the withholding attack in the Bitcoin mining pools. Then, we investigate different types of withholding attack from the delayed releasing attack in the proportional mining pools to the selfish mining attack then to the traditional infiltrative or bribe-style BWH attack, in which an in-depth understanding of the withholding attack is obtained. According to the current research results, the feasibility of withholding attack still exists. In addition, the withholding attack is usually undetectable, because it can be simply interpreted as bad luck. Attackers can use many different identities to mine at the same time, so it is not suspicious of him with no solution submitted.

Although the profitability of selfish mining attack and BWH attack in practice is still open to question, variant attacks based on these two types of withholding attack, such as Block Discarding Attack [2], may still pose a significant threat to the Bitcoin network. More and more researchers have begun to tackle the withholding attack and possibly centralized problems via trying to change the Bitcoin authentication mechanism or adding some countermeasures to the mining pools. In the next section, we present various approaches to defend the withholding attack and make a future outlook on defense strategies.

5. Defense method. Researchers have never stopped the study of defensive measures to resist the withholding attack. In this section, we summarize the mainstream defense methods against the withholding attack. The principle to adopt reward distribution mechanism is that the mining pool should ensure the incentive compatibility of each share, requiring that the estimated revenue/reward of each miner does not greatly depend on the time duration of finding a full solution. Therefore, in general, the defense methods need to ensure fairness of the entire Bitcoin network.

Currently, delaying full solution mentioned in Section 4.2 is uncommon, because the proportional distribution mechanism has been abandoned by most mining pools. To defend BWH attack, the most straightforward strategy is to give a bonus to every miner who has found a full solution. The reason is that increasing the expected revenue of discovering a full solution can significantly raise an attacker’s cost of implementing BWH attack. Meanwhile, the additional bonus of the full solution also enlarges the variance of pool miners’ rewards, which runs counter to the original intention of performing pool mining. Thus, one potential solution to defend the withholding attack is to design new incentive compatible reward distribution mechanisms while achieving fairness among pool miners. In the existing works, there are three major types of defense strategies against selfish mining and BWH attack,
including tie-breaking defense [11, 18], backward-incompatible defense [2, 38], and timestamp-based defense [39, 45], which are detailed in the following.

5.1. **Tie-Breaking defense.** Tie-breaking defense approach was proposed by Eyal [11] and Heilman [18], which can reduce the resources that honest miners waste on isolated blocks. The tie-breaking defense is a kind of relatively primitive the backward-compatible defense.

In [11], to prevent selfish mining, all the miners should choose one branch to mine at random when a miner learns that competing branches have the same length, in which a race competition emerges. An example of such race competition is presented in Figure 8.

![Figure 8. Race Competition](image)

The race competition, theoretically, could lead to that half the miners (in expectancy) mining on the pool’s branch (e.g. the public chain in Figure 8) and the other half mining on the other branch (e.g., the private chain in Figure 8). As a result, an attacker would win the race competition with a probability of 50%. What’s more, since the attacker makes calculation in advance on his private chain, the probability of digging up the next block is greatly increased once the attacker win the race competition. Additionally, when 50% of the honest miners select the wrong public chain, a huge waste of computation power would be brought to the honest miners. Thus, the key idea of tie-breaking defense is that the choice of each miner is arbitrarily determined by the network topology and latency when there are two branches of equal length. Notice that this design does not introduce new vulnerabilities to the Bitcoin network. The tie-breaking defense in the Bitcoin network ensures that attackers cannot profitably engage selfish mining activities if the computational power of their mining pools is less than $\frac{1}{4}$ of the total mining power. Later in 2015, Sapirshtein et al. [36] proposed the optimal tie-breaking defense for performance improvement. In this defense strategy, selfish miners will lose every tie, and thus they need to control more computational power to execute selfish mining attack.

However, the tie-breaking defense cannot withstand an attacker who owns a large computational power when the length of the selfish chain is longer than the public chain.

5.2. **Backward-Incompatible defenses.** Backward-incompatible defense was developed by Bahack [2] and Shultz [38], which aims to change the validity rule of the entire block such that selfish mining and BWH attacks no longer exist.

The main purpose of most attack strategies is to persistently fork the public chain into short branches for getting extra unfair revenue. Therefore, to resist there malicious behaviors, the backward-incompatible defense tries to make fundamental
changes on the block validity rules or reward distribution mechanisms. In [2], Bak-hack introduced a fork-punishment rule that can make withholding attack strategies unprofitable. In particular, it is suggested that not rewarding the miners of a block that has a competitive block of another same-length branch though the miners belong to the winner branch. However, to achieve such changes, the Bitcoin network participants need to upgrade their clients. This may lead to a soft fork because of the different upgrade propagation speed.

By adopting the backward-incompatible defense, the problem of withholding attack can be solved theoretically. But, due to the consensus-level change of the entire Bitcoin network, instability may yield during network updating process, leading to a Bitcoin soft-fork. Therefore, the backward-incompatible defense is often more applicable to newly invented cryptocurrencies rather than to Bitcoin.

5.3. Timestamp-Based defense. Timestamp-based defense, which was proposed by Solat et al. [39] and Zhang et al. [45], is designed based on timestamp in the block. The common denominator of this type of protection is that if a selfish miner keeps a block privately more than a threshold of time duration, his block will be rejected by honest miners. The major advantages of the timestamp-based defense include: (i) it is backward-compatible; and (ii) it can be easily updated on the originally entire Bitcoin network without causing instability.

According to Zhang et al. [45], the main idea of their work is to replace the original Bitcoin fork-resolving policy (FRP) with a new kind of weighted FRP. They claim that their proposed timestamp-based defense method is the first backward-compatible defense method. To better make their weighted FRP compatible with the original Bitcoin network, a parameter $k$ is introduced. If there are at least $k$ blocks ahead of miners’ competitors, the miners are requested to select the longest chain; otherwise, the miners adopt the weighted FRP. In [45], there is only one colluding pool of selfish miners in the system model, which is considered to be the strongest attack scenario due to the reason that all the attackers gather their computational power together.

It is assumed that the upper bound of block propagation delay among miners is $\delta$ [45]. If a block received after a competing block of the same length is within the time threshold $\delta$, this block is considered to be in time; otherwise, the block is treated as late and can not be calculated in weight. A concept called uncle was also proposed, where a block $B_1$ is considered to be the uncle of another block $B_2$ if $B_1$ is a competing in time block of $B_2$’s parent block. Accordingly, the weight of a chain is counted as the number of its in time blocks plus the number of in time uncle hashes embedded in those in time blocks. Based on these settings, Zhang et al. designed the following mining scheme for the Bitcoin network:

- **Case 1:** If the difference between two chains is more than $k$ blocks, miners mine on the longest chain.
- **Case 2:** If the difference between two chains is less than $k$ blocks, miners choose the chain with the largest weight to work on.
- **Case 3:** If the largest weight is the same on multiple chains at the same time, miners select one among them randomly.

In particular, when $k = 1$, their proposed mining scheme is exactly the same as the tie-breaking defense. According to their analysis, the proposed scheme leads to attacker’s dilemma for selfish miners. An example of such attacker’s dilemma is shown in Figure 9. To be concrete, if an attacker publishes his block found in
the private chain, this block will be counted in the weights of both the public and private chains (e.g., the weights of both the public and the private chains are 3 in Figure 9); if he does not publish it, this block is not counted in the weight of any chain because of being late (e.g., the weights of both the public and the private chains are 2 in Figure 9). Thus, in both situations, the weights of public chain and private chain keep the same, which indicates that the attacker can not get any advantages by withholding blocks.

![Figure 9. Attacker’s dilemma](image)

With the increase of value $k$, the performance of the proposed mining scheme can be improved. Let $\alpha$ represent the probability that a new block is mined by selfish miners; that is, $\alpha$ is the ratio of selfish miners’ mining power to the total mining power of the Bitcoin network. The weighted FRP outperforms other mining algorithms when $0.4 < \alpha < 0.5$. As $\alpha$ is reduced less than 0.4, only the optimal tie-breaking algorithm performs better than the weighted FRP.

However, due to modifications of the block validation rule, the timestamp-based defense strategies are likely to result in more energy consumption compared with original Bitcoin POW, which are inherently proven to be wasteful.

5.4. Further discussion. First of all, as we know, the Bitcoin network is constantly being updated as a deflationary digital cryptocurrency. The block reward for each block is halved every four years. It is predicted in the future that transaction fee will be the dominant reward for Bitcoin. If there is only transaction fee, the variance of the block reward will become very high due to the exponentially distributed block arrival time. Thus, the existence of current withholding attack becomes questionable, which needs more investigation in the future.

Second, besides the withholding attack, there exist other attacks in mining pools. For instance, in 2016, Carlsten et al. introduced an undercutting attack strategy [7]. In a blockchain with constant forks caused by undercutting, an attacker’s effective hash power is magnified because he can always mine to increase his blocks whereas other miners may not be unified. This would make the 51% attack possible when
an attacker’s hash power is much less than 51% of the hash power of the Bitcoin network. The undercutting mining behavior could hurt the stability of mining process in the Bitcoin network. Thus, more effort should be done to study different kinds of malicious attacks and corresponding counter measures in pool mining for performance enhancement in the Bitcoin network.

Third, almost of all the existing works focus on miners’ malicious behaviors and simply assume that pool managers are trustworthy. The behaviors of pools managers have not been studied yet. However, as a matter of fact, it is still doubtable whether pool managers can receive more profits through manipulation in their mining pools. Thus, to gain more new insights into pooling mining in the Bitcoin network, the research on pool managers’ behaviors would be conducted.

Last but not least, transparency of Bitcoin transactions can help trace back miners’ historical transactions and mining behaviors but also leak miners’ sensitive and private information at the same time. As privacy has been one of the most important issues in many real-world applications and is getting more and more attentions, it is inevitable that privacy protection will become a critical research direction to further promote widespread adoption of Bitcoin-based applications. Protecting privacy should take into account requirements of various application scenarios, such as Internet of Things (IoT) and online social networks, which could be inspired by many existing works [19, 20, 24, 5, 6, 25, 46, 47, 43, 26, 49, 17, 48, 15, 16].

6. Conclusion. In this paper, we compared several existing reward distribution mechanisms, summarized the withholding attacks and categorized some defense strategies. As a case study, Bitcoin Pool Mining is a good place to apply game theory model.

Discussions and debates on the Bitcoin and blockchain have never ceased since its inception. Perhaps we can get some inspiration from the evolution of Pool Mining technology. With the further development of technology, we believe that Mining Pool will be more fair and secure.

Acknowledgments. This research work was partly supported by the NSF NOs. 1704287, 1252292, 1741277, Science and Technology Innovation Seed Project of Sichuan Province under grant No.(2017RZ0008), International cooperation and exchange Project of Sichuan Province under grant No.(2017HH0075), Chinese Scholarship Council under grant No.(20173109), the National Natural Science Foundation of China (61702062), the Natural Science Foundation of Chongqing, China (cstc2018jcyjAX0334), and Overseas Returnees Support Program for Innovation and Entrepreneurship of Chongqing (cx2018015).

REFERENCES

[1] S. Bag, S. Ruj and K. Sakurai, Bitcoin block withholding attack: Analysis and mitigation, *IEEE Trans. Information Forensics and Security*, 12 (2017), 1967–1978.
[2] L. Bahack, Theoretical bitcoin attacks with less than half of the computational power (draft), CoRR, vol. abs/1312.7013, 2013. [Online]. Available: arXiv:1312.7013
[3] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer and S. Welten, Have a snack, pay with bitcoins, *IEEE P2P 2013, Proceedings*, 2013, 1–5.
[4] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll and E. W. Felten, Sok: Research perspectives and challenges for bitcoin and cryptocurrencies, in *2015 IEEE Symposium on Security and Privacy*, 2015, 104–121.
[5] Z. Cai, Z. He, X. Guan and Y. Li, Collective data-sanitization for preventing sensitive information inference attacks in social networks, IEEE Transactions on Dependable and Secure Computing, 15 (2018), 577–590.

[6] Z. Cai and X. Zheng, A private and efficient mechanism for data uploading in smart cyberphysical systems, IEEE Transactions on Network Science and Engineering (Early Access), 2018, 1–1.

[7] M. Carlsten, H. Kalodner, S. M. Weinberg and A. Narayanan, On the instability of bitcoin without the block reward, in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, 154–167.

[8] N. T. Courtois and L. Bahack, On subversive miner strategies and block withholding attack in bitcoin digital currency, CoRR, vol. abs/1402.1718, 2014. [Online]. Available: arXiv:1402.1718

[9] N. T. Courtois, M. Grajek and R. Naik, Optimizing sha256 in bitcoin mining, in Cryptography and Security Systems - Third International Conference, CSS 2014, Proceedings, 2014, 131–144.

[10] C. Decker and R. Wattenhofer, Information propagation in the bitcoin network, in IEEE P2P 2013, Proceedings, 2013, 1–10.

[11] I. Eyal, The miner’s dilemma, in 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17–21, 2015, 2015, 89–103.

[12] I. Eyal and E. G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, Communications of the ACM CACM Homepage Archive, 61 (2018), 95–102.

[13] B. A. Fisch, R. Pass and A. Shelat, Socially optimal mining pools, in International Conference on Web and Internet Economics, 2017, 205–218.

[14] J. Garay, A. Kiayias and N. Leonados, The bitcoin backbone protocol: Analysis and applications, in Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II, 9057 (2015), 281–310.

[15] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun and Y. Li, Cost-efficient strategies for restraining rumor spreading in mobile social networks, IEEE Transactions on Vehicular Technology, 66 (2017), 2789–2800.

[16] Z. He, Z. Cai and X. Wang, Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks, in IEEE ICDCS, 2015, 205–214.

[17] Z. He, Z. Cai and J. Yu, Latent-data privacy preserving with customized data utility for social network data, IEEE Transactions on Vehicular Technology, 67 (2018), 665–673.

[18] E. Heilman, One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner, in Financial Cryptography and Data Security - FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers, 2014, 161–162.

[19] C. Hu, W. Li, K. Cheng, J. Yu, S. Wang and R. Bie, A secure and verifiable access control scheme for big data storage in clouds, IEEE Transactions on Big Data (Early Access), 4 (2018), 341–355.

[20] Q. Hu, S. Wang, H. Hu, J. Huang, W. Li and X. Cheng, Messages in a concealed bottle: Achieving query content privacy with accurate location-based services, to appear in IEEE Transactions on Vehicular Technology, 67 (2018), 7698–7711.

[21] G. O. Karame, E. Androulaki and S. Capkun, Double-spending fast payments in bitcoin, in the ACM Conference on Computer and Communications Security, CCS’12, Raleigh, NC, USA, October 16-18, 2012, 2012, 906–917.

[22] A. Kothapalli, A. Miller and N. Borisov, Smartcast: An incentive compatible consensus protocol using smart contracts, in International Conference on Financial Cryptography and Data Security, 2017, 536–552.

[23] J. A. Kroll, I. C. Davey and E. W. Felten, The economics of bitcoin mining, or bitcoin in the presence of adversaries, in Proceedings of WEIS, 2013 (2013), p. 11.

[24] W. Li, M. Larson, C. Hu, R. Li, X. Cheng and R. Bie, Secure multi-unit sealed first-price auction mechanisms, Security and Communication Networks 11 on Cyber Security, Crime, and Forensics of Wireless Networks and Applications, 9 (2016), 3833–3843.

[25] Y. Liang, Z. Cai, J. Yu, Q. Han and Y. Li, Deep learning based inference of private information using embedded sensors in smart devices, IEEE Network Magazine, 32 (2018), 8–14.

[26] Y. Liang, Z. Cai, Q. Han and Y. Li, Location privacy leakage through sensory data, Security and Communication Networks, 2017 (2017), Article ID 7576307, 12 pages.
[27] L. Luu, R. Saha, I. Parameshwaran, P. Saxena and A. Hobor, On power splitting games in distributed computation: The case of bitcoin pooled mining, in IEEE 28th Computer Security Foundations Symposium, CSF 2015, Verona, Italy, 13-17 July, 2015, 2015, 397–411.

[28] L. Luu, Y. Velner, J. Teutsch and P. Saxena, Smart pool: Practical decentralized pooled mining, in 26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017, 2017, 1409–1426. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/luu

[29] R. Meni, A short note about variance and pool payouts, [Online], 2011, Available: https://bitcointalk.org/index.php?topic=5264.0

[30] C. Mooney and S. Mufson, Why the bitcoin craze is using up so much energy, 2018, [Online]. Available: https://www.washingtonpost.com/news/energy-environment/wp/2017/12/19/why-the-bitcoin-craze-is-using-up-so-much-energy/?utm_term=.2ec154762efa

[31] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, [Online], 2009, Available: http://www.bitcoin.org/bitcoin.pdf

[32] K. Nayak, S. Kumar, A. Miller and E. Shi, Stubborn mining: Generalizing selfish mining and combining with an eclipse attack, in IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016, 2016, 305–320.

[33] M. Rosenfeld, Analysis of bitcoin pooled mining reward systems, CoRR, vol. abs/1112.4980, 2011. [Online]. Available: arXiv:1112.4980

[34] M. Rosenfeld, Mining pools reward methods, presentation at bitcoin 2013 conference, [Online]. Available: http://www.youtube.com/watch?v=5sgdD4mGPfg

[35] M. Rosenfeld, Double geometric method: Hopping-proof, low-variance reward system, [Online], Available: https://bitcointalk.org/index.php?topic=39497.0

[36] A. Sapirshtein, Y. Sompolinsky and A. Zohar, Optimal selfish mining strategies in bitcoin, in Financial Cryptography and Data Security - 20th International Conference, FC 2016, Christ Church, Barbados, February 22-26, 2016, Revised Selected Papers, 9603 (2017), 515–532.

[37] O. Schrijvers, J. Bonneau, D. Boneh and T. Roughgarden, Incentive compatibility of bitcoin mining pool reward functions, in Financial Cryptography and Data Security - 20th International Conference, FC 2016, Christ Church, Barbados, February 22-26, 2016, Revised Selected Papers, 9603 (2017), 477–498.

[38] B. L. Shultz and D. Bayer, Certification of witness: Mitigating blockchain fork attacks, Undergraduate Thesis in Mathematics, Columbia University in the City of New York, 2015.

[39] S. Solat and M. Potop-Butucaru, Brief announcement: Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin, in Stabilization, Safety, and Security of Distributed Systems - 19th International Symposium, SSS 2017, Boston, MA, USA, November 5-8, 2017, Proceedings, 2017, 356–360.

[40] E. Swanson, Bitcoin mining calculator, 2013, [Online], Available: https://alloscomp.com/bitcoin/calculator/

[41] D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat and L. Njilla, Security implications of blockchain cloud with analysis of block withholding attack, in 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID, 2017, 458–467.

[42] Y. Velner, J. Teutsch and L. Luu, Smart contracts make bitcoin mining pools vulnerable, in Topics in Cryptology - CT-RSA 2017 - The Cryptographers’ Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings, 10159 (2017), 277–292.

[43] X. Zheng, Z. Cai, J. Yu, C. Wang and Y. Li, Protecting query privacy with differentially private k-anonymity in location-based services, Personal and Ubiquitous Computing, 2018, 1–17.

[44] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper, 151 (2014), 1–32.

[45] R. Zhang and B. Preneel, Publish or perish: A backward-compatible defense against selfish mining in bitcoin, in Topics in Cryptology - CT-RSA 2017 - The Cryptographers’ Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings, 10159 (2017), 277–292.

[46] X. Zheng, Z. Cai, J. Yu, C. Wang and Y. Li, Follow but no track: Privacy preserved profile publishing in cyber-physical social systems, IEEE Internet of Things Journal, 4 (2017), 1868–1878.
[48] X. Zheng, Z. Cai, J. Li and H. Gao, Location-privacy-aware review publication mechanism for local business service systems, in IEEE INFOCOM, Atlanta, USA, 2017, 1–9.

[49] X. Zheng, G. Luo and Z. Cai, A fair mechanism for private data publication in online social networks, IEEE Transactions on Network Science and Engineering (Early Access), 2018, 1–1.

[50] Y. Zolotavkin, J. García and C. Rudolph, Incentive compatibility of pay per last N shares in bitcoin mining pools, in Decision and Game Theory for Security - 8th International Conference, GameSec, 2017, 21–39.

[51] Blockchain info: Detailed information and charts on all bitcoin transactions and blocks, [Online], Available: https://blockchain.info/

[52] Bitcoin block reward halving countdown, [Online], Available: https://www.bitcoinblockhalf.com/

[53] Comparison of mining pools, [Online], Available: https://en.bitcoin.it/wiki/Comparisonofminingpools

[54] The cryptocurrency for payments based on blockchain technology, [Online], Available: https://litecoin.org/

[55] How to destroy bitcoin with 51% (pocket guide for governments), [Online], Available: https://medium.com/@homakov/how-to-destroy-bitcoin-with-51-pocket-guide-for-governments-83d9bf12e6

[56] Pps vs pplns – mining pool payment reward structure explained, [Online], 2018, Available: https://coinguides.org/pps-vs-pplns/

[57] Slush pool reward system, [Online], Available: https://slushpool.com/help/manual/rewards

Received August 2018; revised October 2018.

E-mail address: szhu5@student.gsu.edu
E-mail address: wli28@gsu.edu
E-mail address: lihong@iie.ac.cn
E-mail address: chu@cqu.edu.cn
E-mail address: zcai@gsu.edu