Doppelganger in Bitcoin Mining Pools: An Analysis of the Duplication Share Attack

Yujin Kwon, Dohyun Kim, Yunmok Son, Jaeyeong Choi, and Yongdae Kim

Korea Advanced Institute of Science and Technology (KAIST),
291 Daehak-ro, Daejeon, Republic of Korea
{dbwls8724,dohyunjk,yunmok00,go1736,yongdaek}@kaist.ac.kr

Abstract. Bitcoin is a cryptocurrency based in peer-to-peer network that uses a blockchain. To maintain the blockchain without trusted third parties, a player called a *miner* proves that he has completed a *proof-of-work*. As the difficulty of *proof-of-work* is increasing, *mining pools*, consisting of a number of miners, have become major players compared with solo miners. Most mining pools consist of a manager and miners. All miners who belong to a mining pool submit their *shares* to the manager and get paid in proportion to the amount of their *shares*. Therefore, the manager has to pay all miners fairly.
However, many Bitcoin mining pools were ruined by an attack called the Duplicate Share Attack (DSA) in 2015. In this paper, we analyze DSA in multiple directions. First, we mathematically analyze DSA against one mining pool and multiple mining pools. As results of our analyses, we derive the optimal attacker's strategy, which shows that DSA can give a large extra profit to an attacker with little computational power. Because the duplicate share vulnerability has been already fixed in a few large mining pools after DSA was introduced, DSA may not be considered a threat any more. However, we show that several small mining pools are still vulnerable to DSA and an attacker can unfairly earn a large extra profit using these unpatched small mining pools. In summary, we argue that honest miners in Bitcoin network are not yet free from DSA.

Keywords: Bitcoin, mining pool, duplicate share, attacker strategy

1 Introduction

Bitcoin is a decentralized cryptocurrency with SHA-256, developed by Satoshi Nakamoto in 2008 [14]. Different from traditional currencies that rely on trusted third parties, Bitcoin uses a peer-to-peer network because of preventing single point of failure. Since Bitcoin's invention, it has been popular, and the current price of 1 BTC (i.e., unit of Bitcoin) has been evaluated to be about 570 USD [7].

In Bitcoin, each exchange of Bitcoin creates a data structure, called a *transaction*. Transactions that are generated in a specific time interval are embedded in a larger data structure, called a *block*. In the header of a block, information on transactions that are included in the block is also stored using a Merkle tree [17]. In addition, by using a hash chain to headers of all blocks (i.e., a blockchain), the

integrity of all blocks can be maintained. To maintain the blockchain without trusted third parties, every block is broadcast to and stored in every peer in a Bitcoin network.

The header of a block includes the Merkle hash value of all transactions in the block, a random number called a *nonce*, and the hash value of the previous block's header. Among these, the nonce is very important in generating a legitimate block. The nonce can be obtained by solving a cryptographic puzzle. On average, every 10 min among the whole Bitcoin network, the puzzle is solved once. Because, for a specific 10 min, all *miners* try to solve the same puzzle, the process for calculate a nonce, called *mining*, is competitive. Additionally, miners that participate in mining take a specific amount of Bitcoin as the subsidy (12.5 BTC in August, 2016) in return for providing computational power to find the nonce.

As the difficulty of the cryptographic puzzle has increased, many *mining pools* have been organized by grouping individual miners. Each miner submits the results of his computation, called a *share*, to the manager of the pool. Then, the manager checks submitted shares and pays the corresponding miners in the pool according to the amounts of their submissions.

However, the payment can be maliciously fabricated by a tricky miner to make unfair profits. In 2015, a technical problem that allows exploitation of duplicated shares was reported [3]. This problem permits an attacker to submit duplicated or crafted shares and be unfairly rewarded for them while providing only a part of her computational power to a mining pool (and, in fact, using the other part of her computational power for solo mining).

In this paper, to the best of our knowledge, we first analyzed the duplicate share attack (DSA) mathematically when a target is one pool. The results of our analyses derive the optimal strategy for maximizing an attacker's profit. The strategy is to send as many duplicate shares as possible while using minimum power. Moreover, if the pool's power is 25%, an attacker can earn a global maximum profit 12.49% of the total 12.5 BTC subsidy with our strategy. We also expand our strategy against one pool to multiple pools. The expanded strategy increases the attacker's profit from the case in which the target is a single pool. To show the feasibility of DSA, we found that multiple small mining pools are still vulnerable to DSA, unlike existing popular ones. According to our analyses, an attacker can earn 4.997% of the total 25 BTC subsidy by trying DSA in multiple small mining pools with a minimum power of 0.24%. This result shows that DSA is more efficient than previous attacks. Therefore, we prove that the Bitcoin system is still unsafe against DSA because of these vulnerable small mining pools. Our contributions are as follows.

- We mathematically analyze DSA for one mining pool and derive the optimal attacker strategy.
- We expand our strategy to target multiple mining pools for maximum profit.
- At the time of submission of this paper, we found that several small mining pools have not patched this vulnerability. We, prove then, that DSA can still

allow an attacker to profit unfairly against multiple small, as yet unpatched, mining pools.

This rest of the paper is organized as follows: Section 2 provides background knowledge, particularly of mining pools and DSA. Section 3 describes the detailed analyses for DSA against one pool and multiple pools. In addition, the feasibility of our analyses is explained in Section 4. Existing related works and our conclusion are presented in Section 5 and 6, respectively.

2 Background

In this section, we describe the mining process in a mining pool and the technical problem related to DSA introduced in 2015.

2.1 Mining Process in a Mining Pool

Most mining pools consist of a manager and miners aiming to solve a cryptographic puzzle effectively. To understand this puzzle, it is first necessary to know the components of a *block header*. A block header contains the Merkle hash value of all *transactions* in the *block*, a random number called a *nonce*, and the hash value of the previous block's header. For a given 256-bit number a, the miner tries to find a valid nonce that makes the hash value of the block header smaller than a [1]. Therefore, the probability to find the nonce is $\frac{a}{2^{256}}$; the process of solving this puzzle is called *proof-of-work* [5]. In mining pools, the difficulty of this puzzle can be adjusted more easily. In other words, in order to incentivize the miners to work in pools, a manager can choose to solve this problem using a divide and conquer methodology, by using another 256-bit number b which is larger than a.

Every miner (i.e., honest miner) who belongs to the pool tries to find a valid nonce for target b given by the pool. If one of the miners finds a nonce, he would submit the nonce for his profit to the manager of the mining pool. This nonce is submitted using the Stratum protocol [6, 16] which defines a data structure called a *share* for the submission. In general, the Stratum protocol is implemented differently for every mining pool, but the share commonly has five parameters: miner's name, share ID, extra-nonce, current time, and nonce. After a share is submitted, the manager checks it to run the mining pool fairly (e.g., preventing duplicated shares). This checking process is also defined in the mining pool's Stratum protocol, and thus any misimplementation of the protocol can become problematic.

2.2 DSA

In DSA, miners unfairly earn greater profits by submitting duplicate shares to their mining pool manager. This was first noted in Bitcoin Forum [2] in 2015 because it can affect many mining pools [3].

[1] Note that for Bitcoin a SHA256-based Hash function is used.

In the Stratum protocol, a mining pool has to detect duplicate shares in order to prevent an attacker from getting paid an unfair profit. Therefore, when an attacker sends a duplicate share to the mining pool, the duplicate share is rejected with a "duplicate share" error. However, the detection processes of many mining pools did not distinguish capital and lowercase letters in data represented as hexadecimal characters (e.g., 0x`A` versus 0x`a`). As a result, an attacker could submit duplicate shares by replacing a capital letter with a lowercase letter or vice versa in three parameters of a share: the extra-nonce, current time, and nonce, which are 32-bit hexadecimal values, without "duplicate share" error. For example, nonce 0x`01abcdef` and 0x`01Abcdef` are regarded as different nonce even if they are the same value. Using DSA, the attacker can submit on average $(\frac{22}{16})^{24} \approx 2085$ duplicate shares per share.

Currently, the DSA problems of many mining pools have been fixed by replacing all capital letters with lowercase ones in those three parameters in submitted shares before managers check whether shares are duplicate shares. However, some small mining pools have not fixed the problem yet.

3 Mathematical Analysis of DSA

In this section, we mathematically analyze DSA and derive an attack strategy for the maximum profit considering two conditions: first, when an attacker makes an attempt at DSA in only one mining pool and second, in multiple pools. In this paper, *profit* is defined as how much Bitcoin miners can gain in one round on average and *power* refers to the computational power of miners or mining pools which is represented as a relative value between 0 and 1 compared to the total computational power of a Bitcoin network.

In our analysis, we assume as follows.

1. The profit of a solo miner or a mining pool is proportional to their computational power.
2. A manager pays its profit to each miner who belongs to the pool proportionally to his computational power that is used for the pool.
3. The computation power of a mining pool is the sum of all miners in the pool.
4. For simplicity, we also assume that the subsidy is 1 BTC instead of 25 BTC. Therefore, the amount of profit of a mining pool or a solo miner is the value of one's computation power.
5. All managers and miners other than the attacker are honest.

Before analyzing DSA, it is required to show how much an honest miner (without DSA) earns when he divides his power as a solo miner and a participant in a mining pool. If the mining pool and the honest miner have the power α and β, respectively, and the honest miner contributes his power with γ ratio to the mining pool (i.e., the power of the honest miner that is used for the mining pool is $\gamma\beta$), then the profit of the mining pool is $\alpha + \gamma\beta$. Therefore, the honest miner earns not only $\frac{\gamma\beta}{\alpha+\gamma\beta}$ of the pool reward P_{pool} but also $(1-\gamma)\beta$ (P_{solo}) because he can use the rest of his power as solo miner. Finally, his total profit P_h is

$$P_h = P_{solo} + P_{pool} = (1-\gamma)\beta + \frac{\gamma\beta}{\alpha+\gamma\beta} \cdot (\alpha+\gamma\beta) = \beta.$$

This shows that the profit of an honest miner is can still earn profit proportionally to his mining power β. In other words, in a fair and true payoff system, miners' profits are proportional to their own computational power without considering any attack. In this paper, therefore, the goal of an attacker is to earn more profit than an honest miner's profit (i.e., β) by gaining unfair advantage against the mining pool and other honest miners.

3.1 For One Mining Pool

For simplicity, we first analyze the case in which a miner attacks only one pool. The power of a mining pool before an attacker's participation and an attacker are α and β, respectively. If an attacker uses γ ratio of her power in the pool, mining pool power after her participation and her solo mining power are $\alpha+\gamma\beta$ and $(1-\gamma)\beta$ each. We assume that she submits duplication shares k times. Then, she receives $\frac{k\gamma\beta}{\alpha+k\gamma\beta}$ ratio of pool's total profit, and the attacker's total profit P is

$$\begin{aligned} P = P_{solo} + P_{pool} &= (1-\gamma)\beta + \frac{k\gamma\beta}{\alpha+k\gamma\beta} \cdot (\alpha+\gamma\beta) \\ &= \beta - \gamma\beta + \frac{k\gamma\beta}{\alpha+k\gamma\beta} \cdot (\alpha+\gamma\beta) \\ &= \beta + f(k,\gamma,\beta). \end{aligned}$$

when

$$f(k,\gamma,\beta) = \frac{\alpha\gamma\beta}{\alpha+k\gamma\beta} \cdot (k-1). \quad (k \geq 1) \tag{1}$$

Equation (1) is an increasing function of k. Therefore, an attacker should copy a share as many as possible to get the maximum profit. If she ideally chooses k as infinity, her total profit P will be

$$\lim_{k\to\infty} P = \alpha + \beta.$$

In other words, an attacker is ideally able to get the maximum $\alpha+\beta$ profit by submitting the same share infinitely. However, k cannot be increased to be infinity in reality because of network delay between the mining pool and an attacker and the possible range of k in technical problem described in Section 2. Specifically, the maximum practical k is 2085.

After determining k, an attacker should determine γ in order to get maximal profit. We assume that the an attacker increases his power for mining in a pool as fake, restricting the fake power of the pool up to 50% of the total substantive power to avoid suspicion of the manager. Hence, γ has to satisfy following condition.

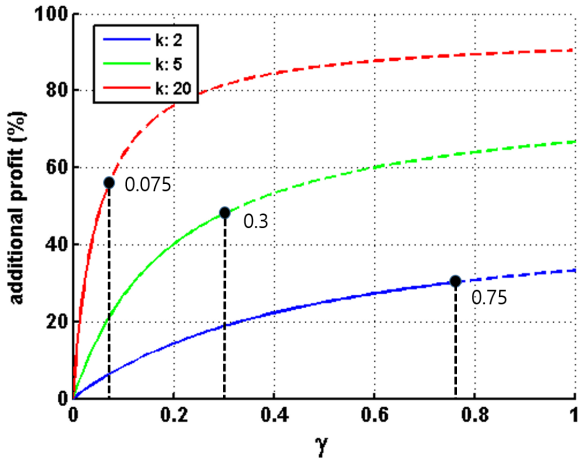

Fig. 1: The x- and y-axes represent γ and %-additional profit, respectively. The percentage additional profit is defined an extra gain compared with the profit P_h of a honest miner and represented as $\frac{P-P_h}{P_h} \cdot 100$. It is also called earnings rate. This figure shows that profit P is an increasing function of γ. Additionally, 0.75, 0.3, and 0.075 are the maximal γ values by Equation (4) when k is 2, 5, and 20, respectively.

$$k\gamma\beta \leq \min(\frac{1}{2} - \alpha, k\beta) \tag{2}$$

By this condition, an attacker has to choose

$$\beta \geq \frac{1-2\alpha}{2k} \tag{3}$$

$$\gamma = \frac{1-2\alpha}{2k\beta} \tag{4}$$

so as to earn maximal profit, because profit P is an increasing function of γ, as in Figure 1. The figure represents the percentage additional profit according to γ when α and β are 0.2. The percentage additional profit is the relative extra profit compared with profit P_h earned by an honest miner, and it is expressed as $\frac{P-P_h}{P_h} \cdot 100$.

Then, an attacker earns profit

$$\beta + \frac{(k-1)\alpha(1-2\alpha)}{k}. \tag{5}$$

The second term of Equation (5) represents extra gain compared to fair profit P_h. It is affected only by the pool's power α, regardless of the attacker's power β. As a result, an attacker can earn a large extra gain with minimal power

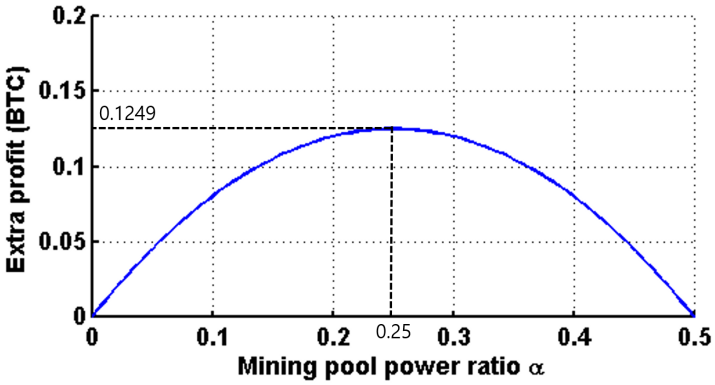

Fig. 2: The x- and y-axes are α and extra profit in BTC, respectively. The extra profit reaches a global maximal of 0.1249 BTC when α is 0.25.

$$\frac{1-2\alpha}{2k}$$

expenditure and, percentage additional profit can increase up to $2(k-1)\alpha \times 100\%$. For example, we assume that one pool has the technical problem described in Section 2.2 and the pool's power is 0.2. First, an attacker will choose her maximum k as 2085 per our strategy. Then she needs minimum power 1.4388×10^{-4} by Equation (3) and can earn maximum 0.1199 BTC additionallly. Also, her percentage additional profit will be 83,360 % (i.e., approximately 834 times larger) when she uses minimum power 1.4388×10^{-4}.

Moreover, we illustrate Figure 2 to show the tendency of maximum extra profit according to a pool's power. Figure 2 shows that extra profit is the global maximum when α is 0.25. Therefore, if an attacker tries DSA with minimum computational power in a pool that has 0.25 computational power, she can earn the global maximum extra profit.

3.2 For Multiple Mining Pools

Second, we expand the attacker's strategy from targeted one pool to targeted multiple pools in this section. We assume that the attacker joins in multiple mining pools (pool 1, 2, ... , n), and notations are defined as follows:

α_i : Power of mining pool i
β : Attacker's power
γ_i : The ratio between her power consumed in mining pool i and α_i.

Then her profit $P_{pool\,i}$ gained from pool i is

$$\frac{k\gamma_i\beta}{\alpha_i + k\gamma_i\beta} \cdot (\alpha_i + \gamma_i\beta)$$

if she duplicates a share k times in pool i. She also earns $(1 - \sum_{i=1}^{n} \gamma_i)\beta$ on average by solo mining. Therefore, her total profit P is

$$\begin{aligned} P &= (1 - \sum_{i=1}^{n} \gamma_i)\beta + \frac{k\gamma_i\beta}{\alpha_i + k\gamma_i\beta} \cdot (\alpha_i + \gamma_i\beta) \\ &= \beta + \sum_{i=1}^{n} f(k, \gamma_i, \beta). \end{aligned}$$

when function f is as defined in Equation (1). Then, we can apply the DSA strategy described in Section 3.1 to all mining pools i because the maximization of the attacker's profit in every mining pool is equivalent to maximizing her total profit in the case that targets are multiple pools, as is Equation (6)

$$\begin{aligned} &\arg\max_{k,\gamma} \beta + \sum_{i=1}^{n} f(k, \gamma_i, \beta) \\ =&\arg\max_{k,\gamma_i} f(k, \gamma_i, \beta)\ (i = 1 \sim n\,) \\ =&\arg\max_{k,\gamma_i} \beta + f(k, \gamma_i, \beta)\ (i = 1 \sim n\,). \end{aligned} \tag{6}$$

Therefore, an attacker can choose as large k as possible and use the minimal computational power

$$\sum_{i=1}^{n} \frac{1 - 2\alpha_i}{2k} \tag{7}$$

according to Equation (3) for DSA. Second, she divides her computational power into

$$\gamma_i = \frac{1 - 2\alpha_i}{2k\beta}$$

for pool i so as to get the maximum extra profit. Then she can earn the maximum extra profit

$$\sum_{i=1}^{n} \frac{(k-1)\alpha(1-2\alpha)}{k}.$$

For example, if three mining pools are vulnerable, an attacker can try to apply DSA against three mining pools. If the computational powers of the three mining pools are 0.1, 0.05, and 0.05, respectively, she has to choose k as 2085 and prepare the minimum power 6.235×10^{-4} by Equation (7). Additionally, she divides her power into 0.3077, 0.3462, and 0.3462, for each of pools 1, 2, and 3 according to our strategy. Then, the attacker can earn the maximum extra profit

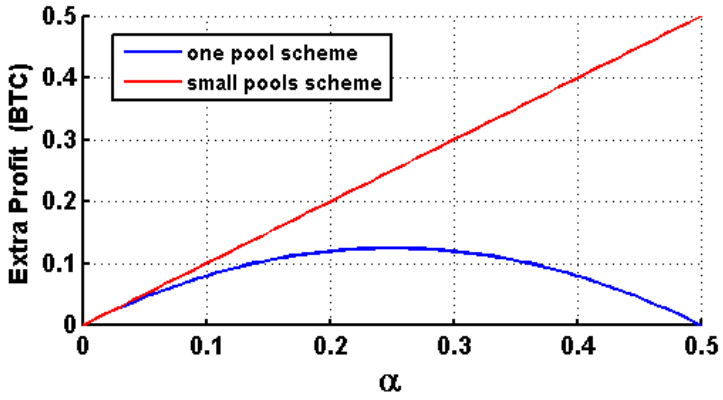

Fig. 3: The x-axis is the total power of target mining pools and the y-axis is additional profit in BTC. The red and blue lines mean extra profits when the target is small pools and one pool, respectively.

of 0.1699 BTC. Note that this profit is larger than the 0.1199 BTC that are the maximum extra profit when attacking against one pool with a computational power of 0.2. As a result, we show that an attack against multiple pools is more profitable than one against a single pool, even if the total power of the targets is the same.

4 Feasibility of DSA

At present, most of the popular mining pools have patched the DSA vulnerability described in Section 2. Therefore, many people may believe that an attacker cannot earn unfairly a large extra profit but a small extra profit by DSA. However, several mining pools still have this problem, though they are unpopular and have relatively little computing power (i.e., less than 1 % of the whole power of the Bitcoin network). For this, we reviewed programs for bitcoin in the Github repository and discovered that at least 28 projects still have the DSA vulnerability. With this, one interesting question is if one can get a significantly large extra profit by utilizing DSA against these small, as yet unpatched mining pools. Before analyzing the extra profit from attacking multiple small mining pools, we first recall the maximum additional profit that an attacker can gain through DSA against one pool with power ratio α:

$$\frac{(k-1)\alpha(1-2\alpha)}{k}. \tag{8}$$

Further, when the target is n pools and each pool i has power α_i, recall that the maximum additional profit is

Table 1: Each mining pool's computational power in the last three years (from May, 2013 to May, 2016). We calculated the distribution of their computational powers from [4]. († includes small mining pools and solo miners.)

Mining Pool	**Computational Power (%)**	**Mining Pool**	**Computational Power (%)**
F2Pool	19.72 %	Eligius	6.27 %
GHash.IO	16.27 %	Slush	5.96 %
AntPool	11.93 %	BW.COM	4.24 %
BitFury	7.77 %	KnCMiner	1.97 %
BTCC Pool	6.71 %	The others†	19.16 %

$$\sum_{i=1}^{n} \frac{(k-1)\alpha_i(1-2\alpha_i)}{k}. \tag{9}$$

Because all α_i (< 0.01) are small, Equation (9) is approximated to

$$\frac{k-1}{k}\sum_{i=1}^{n}\alpha_i. \tag{10}$$

Equation (10) is always greater than the Equation (8), in concurrence with what we see in Figure 3. The x- and y-axes of Figure 3 represent targeted pools' total power ratio and the attacker's extra profit in BTC, respectively. The scheme for small pools is always more efficient than the scheme for one pool even if the targets' total power is the same as the power of the single target pool.

To show the impact of DSA, we estimate the attacker's extra profit in current pools' computational power distribution according to our strategy. Table 1 is mining pools' power distributions for recent three years and, *the others* includes small pools and solo miners. Assume that the attacker's targets are a set of unpatched small mining pools possibly belonging to *the others*. Further, we assume that the number of vulnerable small pools is 28 and the total computing power of the set is 5 % (i.e., $\sum_{i=1}^{28}\alpha_i = 0.05$). In this case, by applying Equation (10) with these numbers, we can conclude that an attacker can unfairly earn a maximum extra profit of 0.04997 BTC by using a minimum power 0.0067.

We note that DSA is significantly more efficient than other previous attacks. Rosenfeld [15] analyzed the pool-hopping attack and claimed that an attacker can earn 0.02815 BTC more in profit than an honest miner by using a computational power 0.1. Additionally, Luu et. al. [13] showed that an attacker's extra profit can be 0.0123 BTC by performing a block withholding attack with a computational power 0.25. In contrast, the DSA attack against a set of small mining pools can let the attacker obtain 0.04997 BTC by using a computational power 0.0067. Therefore, we can conclude that DSA against small mining pools is the strongest attack compared to previous ones.

5 Related Work

In recent years, a number of papers have studied the security of the Bitcoin world, such as double spending for fast payment, anonymity, and selfish mining. Particularly, attacks and a competition among mining pools also have been studied, as pools have become major players in the Bitcoin world.

An attacker can perform selfish mining in a mining pool, which means that the attacker unfairly receives more pay from the pool manager [1, 10]. Rosenfeld [15] introduced the most widely known attack to mining pools, called a block withholding attack (BWH). The attack is that miners in the pool do not submit valid a nonce which makes a block legitimate, so as to degrade the mining pool's power. The author argued that an attacker does not earn any benefit from the pool by the BWH attack. However, Courtois et. al. [8] proved that the BWH attack allows an attacker to earn more profit in the long term, proposing a practical BWH attack that generalizes the BWH attack introduced by Rosenfeld [15]. Moreover, Luu et. al. [13] found an optimal strategy of an attacker, focusing on splitting his power into several mining pools in order to get a maximum profit. At the same time, Eyal [9] introduced a notion called the miner's dilemma, which refers to the decision of whether or not to perform a BWH attack in a game between two pools and estimated a BWH attack's effect on two pools. We do not consider BWH attacks in this paper.

The competition among mining pools has been analyzed based on game theoretic models [8, 12]. Johnson et. al. [11] considered a competition between two mining pools which can make an attempt at Distributed Denial-of-Service (DDoS) attacks each other. They claimed that mamagers of two pools can choose mischievous tactics, such as triggering a DDoS attack to lower a competing pool's mining power, and compared the trade-off between mischievous tactic and the benign tactic, which is to perform honest mining in the Bitcoin network, under game-theoretic analysis. Meanwhile, we focus herein on the DSA attack of an miner against mining pools instead of the relation between mining pools.

6 Conclusion

In 2015, DSA introduced in Bitcoin Forum was caused by an implementation problem in many mining pools. In this paper, we first derive the optimal strategy of an attacker for DSA mathematically. Our results show that, considering multiple small unpatched mining pools, an attacker can gain a large amount of unfair profit using minimal computational power by DSA using our optimal strategy. Therefore, we conclude that, as ever, DSA is a practical threat to Bitcoin systems, even if many major mining pools have patched the vulnerability. We also argue that managers or developers of existing mining pools must patch the problem in their Stratum protocols as soon as possible.

Acknowledgments. This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.B0717-16-0116, Development of information leakage

prevention and ID management for secure drone services) and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R-20160222-002755, Cloud based Security Intelligence Technology Development for the Customized Security Service Provisioning)

References

1. Bahack, L.: Theoretical Bitcoin Attacks with Less than Half of the Computational Power (draft). arXiv preprint arXiv:1312.7013 (2013)
2. Bitcoin Forum: Bitcoin Forum (2016), https://bitcointalk.org/, [Online; accessed 05-June-2016]
3. Bitcoin Forum: Duplicate Shares Exploit – Most Pools Affected (2016), https://bitcointalk.org/index.php?topic=1065576.0, [Online; accessed 05-June-2016]
4. BitcoinChain: Bitcoin Mining Pools (2016), https://bitcoinchain.com/pools, [Online; accessed 05-June-2016]
5. bitcoinwiki: Proof of Work (2016), https://en.bitcoin.it/wiki/Proof_of_work, [Online; accessed 05-June-2016]
6. bitcoinwiki: Stratum Mining Protocol (2016), https://en.bitcoin.it/wiki/Stratum_mining_protocol, [Online; accessed 05-June-2016]
7. CoinDesk: BITCOIN PRICE INDEX CHART (2016), http://www.coindesk.com/price/, [Online; accessed 05-June-2016]
8. Courtois, N.T., Bahack, L.: On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency. arXiv preprint arXiv:1402.1718 (2014)
9. Eyal, I.: The Miner's Dilemma. In: Security and Privacy (SP), 2015 IEEE Symposium on. pp. 89–103. IEEE (2015)
10. Eyal, I., Sirer, E.G.: Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In: Financial Cryptography and Data Security, pp. 436–454. Springer (2014)
11. Johnson, B., Laszka, A., Grossklags, J., Vasek, M., Moore, T.: Game-Theoretic Analysis of DDoS Attacks Against Bitcoin Mining Pools. In: Financial Cryptography and Data Security, pp. 72–86. Springer (2014)
12. Laszka, A., Johnson, B., Grossklags, J.: When Bitcoin Mining Pools Run Dry: A Game-Theoretic Analysis of the Long-Term Impact of Attacks Between Mining Pools. In: In BITCOIN15: The Second Workshop on Bitcoin Research. Citeseer (2015)
13. Luu, L., Saha, R., Parameshwaran, I., Saxena, P., Hobor, A.: On Power Splitting Games in Distributed Computation: The Case of Bitcoin Pooled Mining. In: Computer Security Foundations Symposium (CSF), 2015 IEEE 28th. pp. 397–411. IEEE (2015)
14. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
15. Rosenfeld, M.: Analysis of Bitcoin Pooled Mining Reward Systems. arXiv preprint arXiv:1112.4980 (2011)
16. SLUSH POOL: Stratum Mining Protocol (2016), https://slushpool.com/help/#!/manual/stratum-protocol, [Online; accessed 05-June-2016]
17. Wikipedia: Merkle tree — wikipedia, the free encyclopedia (2016), https://en.wikipedia.org/w/index.php?title=Merkle_tree&oldid=720708959, [Online; accessed 05-June-2016]