A Poisoning Attack Against Cryptocurrency Mining Pools

Mohiuddin Ahmed, Jinpeng Wei(✉), Yongge Wang, and Ehab Al-Shaer

University of North Caroline at Charlotte, Charlotte, NC 28223, USA
{mahmed27,jwei8,yonwang,ealshaer}@uncc.edu

Abstract. This paper discusses a potentially serious attack against public crypto-currency mining pools. By deliberately introducing errors under benign miners' names, this attack can fool the mining pool administrator into punishing any innocent miner; when the top miners are punished, this attack can significantly slow down the overall production of the mining pool. We show that an attacker needs only a small fraction (e.g., one millionth) of the resources of a victim mining pool, which makes this attack scheme very affordable by a less powerful competing mining pool. We experimentally confirm the effectiveness of this attack scheme against a few well-known mining pools such as Minergate and Slush Pool.

Keywords: Crypto currency · Mining pool · Invalid share
DoS attack · Stratum protocol

1 Introduction

The emergence of bitcoin in 2009 [24] paves the way for many other crypto-currencies like monero, litecoin, and etherium. Crypto-currency is the cryptographically protected digital currency that is built upon the blockchain platform. Blockchain is like a public ledger that keeps track of all transactions in each crypto-currency. Blockchain is shared across all users of a specific crypto-currency. Before adding any transaction to the blockchain, the transaction needs to be verified, which is called *mining*. The person or group of people who is verifying the transaction is called miner. For the verification of a transaction, the miner receives a reward in the form of crypto-currency where the mining is performed.

In order to verify a transaction, miners have to solve a cryptographic puzzle. Since the task of solving the cryptographic puzzle is computation-intensive, the more computation power a miner has, the more likely he/she can solve the puzzle. If miners follow mining protocol honestly, they can increase their mining power in two ways: (1) *solo mining*, in which a miner can buy new resources and deploy those to mine transactions, and (2) *pool mining*, in which miners form a pool and share their resources to solve the cryptographic puzzle. In pool mining, the reward is distributed among the pool members based on their contributions. Solo

J. Garcia-Alfaro et al. (Eds.): DPM 2018/CBT 2018, LNCS 11025, pp. 140–154, 2018.
https://doi.org/10.1007/978-3-030-00305-0_11

mining is almost obsolete now due to the increasing difficulty of crypto-currency mining and the emergence of task specific hardware like ASICs. Instead, pool mining has become a more promising way of mining due to the trade-off between revenue gain and power usage by resources.

One important design issue of pool mining is the accurate measurement of member contributions. To achieve this, two methods have been proposed: Proof of Work (PoW) and Proof of Stake (PoS). PoW has been used in bitcoin system to reach consensus on the blockchain status and to defend against double-spending attacks: each worker's computational power is calculated based on the *shares* he submits to the pool. Under PoS, each pool member's capability of creating the next block is proportional to the amount of *coin ages* he has, and the coin age is defined as currency amount times holding period [11,20].

The reward provided by the crypto-currency network encourages miners to increase their mining power through illegal means, represented in multiple kinds of attacks. One such way is hijacking benign users' machines and using them to mine on behalf of the attacker, e.g., botnet mining [17] and drive-by-cryptocurrency mining [4]. A second major type of attack is DDoS [16]. There are two incentives to perform those DDoS attacks. Firstly, slowing down the mining task of a pool through DDoS attack might give an unfair decisive advantage to other pools to win the race for the next bundle of crypto-currency rewarded for verifying a transaction. Secondly, delayed operation of a pool may discourage future users to join victim pool and current users might leave the pool for a better one.

We propose a way of indirect DDoS attack on the mining or crypto-currency protocol or implementation. Although there have been some attacks [27] using implementation or protocol vulnerability of mining and crypto-currency, no previous work used indirect DDoS on the user and pool at the same time. Our attack is mainly focused on PoW (Proof of Work) based pool mining.

More specifically, we propose to degrade the productivity of a target mining pool by poisoning its members' mining results, which causes the pool server to penalize its benign miners. This attack is enabled by two factors: (1) a lack of miner authentication and (2) the invalid share policy of the mining pools. The first factor allows an attacker to submit invalid shares on behalf of a benign miner, and when the number of invalid shares reaches certain threshold, they trigger penalty or ban of benign members of the mining pool based on the second factor. Since the ban or penalty to benign pool members are imposed inadvertently by the pool manager, we consider our attack technique indirect and subtle. This attack can be employed by one mining pool to lower the expected success outlook of a competing mining pool. The essence of our attack is to turn the invalid shares policy of mining pools against themselves.

We make the following contributions:

- We propose a novel attack scheme that can fool the mining pools into punishing their productive members.
- We implement a prototype of attack tool that can submit a large number of invalid shares using the Stratum protocol.

– We evaluate the effectiveness of our attack against Slush Pool and Minergate.

The rest of this paper is organized as follows. Section 2 gives technical background information about pool-based crypo-currency mining and invalid share policy adopted by mining pools. Section 3 describes the details of our attack method. Section 4 presents both theoretical and empirical evaluation of our attack scheme against Minergate and Slush Pool. Section 5 discusses possible remediation of the attack. Related work is mentioned in Sect. 6, and Sect. 7 concludes the paper.

2 Technical Background

Most of the crypto-currencies currently available on the market are distributed and decentralized in nature. Those crypto-currency ecosystems consist of users, miners, blockchain, and mining pools. Users use crypto-currency in the form of transaction. Miners verify the transaction and append the verified transaction to the publicly shared ledger called blockchain. Miners are rewarded by the crypto-currency network for verifying each transaction, which gives them incentives.

2.1 Blockchain and Mining Pool

Blockchain is a public ledger that records all of the verified transactions. Miners add new transactions to the blockchain after verification. Verification of transaction is called mining, which is to solve a cryptographic puzzle. The cryptographic puzzle to solve is generating a hash that is smaller than a set value provided by the network. The set value is called *difficulty* of the network. For bitcoins, this difficulty value is adjusted dynamically such that blocks are generated at an average rate of one every ten minutes [1]. Different cryto-currency use different hashing algorithms. For example, bitcoin and bitcoincash uses SHA-256 hashing algorithm, Litecoin and Dogecoin uses Scrypt hashing algorithm, Dash (DASH) and CannabisCoin (CANN) uses X11 hashing algorithm, Monero and Bytecoin uses Cryptonight algorithm and, ethereum and ethereum classic use Ethash algorithm.

Since solving the cryptographic puzzle is a computation intensive task, it became increasingly difficult task for solo miners to solve the puzzle. To solve this problem mining pool has emerged. In a mining pool, all members work together to mine each block and share their revenues when one of them mines a block. Although joining a pool does not change a miner's expected revenue, it decreases the variance and makes the monthly revenues more predictable.

Popular mining pools consist of thousands of miners. For example, btc.com mining pool has around 56k workers for bitcoin and around 31k workers for bitcoin cash crypto-currency. The hashrate of this pool is 8.7 EH/s for bitcoin and 401.9 PH/s for bitcoin cash. Here, pool hashrate is the aggregation of all workers' hashrate. Each worker's hashrate is calculated based on the valid shares that he submits. For example, if a worker submits one share of difficulty one, it

means that this worker checks $2^{number\ of\ trailing\ zeros\ in\ target\ value}$ hash values to generate the valid share. Again, submitting one share of difficulty two is like submitting two shares of difficulty one.

Moreover, mining pools often offer variable share difficulty, in which the pool assigns share targets to miners adaptively based on their computational ability. The purpose of adaptive share difficulty is to make sure that the task is neither too difficult, thus enabling miners to prove work is done, nor too easy, thus reducing the overhead on the pool to verify submitted shares.

2.2 Cryptographic Puzzle

The cryptographic puzzle to solve is to find a hash using data from assigned job that is less than a provided *target value*. For bitcoin, the puzzle consists of a target value and a tuple, F = *(block version number || hash of previous block || root of merkle tree || timestamp || Nbits)*, here || denotes concatenation. *target* and all fields of tuple F will be proved in the assigned job. *Nbits* is the encoded share difficulty. *Extranonce2* is changed by the miners by incrementing it in addition to incrementing nonce, so that there are more possible hashes that can be tried with a given set of transactions. Given the target and tuple F, the miners will try to find a pair iterating over *Extranonce2* and *nonce* such that it satisfies the following equation

$$H^2(nonce||F) < target \tag{1}$$

Here, H^2 means double SHA-256 hashing operation for bitcoin.

2.3 Stratum-Mining Pool Communication Protocol

Stratum is a clear text communication protocol built over TCP/IP using JSON-RPC format [6]. Although there is no official documentation for this protocol, biticoinwiki [2] provides details about the protocol. In this section we provide an overview of Stratum protocol implemented by Slushpool [6] as observed in captured packets of mining in slushpool.

Subscription of Miner. In order to register in a mining pool that supports Stratum protocol, the miner first subscribes through a subscription connection message *[Mining.subscribe, params]*, which describes the miner's capability through *params*. The mining pool server will respond with a subscription response message in the following format *[subscription, extranonce1, extranonce2_size]*. Here, *subscription* is an array of 2-item tuples, each with name of subscribed notification and subscription ID. *extranonce1* is a hex-encoded, per-connection unique string that will be used for creating generation transactions later, and *extranonce2_size* is the number of bytes that the miner uses for its *Extranonce2* counter.

Authorization of Miner. After each connection subscription request, the miner authenticates with the pool through a miner authorization request message in the following format: *[Mining.authorize, username, password]*

Here, the *username* has two parts: miner's username and worker id to authorize multiple workers. The *password* field is provided in clear text, and it is optional for most mining pools.

Share Difficulty Notification. Following a successful authorization of miner, the pool server sends a share difficulty notification with the minimum share difficulty the pool server is willing to accept using *Mining.set_difficulty*.

Assignment of Job. Since the Stratum mining protocol works in publish-subscribe manner, all of the subscribed and authorized miners will be notified when a new job is available in the pool and will be assigned using different parameters in the following format: *[Mining.notify, job_id, params, clean_jobs]*. Here, the *params* field contains all of the puzzle parameters such as the fields of tuple F mentioned in Sect. 2.2. *clean_jobs* is a Boolean which indicates whether a miner should drop all previous jobs and work exclusively on the current one.

Submission of Shares. Once a miner finds a solution that satisfies the requirement provided in *Mining.set_difficulty* method using all of *params* from *Mining.notify* job assignment response and miner's calculated *nonce* and *extranonce2*, it will send the solution to the pool for verification and credit in the following format *[Mining.submit, user_id, job_id, time, nonce, extranonce2]*.

Here, *user_id* is obtained from the response of *Mining.authorize* request, *job_id* is obtained from the *Mining.notify* response. *nonce* and *extranonce2* is the puzzle solution which meets the difficulty provided in *Mining.set_difficulty*. The pool server will use these values to reconstruct the F value mentioned in Sect. 2.2 and verify that Relation 1 is satisfied. The pool server will respond with a status message denoting *accepted* or *rejected*. A share can be rejected for two reasons: *stale share* which is submitted too late, and *bad share* which does not satisfy the difficulty requirement.

In summary, as shown in Fig. 1, after the subscription and authorization of miner by pool server through *mining.subscribe* and *mining.authorize* API, the pool server will send the share difficulty and multiple new jobs to solve through *mining.set_difficulty* and *mining.notify* API. Now, the miner has to find a *nonce* and *extranonce2* for every job it wants to solve that will satisfy the share difficulty set by the pool server and submit it to the pool server through *mining.submit* API.

2.4 Invalid Share Policy

From our study, crypto-currency mining pools have to face the issue of cheating by pool members (i.e., those who do not do the job but just submit invalid

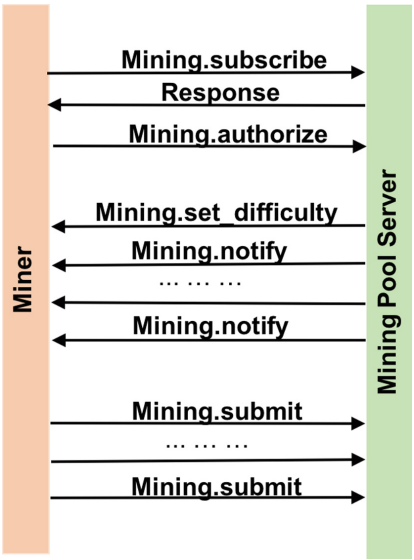

Fig. 1. Interaction between miner and mining pool server

shares). Therefore, they establish various penalty policies for such participants. For example, the Minergate policy [23] mentions: "Open source pools by default ban users if the percentage of invalid shares is bigger than 25%. However, MinerGate does not allow hackers to cheat the pool and follow certain policies for invalid share. For sending invalid share user gets penalty and his unconfirmed balance decreases, sending multiple invalid shares will lead to negative unconfirmed balance. This will prevent cheaters from having funds in confirmed balance." In Table 1, we summarize the negative impact of submitting invalid shares to several public mining pools. We can see that misbehaving users are often banned to some extent and their wallets can even be locked.

As we mention in Sect. 2.3, password is not required at most mining pools when authorizing miners, which means that anyone can impersonate other miners during crypto-currency mining. Therefore, an adversary can leverage this fundamental "vulnerability" of the mining pool protocol for an effective attack, in which the attacker's goal is to cause the mining pool administrator to mistakenly penalize innocent miners.

3 Attack Method

In this paper, we propose a way to attack mining pools using the publicly available information about the mining pool, miners and mining APIs. This attack will decrease the hashrate of a mining pool in two ways. First, since the pool server will have to validate invalid shares submitted by the attacker, it will add workload to the pool server. Second, decreasing pool hashrate will decrease the

Table 1. The negative impact of submitting invalid shares

Mining site	Banned	Payouts locked	Balance reduced
moneroocean.stream	Temporary (1–10 min)	No	No
xmrpool.net	Yes	No	No
supportxmr.com	Yes	Yes	No
www.viaxmr.com	Temporary	No	No
minergate.com	No	No	Yes
slushpool.com	Yes	No	No
moriaxmr.com	Temporary (10 min)	No	No
ratchetmining.com	Temporary (10 min)	No	No

earning of the pool, which will discourage new miners to join the pool and encourage affected miners to leave the pool. Third, since this attack submits invalid shares on behalf of top benign miners and the pool policy imposes penalty like ban of miners or penalizing balance, which can greatly reduce the productivity of the pool and encourage affected members of the pool to leave.

In this section, we outline steps to perform an attack on mining pools. Each step will be elaborated in subsequent subsections. The steps as shown in Fig. 2 are given below:

1. Collect mining pool and miner information
2. Collect mining API information
3. Submit invalid shares to attack miner reputation
4. Check attack results
   (a) if the pool marks the submitted shares as invalid we are successful and exit
   (b) otherwise restart from step 1 by collecting new information about pool, miner and API

In the following, we describe the details of how an attacker can collect miner account information in Sect. 3.1 and mining API information in Sect. 3.2. Then we discuss the actual attack scheme in Sect. 3.3.

3.1 Collecting Mining Pool and Miner Information

Collecting the mining pool and miner information is the first step of our attack. Based on the available resources to perform the attack, we can select the appropriate pool. In the case of bitcoin, BTC.com provides a list of pools and their hash rate distribution (Fig. 3 and [3]). Using this list, we can decide which pool(s) can be attacked successfully with the available resources. Since all of the mining pools publicly share their server addresses for each supported crypto-currency, mining pool server addresses can be collected from the targeted pool's website. Miner's username and contribution to a specific pool can be collected in two

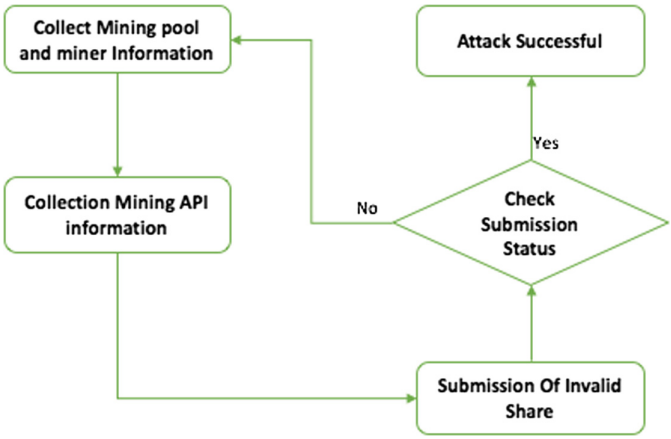

Fig. 2. The flowchart of proposed approach

ways. First, some pools like sluspool.com and minergate.com share information about the top contributors [7,25] of the pool and the corresponding user names of the miners. Second, since the communication between the mining pool and miners happens through the Stratum protocol, a clear-text JSON format, we can figure out the mining pool's top contributors by traffic analysis if we know the miner or mining pool's address as described in [27]. For slushpool [25], the top 100 miners contribute more than 90% hashrate of the pool. Therefore, submitting invalid shares on behalf of these top contributors will trigger the pool policy to penalize the miners, which will encourage the top miners to leave the pool. Additionally, departure of attacked miners will drastically decrease the hashrate of the pool. For our analysis, we used publicly available information about miners and mining pools.

3.2 Collecting Mining API Information

Most of the mining pools follow standard API name provided by Stratum protocol [2,6]. However, some of the mining pools like Minergate [5] do not follow the standard API name. These pools use customized API name instead. Thus, lack of standard Stratum protocol API name calls for network traffic analysis to discover API name used by the corresponding mining pool. To get the API name used by a specific mining pool, we can mine in the pool using their mining application (e.g., [8–10]) as a benign user and capture the network traffic of the mining application. Since Stratum is a clear text JSON format protocol, the captured traffic will reveal the API name used by the mining pool.

In our analysis, Minergate [5] mining pool does not follow the Stratum protocol standard [6]. However, mining as a benign user using Minergate's mining application [10] and capturing the mining traffic from the application reveal the API name used by the Minergate pool server. For the subscription and authorization of miner, Minergate's API name is "login" wheres the standard Stratum

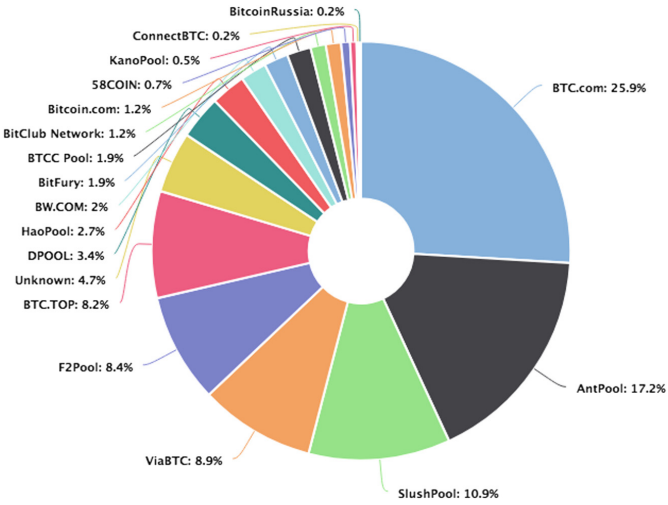

Fig. 3. Hash rate distribution of mining pools

API name is "subscribe" and "authorize". Captured traffic of Minergate application for miner authorization is given below:

```
{"id":"1","jsonrpc":"2.0","method":"login","params":{"agent":
"MinerGateMac/6.9","login":"iden1930@gmail.com", "pass":""}}
```

Similarly, traffic analysis of Minergate application also tells us the structure of the JSON-RPC method to submit shares:

```
{"id":"2","jsonrpc":"2.0","method":"submit","params":{"id":"id
corresponding to username returned in login response", "job_id":
"Job id corresponding to job returned in login response", "nonce":
"Random value", "result":"Random value"}}
```

3.3 Attacking Miners' Reputation

After getting a victim miner's username by the approach described in Sect. 3.1, our next step is to mine badly on behalf of this miner. We cannot use the official mining software because it is designed to mine honestly. Therefore, we have to create a special tool that speaks the mining pool language in order to interact with the pool server but actually does not do any real mining.

We build a tool to carry out our scheme. First we need to know the protocol to login to the pool server, get new jobs, and submit results, including method names and the parameters.

We obtain the above necessary information by analyzing the official mining software (e.g., [8–10]). From the documentation provided by mining pools, we learn the command line arguments needed to run the mining executable, such as the username, mining pool address, and port number. Next, we apply the traffic analysis discussed in Sect. 3.2 to learn the mining API.

Using what we learned from software analysis and traffic analysis, we can now form the login request and submit it to the pool server using a TCP connection. In response to the login request, the pool server returns data containing *job data, job_id, target, time_to_live* and *user_id* corresponding to the username. At that point, the task of a legitimate miner is to find a nonce and extranonce2 that will generate a hash of job data concatenated with a nonce and extranonce2 that is less than the provided target. As our goal is not to help the legitimate miner, we will not generate the hash, instead we randomly generate a nonce, extranonce2, and the result. It is less likely to get a valid share from random selection as getting a good share is a difficult task. Now, we submit the random nonce, random extranonce2, and result to the pool server which will most likely recognize it as an invalid share. We have developed a tool that can send a large number of invalid shares to the pool server in a short period of time.

4 Evaluation

4.1 Feasibility of the Attack in Terms of Required Resources: Theoretical Analysis

We will show that an attacker needs only *a small fraction of* the resources of a top miner in order to successfully attack the top miner, i.e., to cause the mining pool administrator to mistakenly penalize the top miner.

Since a top miner submits valid shares at a very high rate, the attacker also has to submit invalid shares at a very high rate in order to make the percentage of invalid shares of the top miner reach the threshold to be punished. This seems to imply that the attacker would also need significant amount of resources, which increases the cost of the attack and if the cost is too high, the attack would not be worth it. However, our analysis below shows that the attacker can reach the required submission rate of invalid shares at a much lower cost (e.g., 1 millionth of) than the top minor. This is due to the fact that the top miner has to perform a large number of (e.g., 2^{32}) hashing operations between share submissions, while the attacker does not have to.

Specifically, the same amount of resource can be used to submit invalid shares at a much higher rate than to mine and then submit valid shares. As an illustration, let's consider a concrete case. Based on empirical data provided in [27], the average time for a miner with hashrate 4096 GH/s to find a share with difficulty 1024 is

$$\frac{1024 * 2^{32}}{4096 * 2^{30}} = 1 \tag{2}$$

seconds. In other words, such a miner can submit one (1) valid share per second.

On the other hand, suppose the network bandwidth between the miner and the pool server is 640 Mbps or 80M bytes per second, since the average size of network packets containing the shares is 80 bytes, a bad miner can send up to

$$80M/80 = 2^{20} = 1,048,576 \tag{3}$$

invalid shares per second. Here we assume that the bad miner can utilize the entire network bandwidth in the ideal case.

Based on the above two equations, the share submission rate difference (invalid vs valid) is 1,048,576 times. In other words, to reach the same share submission rate, an attacker requires one millionth of resources that a benign miner would need.

The above analysis of attack cost is still an over-estimation because the attacker does not need to submit the same number of invalid shares as the top miner in order to succeed. This is because the percentage threshold of invalid shares to punish a miner is much lower than 50%. Formally, suppose the attacker and the top miner use the same kinds of mining nodes, the share submission rate difference between malicious nodes (MNs) and honest nodes (HNs) is n times, the percentage threshold to punish a miner is r, and one MN can beat x HNs, we can compute x as follows. In one time unit, the MN can produce n invalid shares, while the x HNs produce x valid shares, so the invalid share percentage is $\frac{n}{x+n}$; when the threshold is reached, i.e., $\frac{n}{x+n} = r$, we have $x = \frac{n}{r} - n = \frac{1-r}{r} * n$. For example, if n = 1,000,000 and $r = 0.2$, one MN can beat $\frac{1-0.2}{0.2} * 1,000,000 = 4,000,000$ HNs. This means that an attacker can use much less resource to get a benign miner punished.

4.2 Experimental Evaluation

We have experimentally confirmed the feasibility of getting a victim miner penalized by mining badly on his behalf.

To validate our approach, we create a user account at Minergate. Now, our tool uses this account username to submit invalid shares to Minergate pool server following the procedure described in Sect. 3.3. After submitting around 40,000 invalid shares, the account balance decreases from 0.00002398 to 0.00001973, which follows the invalid share policy of Minergate pool. Figures 4 and 5 show the change in our account state before and after we ran our tool.

We also perform the same actions using one existing miner's username at Minergate, and the response from the mining pool server indicates that it detects the submitted shares as invalid. Although we are not able to check whether the pool administrator penalizes the victim miner (because we do not know that miner's password), the Minergate invalid share policy mentions that the pool administrator should penalize the miner for every invalid share. As the pool administrator penalizes our account for submitting invalid shares, it should penalize the victim miner as well for submitting invalid shares. For ethical consideration, we did not carry out a large-scale and sustained attack against the victim miner's account. In reality, it is quite likely that a determined attacker would launch a serious attack in order to bring down the productivity of a victim mining pool.

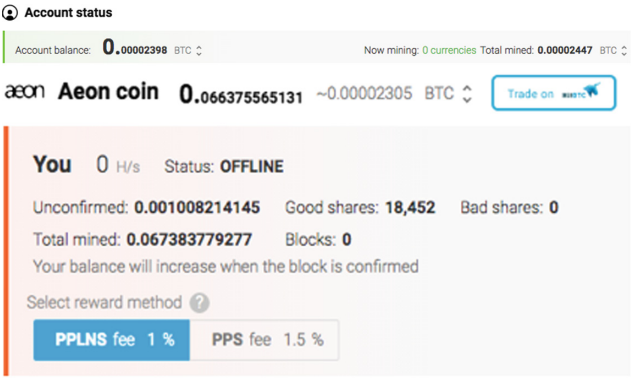

Fig. 4. Account status before mining for Aeon Coin

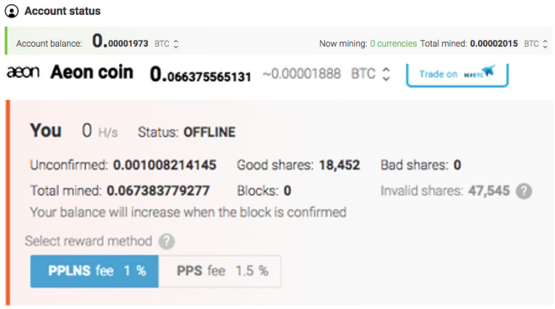

Fig. 5. Account status after submission of 40,000 invalid shares for Aeon Coin

4.3 Responsible Disclosure

Due to the potential serious damage that our attack scheme can inflict to public mining pools and the pool-based mining ecosystem in general, we have initiated the process to notify affected mining pools. We have contacted Slush Pool through their Twitter account (@slush_pool) and the Twitter account of the CEO and co-founder Jan Capek. Jan has expressed great interest in our proposed attack, and the discussion between Slush Pool and us is still going on. We also contacted Minergate through its customer service email.

5 Possible Remediation of the Attack

The poisoning attack described in this paper would be defeated if the mining pool server enforces miner authentication. Since the attacker would not know the password of the innocent miner, she will not be able to authenticate and then mine on behalf of the victim miner. The Stratum protocol, which is used by many popular mining pools, already supports user authentication (see Sect. 2.3).

Unfortunately, this feature is often not used at those mining pools, perhaps to minimize the performance overhead. Therefore, we highly recommend that mining pools enforce miner authentication.

We also recommend that pool mining protocols adopt encryption (e.g., HTTPS). Currently, mining protocols such as Stratum is clear text based, which is susceptible to Man-in-the-Middle (MITM) attacks. Basically, an MITM attacker can eavesdrop on the communication between an innocent miner and a mining pool server to steal security credentials and session tokens, and then use them to inject bogus messages that translate to submissions of invalid shares on behalf of innocent miners. Adding encryption would raise the bar for the MITM attackers.

6 Related Work

As [13] points out, a mining pool might be able to increase its revenue by attacking other pools. Eyal et al. propose Selfish Mining [14], an attack against the Bitcoin mining protocol that allows colluding miners to obtain a revenue larger than their fair share.

Mining pools have been constant targets of DDoS attacks. According to empirical studies [16], mining pools are the second-most targeted Bitcoin service after currency exchanges. Among 49 mining pools, 12 encountered DDoS attacks, and at least one mining pool (Altcoin.pw) had to shut down because of DDoS attacks. However, most of these DDoS attacks are performed actively by isolating targeted pool from other parts of the network or making it unavailable to the pool members. Most of those DDoS attacks can be detected using current DDoS detection tools like cloudFire since the attackers are using the network, not the mining or crypto-currency protocol or implementation to perform those attacks. Moreover, [18] presents a game-theoretic analysis of DDoS attacks against bitcoin mining pools.

Most of the existing attacks [15,19,21,22,26] against mining pools are at the network level, not at the protocol level. In [15,19,26], the authors discuss an eclipse attack on bitcoin network that is at the network level. [21] proposes the fork after withholding attack in which miner's dilemma [13] does not hold. [12] discusses the block withholding attack and the corresponding attacker's strategies based on the mining consensus protocol. In [13], Eyal presents a 51% attack using 51% resources of the network, which can be achieved using our proposed attack scheme.

7 Conclusion

The increasing popularity of crypto-currency has encouraged the formation of large and collaborative mining pools. Unfortunately, the huge economic impact of crypto-currency mining has also brought forth various attacks against mining pools. In this paper, we identify a serious attack scheme that can significantly slow down the production rate of a mining pool. The attacker can cause innocent

and productive miners of a pool to be punished by submitting invalid mining results on behalf of the victim miners. This attack essentially takes advantage of a combination of the lack of miner authentication and the penalty policy established by mining pools with respect to invalid shares. We present a theoretical analysis to show that an attacker needs only a small fraction (e.g., millionth) of the resources of a victim miner to succeed, making the attack very affordable. We also experimentally confirm the feasibility of our attack against Slush Pool and Minergate. Our study strongly suggests that we should rethink the design of pool mining protocols.

Acknowledgement. This work is partially supported by the US National Security Agency (NSA) under grant number H98230-17-1-0354, and the US DoD Army Research Office (ARO) under grant number W911NF-17-1-0437. The views and conclusions contained in this paper are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the United States National Security Agency or Army Research Office. We also thank the anonymous reviewers for their insightful comments.

References

1. Hash Rate Proof. https://slushpool.com/help/manual/hashrate-proof
2. bitcoinwiki Stratum-protocol (2018). https://en.bitcoin.it/wiki/Stratum_mining_protocol. Accessed 28 May 2018
3. btc.com (2018). https://pool.btc.com/pool-stats. Accessed 28 May 2018
4. Drive by cryptocurrency mining (2018). https://www.malwarebytes.com/pdf/white-papers/Drive-By-Cryptocurrency-Mining_Malwarebytes-Labs-Report.pdf. Accessed 28 May 2018
5. minergate (2018). https://minergate.com/. Accessed 28 May 2018
6. Slushpool Stratum-protocol (2018). https://slushpool.com/help/manual/stratum-protocol. Accessed 28 May 2018
7. slushpool top contributor (2018). https://slushpool.com/stats/hall-of-fame/. Accessed 28 May 2018
8. ASIC and FPGA miner in C for bitcoin (2018). https://github.com/ckolivas/cgminer. Accessed 16 June 2018
9. BFGMiner a modular ASIC/FPGA Bitcoin miner (2018). http://bfgminer.org/. Accessed 16 June 2018
10. Cryptocurrency GUI miner 8.1 & Mining Pool (2018). https://minergate.com/download/win (2018). Accessed 16 June 2018
11. Buterin, V., Griffith, V.: Casper the friendly finality gadget. In: arXiv preprint arXiv:1710.09437 (2017)
12. Courtois, N.T., Bahack, L.: On subversive miner strategies and block withholding attack in bitcoin digital currency. CoRR abs/1402.1718 (2014). http://arxiv.org/abs/1402.1718
13. Eyal, I.: The miner's dilemma. In: 2015 IEEE Symposium on Security and Privacy, pp. 89–103, May 2015. https://doi.org/10.1109/SP.2015.13
14. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: Proceedings of the Eighteenth International Conference on Financial Cryptography and Data Security (FC 2014) (2014)

15. Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on bitcoin's peer-to-peer network. In: Proceedings of the 24th USENIX Conference on Security Symposium, SEC 2015, pp. 129–144. USENIX Association, Berkeley (2015). http://dl.acm.org/citation.cfm?id=2831143.2831152
16. Huang, D.Y., Dharmdasani, H., Meiklejohn, S., Dave, V., Grier, C., Mccoy, D., Savage, S., Weaver, N., Snoeren, A.C., Levchenko, K.: Botcoin: monetizing stolen cycles (2014)
17. Huang DY, Dharmdasani H, M.S.: Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. In: Proceedings of the Network and Distributed System Security Symposium. Reston, Virginia: Internet Society (2014)
18. Johnson, B., Laszka, A., Grossklags, J., Vasek, M., Moore, T.: Game-theoretic analysis of DDoS attacks against bitcoin mining pools. In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) FC 2014. LNCS, vol. 8438, pp. 72–86. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44774-1_6
19. Karame, G.O., Androulaki, E., Capkun, S.: Double-spending fast payments in bitcoin. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS 2012, pp. 906–917. ACM, New York (2012). https://doi.org/10.1145/2382196.2382292, https://doi.acm.org/10.1145/2382196.2382292
20. King, S., Nadal, S.: PPCoin: peer-to-peer crypto-currency with proof-of-stake. In: self-published paper, August 2012
21. Kwon, Y., Kim, D., Son, Y., Vasserman, E., Kim, Y.: Be selfish and avoid dilemmas: fork after withholding (FAW) attacks on bitcoin. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pp. 195–209 ACM, New York (2017). https://doi.acm.org/10.1145/3133956.3134019, https://doi.org/10.1145/3133956.3134019
22. Luu, L., Saha, R., Parameshwaran, I., Saxena, P., Hobor, A.: On power splitting games in distributed computation: the case of bitcoin pooled mining. In: 2015 IEEE 28th Computer Security Foundations Symposium, pp. 397–411, July 2015. https://doi.org/10.1109/CSF.2015.34
23. MinerGate: Invalid shares policy. https://minergate.com/faq/invalid-shares-policy. Accessed 05 Feb 2018
24. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008). http://bitcoin.org/bitcoin.pdf. Accessed 28 May 2018
25. Nakamoto, S.: Slushpool hashrate (2018). https://slushpool.com/stats/?c=btc. Accessed 28 May 2018
26. Nayak, K., Kumar, S., Miller, A., Shi, E.: Stubborn mining: generalizing selfish mining and combining with an eclipse attack. In: 2016 IEEE European Symposium on Security and Privacy (EuroS P), pp. 305–320, March 2016. https://doi.org/10.1109/EuroSP.2016.32
27. Ruben Recabarren, B.C.: Hardening stratum, the bitcoin pool mining protocol. In: 1st Workshop on Bitcoin Research