The Korean Institute of
Communications and
Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express xxx (xxxx) xxx

www.elsevier.com/locate/icte

Rethinking selfish mining under pooled mining

Suhyeon Lee, Seungjoo Kim[*]

ICSP (Institute of Cyber Security & Privacy), Korea University, 145 Anam-ro, Seoul, Republic of Korea

Received 4 November 2021; accepted 6 March 2022
Available online xxxx

Abstract

Bitcoin's core security requires honest participants to control at least 51% of the total hash power. However, it has been shown that several techniques can exploit the fair mining in the Bitcoin network. This study focuses on selfish mining, which is based on the idea, "keeping blocks secret." Herein, we analyze selfish mining regarding competition between mining pools. We emphasize that mining-related information is shared between mining pools and participants. Based on shared information about selfish mining, we have developed an effective and practical counter strategy.

© 2022 The Author(s). Published by Elsevier B.V. on behalf of The Korean Institute of Communications and Information Sciences. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: Blockchain; Mining pool; Security analysis; Selfish mining; Proof-of-Work

1. Introduction

Bitcoin [1] is the first successful electronic cash system which applied a permissionless blockchain network. Bitcoin uses Proof-of-Work (PoW) as a Sybil control mechanism and guarantees a fair mining competition based on each miner's computing power. Satoshi Nakamoto originally predicted that it would be impossible for an individual to have 51% of the total hash power. However, when mining is being performed by groups called mining pools, this just might be possible. What is concerning is that researchers have found various security problems without 51% mining power [2]. In particular, attack methodologies that allow for higher mining performance than owned mining power pose a severe threat to PoW blockchain ecosystems [3]. This type of attack includes the selfish mining attack [4], and block withholding (BWH) attack [5], and the block denial of service (BDoS) attack [6].

Researchers have been looking into solutions to these two attack techniques. Most of these studies [7–10] focus on stultifying blocks. In contrast, other studies focus on preventing attackers from withholding the blocks. In theory, the solutions work well, but two-phase PoW mechanisms [11] are difficult to apply in the real world. This is because the mining mechanism itself must be modified. Moreover, Application Specific Integrated Circuit (ASIC) miners do not want to adopt this kind of method because they must make changes in hardware. Solutions [12–14] that modify the incentive function have a fairness issues to miners.

Nowadays, the mining environment for PoW-based cryptocurrencies is highly concentrated, to the point where mining is no longer a competition between individuals, but rather between mining pools. In this article, we analyze selfish mining across mining pools and provide a countermeasure against these attacks. The following summarizes the paper's major contributions:

- By leveraging the structure of the mining pool and internal communication protocol, we suggest a novel countermeasure named "detective mining" against selfish mining pools.
- A countermeasure that is very practical compared to existing methods. This method does not include any modification of the PoW algorithm itself, but only modifies mining pool mechanisms.

The rest of this paper is organized as follows. Section 2 introduces a system model of our analysis on pooled mining. we propose a counter strategy against selfish mining in Section 3 and the simulations are illustrated in Section 4. We compare and discuss our approach with similar studies in Section 5 before we conclude our research in Section 6.

[*] Corresponding author.
E-mail addresses: orion-alpha@korea.ac.kr (S. Lee), skim71@korea.ac.kr (S. Kim).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2022.03.003

2405-9595/© 2022 The Author(s). Published by Elsevier B.V. on behalf of The Korean Institute of Communications and Information Sciences. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

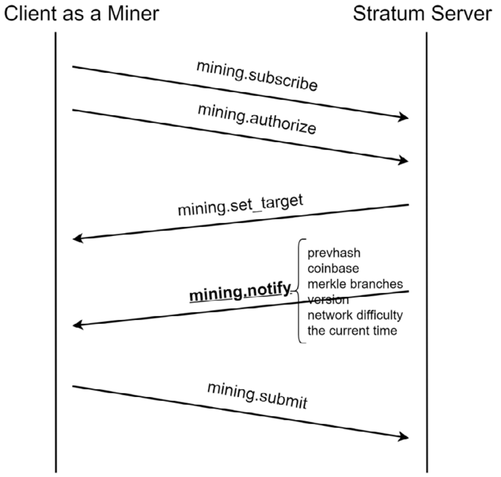

Fig. 1. Example communication between a miner and Stratum server in the Stratum protocol.

2. Pooled mining model

In this section, we introduce the pooled mining model in this study based on the Bitcoin mining and the real world pooled mining.

2.1. Pooled mining structure

In mining pools, miners subscribe to mining tasks from mining pool servers and get rewards by submitting proper solutions. The target difficulty of mining tasks in pooled mining is more accessible than the pure Bitcoin mining. Miners in the pooled mining prove that they are working by solving small solutions. If one miner finds a valid block to be a valid Bitcoin block, the miners will share just as many rewards as the amount of their contribution.

To describe communication between miners and mining pools, we refer to Stratum protocol [15]. This protocol is not only used for the Bitcoin pooled mining, but also for PoW based cryptocurrencies like Ethereum and Zcash [16,17]. A miner registers him to a Stratum server, gets mining information, and then submits mining results. It provides an intuitive and efficient JSON-RPC interface for allowing targeted mining for miners. The primary methods of the Stratum mining protocol are *subscribe*, *authorize*, *notify*, *set_target*, *submit*, and *reconnect*. It also includes error codes. It was proposed for the pooled mining for Bitcoin. By subscription, miners receive notification that includes prevBlockHash, coinbase1, coinbase2, a list of Merkle branches, version, timestamp, and bits (see Fig. 1).

2.2. Our model

We assume that all of the PoW mining is performed by pooled mining. That is, all miners join in mining pools. Also, in reality, the majority of block mining is performed by mining pools in Bitcoin and Ethereum. If we consider that most of the blocks are generated by mining pools, it is acceptable. Secondly, we assume that mining pools can monitor other pools' mining information. It is possible by sensors that subscribe to mining tasks of other mining pools.

Therefore, in a notification by subscription, mining pools can get prevBlockHash, coinbase, Merkle branches, version, and the difficulty as we described in Section 2.1. PrevBlockHash indicates what block is the former block of the mining. For example, two blocks have the same height in a fork situation. Mining pools should choose one branch to mine. By PrevBlockHash information from sensors, mining pools can figure out which mining pools work for which branches.

3. Countermeasure against selfish mining

In this section, we propose a countermeasure against the selfish mining attack.

3.1. Detective mining

Our model shows that it is possible to obtain a part of private blocks using sensors and will notify a system if a selfish mining attack occurred. This can trigger other mining pools to execute mining on the private chain of the selfish mining pool. The point is that the pooled-mining information includes the previous block hash as Bitcoin mining is possible if a miner knows the previous block hash. Because this countermeasure includes a proactive mining behavior, we call it "Detective Mining" [18]. We get an idea of naming this word detective mining, otherwise known as ghost mining, was a name originally used to describe people that worked old mines and found gold that previous miners left behind. By using the pooled mining information of the selfish mining pool, other miners (referred to as "detective miners" or "detective mining pools") can reconstruct the selfish mining pool's hidden chain and neutralize the selfish mining's effect.

The detective mining's algorithm is described in Algorithm 1. At first, a detective mining pool keeps mining jobs from sensors. If the mining jobs contain an unknown previous block hash value, then it is apparent that another mining pool is operating the selfish mining technique. If the mining pool finds unknown previous block hash continuously, the mining pool regards the selfish mining pool as continuously generating blocks — the detective mining pool works with the latest unknown previous block hash value. Furthermore, if the detective mining pool generates a block, the new block is broadcast to the Bitcoin network. At last, if $H \cap U = U$, it means that the selfish mining is finished as all the private blocks are released. So the list of unknown previous blocks is initialized as a null set ϕ.

In the selfish mining strategy [4], this case is not defined. We assume that if a new block is released on its private chain, the selfish mining pool will release blocks in its private chain. Because the crux of selfish mining is to make other miners waste on a shorter chain than the selfish miner's longer private chain. If another miner finds a block that can construct a longer chain, the selfish miner does not need to mine on a shorter chain. If the list of confirmed blocks contains unknown previous block hash values, it implies that the selfish mining

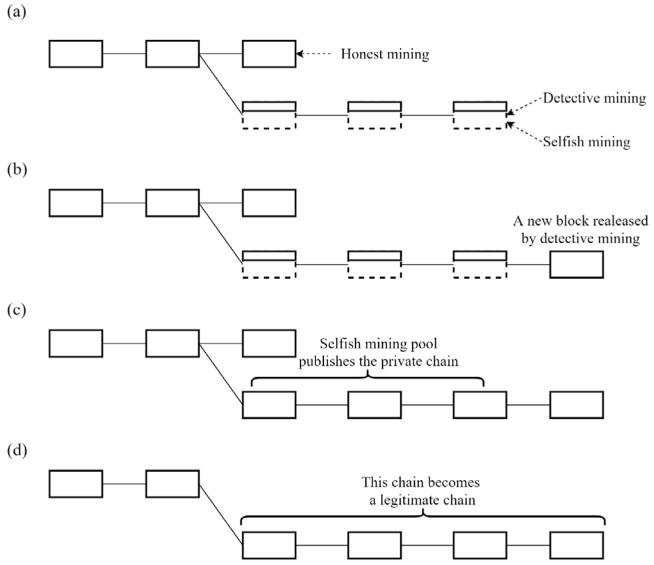

Fig. 2. Detective mining example.

Algorithm 1: Detective mining

```
Data: $H$, the list of confirmed block hashes
Data: $U$, the list of unknown previous block hashes
initialization
$H \leftarrow UpdateBlocks()$
while mining do
    $h \leftarrow Sensors()$
    /* If previous block hash is unknown */
    if $h \notin H$ then
        put $h$ into $U$
    /* Mining on unknown prevBlockHash */
    if $U > 0$ then
        do mining with $U$
    else
        do mining with $H$
    $H \leftarrow UpdateBlocks()$
    /* When no hidden chain */
    if $H \cap U = U$ then
        $U \leftarrow \phi$
```

pool released all the private blocks. The detective mining pool can avoid loss from the selfish mining strategy and receive an extra profit by repeating this algorithm.

For example, Fig. 2 illustrates situations related to the detective mining. In Fig. 2(a), the selfish mining pool has 3 private blocks, and its chain is 2 blocks longer than the public block. The detective mining pool uses the sensors to get to know a part of the private blocks, significantly block hash values. That is why the private block from the selfish mining is illustrated as solid lines, and the private blocks are illustrated as dotted lines. The detective mining pool can mine at the latest private block by the hash values of such blocks. In Fig. 2(b), the detective mining pool generated a block and published the block. In Fig. 2(c), because of the block from the detective mining, the selfish mining pool releases the private blocks. In Fig. 2(d), the private blocks and the block from the detective mining become a legitimate chain as they are the longest.

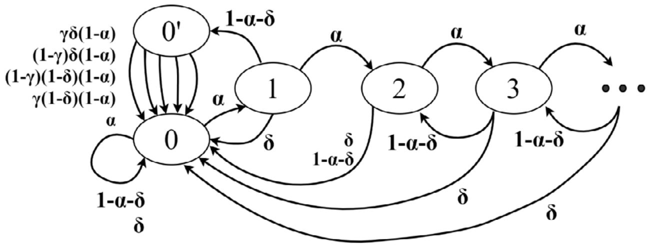

Fig. 3. State machine of the detective mining.

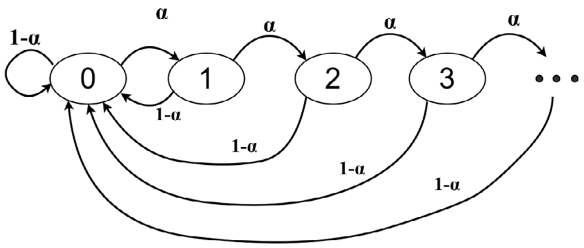

Fig. 4. State machine when all mining pools join in the detective mining except for a selfish mining pool.

3.2. A brief analysis on detective mining

Fig. 3 describes the mining game with the detective mining strategy. α indicates the mining power of the selfish mining pool. γ indicates the network preference. And, δ indicates the mining power of the detective mining pools. The detective mining pool always tries to find a block in the private chain of a selfish mining pool. It becomes too complicated for the state machine to evaluate each state's probability. Instead, we examine the detective mining's effect through simulations based on this state machine in Section 4.

In a particular case, we can evaluate the state probability. Assuming that all mining pools use the detective mining strategy, it becomes that $1 - \alpha - \delta = 0$. Fig. 4 shows the state machine with the aforementioned assumption. The selfish mining pool cannot take any extra profit proportional to its mining power in this situation. The result is that the selfish mining pool generates α of the total blocks.

In the next section, we analyze efficiency of subscription sensors for countermeasures. Then we will share our results from the simulation that used our countermeasures.

4. Evaluation

In this section, we evaluate the proposed countermeasure. At first, the scalability of monitoring mining jobs form a mining pool is analyzed. Secondly, we show simulation results of mining pools' revenue with our counter strategies. The results show that the countermeasure efficiently defends a victim mining pool from selfish mining.

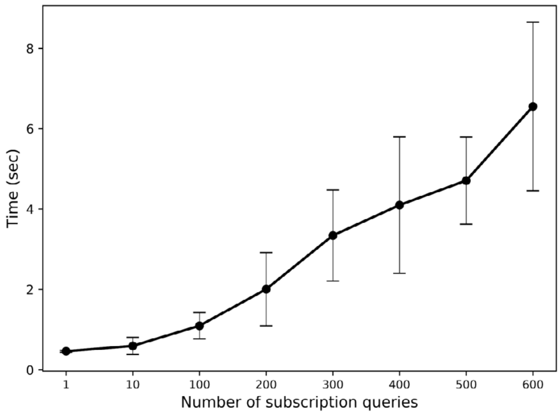

Fig. 5. Average time of mining queries.

4.1. Scalability of mining job subscription

For the proposed countermeasure, mining pools should catch a mining job (notification) which possibly contains a clue of attacks in the sense of scalability. Fig. 5 shows collecting mining jobs can be carried out in a scalable way.

We performed a test to measure the number of mining job subscription queries by time. This experiment was conducted by a Python simulator on a desktop computer with the processor AMD Ryzen 5 2600 and the RAM size 32 GB. The target mining pool is the Slush pool that provides a standard Stratum communication. We measured the time between the moments we received job notifications by sending mining subscription queries. The figure shows the average and standard error values. For every number, we tested 10 times. The time consumption appears to increase linearly with the number of subscription queries. However, the time consumption of over 600 mining job processes is exceedingly high and unstable. If more than 600 requests are sent at once, the connection may be lost for network status or security reasons. The process to send and receive 500 mining job notifications required less than 5 s. We consider it as enough performance concerning the block generation time, which is about 10 min.

4.2. Simulations of detective mining

We simulated mining pools' revenue with the detective mining against the selfish mining. The simulations are performed based on state machines. This means that we did not construct real mining pools or detailed mining blocks including transactions, and headers. For the detective mining simulation, we used the state machine in Fig. 3. For the reward reduction simulation, we made a state machine that mining pools generate and share pools' revenue to their miners without any fork. Commonly, the state machines simulated that mining pools generate 1000000 blocks to make statistically meaningful data. These are the amounts of blocks that the Bitcoin network has produced over 19 years.

In this section, we measure the revenue of mining pools based on relative revenue. The relative revenue is calculated as the proportion of block reward obtained relative to a mining pool's mining power. For example, suppose that a mining pool obtained 40% of the total block reward of the network and the pool has 20% mining power of the entire network. In this case, the relative revenue is 0.4/0.2, that is, 2. Therefore, if all mining pools take the honest mining strategy, theirs will be equal to 1. If a mining pool's relative revenue exceeds 1, then its mining technique is efficient. Because it is the relative revenue, at least one mining pool's relative revenue must be less than 1 if the relative revenue of another mining pool is more than 1.

Table 1
Comparison with the previous research on selfish mining.

Research	No loss	Compatibility	Fairness
[10]	✓	✗	✓
[19]	✓	✗	✓
[8,9]	✓	✗	✓
[19]	✓	✗	✓
[20]	✗	Δ	✓
Detective mining	✓	✓	✓

In Fig. 6, the detective mining simulation results are shown. The plots (a), (b), and (c) depict the revenue generated by the selfish mining pool, the detective mining pool, and the honest mining pool, respectively. The x-axis in the charts represents the mining power of the selfish mining pool. The y-axis represents the mining power of the detective mining pool. The mining power of the honest mining pool is deducted from the total mining power by adding the mining powers of the selfish mining pool and the detective mining pool. Fig. 6(a) shows that the selfish pool's revenue decreases steeply by the detective pool's mining power increase at parts that the selfish pool performs obviously efficient mining. Also, Fig. 6(b) shows that the detective mining pool's revenue increase over 1.5 as the relative revenue. It means that the detective mining pool is an efficient strategy against the selfish mining. Fig. 6(c) shows that the honest mining pool's revenue remains significantly low even though the detective mining is performed in another pool.

Interestingly, in Fig. 6(a), the selfish mining pool's revenue increases because of the detective mining when the selfish pool's mining power is relatively low. This is because the detective mining contributes to the private chain of the selfish mining pool. This indicates that detective mining can be enhanced using the mining power of the selfish mining pool. For instance, detective mining is possible only when the selfish mining is environmentally efficient.

5. Discussion

This section compares the proposed methods with previously studied methods and discusses the applicability of modified methods of PoW attacks.

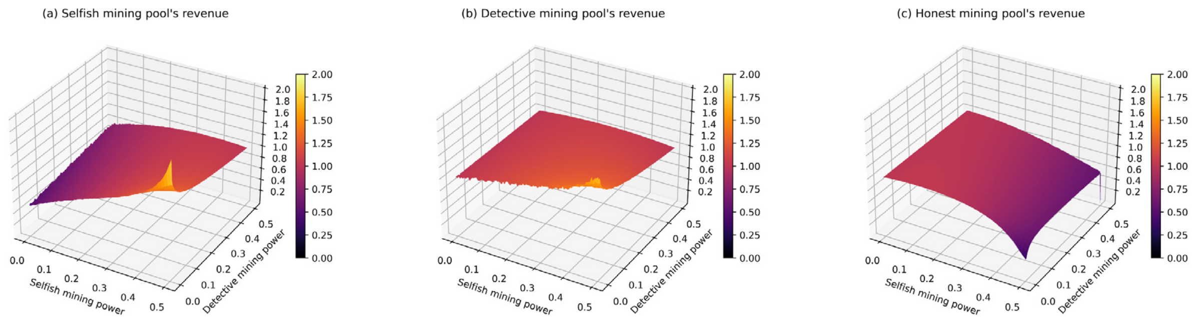

Fig. 6. Detective mining simulation results. The figures show the relative revenue that is a block reward divided by the pool's mining power. In (b), the detective mining pool's revenue is always greater than the honest mining.

5.1. Comparison with countermeasures in the previous literature

Table 1 compares the previous research and our proposed solution. We provide three crucial properties to evaluate countermeasures against selfish mining. *(Compatibility)*: A countermeasure should provide compatibility to miners in aspects of software, hardware, and network. *(No Loss)*: A countermeasure should not give loss to miners. *(Fairness)*: A countermeasure should provide fairness to miners in the pools which use it.

The most crucial property is compatibility. As the nature of decentralized networks, modification in a protocol takes a long time. Also, some miners are sensitive to a change in mining protocols. For example, ASIC miners will not accept any change in mining mechanisms. The other two properties, 'no loss' and 'fairness', may be considered to be sound solutions. The first property, no loss, requires solutions to prevent mining pools from damages effectively. The second property, fairness, requires solutions to be fair to miners that join in mining pools.

Most of approaches solve the problem fundamentally by changing the protocol and mining design itself. It is usually the most effective, but a compatibility issue always follows it. Countermeasures that try to change the long-chain rule, including fruitchain by Pass and Shi [10], strongchain by Szalachowski et al. [19], and intermediate block by Zhang and Preneel [21] are included in this approach. Also, countermeasures [8,9,22] using block publication timestamp are included. The countermeasure by Bag et al. [13] needs to change the block structure. Greedy Heaviest Object subTree (GHOST) protocol [20] used in Ethereum is included as the long-chain rule is substituted. However, the GHOST protocol is not a reasonable solution for the selfish mining [23,24]. Smart pool by Luu et al.

Contrarily, detective mining satisfy all three properties. As we already discussed, the solutions do not require modification in the PoW mining protocol. We showed that detective mining successfully prevents mining pools from the damage of revenue.

5.2. Adaptability to selfish mining variations

The optimal selfish mining strategy was studied by Saprishtein [25], and Nayak [26]. Their strategy uses mining states' flexible management based on mining power. In the middle of them, Saprishtein [25] proved their work is the optimized strategy with one selfish miner. Our strategy is based on how to find adversary-related information and how to mine after private chains. Hence, we are confident that they are also significantly affected by our detective mining. Leelavimolsilp [27], and Bai [28] studied the environment with multiple selfish miners. Their study shows that multiple selfish mining does not yield a sufficient profit compared to the solo selfish mining environment. Our strategy can discover which selfish miner has the longest private chain and can get a head start. It is remained as a future work.

6. Conclusion

This paper analyzed selfish mining in terms of competition among the mining pools. We proposed a countermeasure, detective mining, based on pooled-mining information from mining pools to be utilized against selfish mining. We demonstrated that the countermeasure are effective against selfish mining. Furthermore, this proposed method has satisfied all three properties to be labeled as a good solution.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2018-0-00532, Development of High-Assurance (⩾ EAL6) Secure Microkernel)

References

[1] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008, Available from: https://bitcoin.org/bitcoin.pdf.

[2] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Gener. Comput. Syst. 107 (2020) 841–853.

[3] K. Saito, M. Iwamura, How to make a digital currency on a blockchain stable, Future Gener. Comput. Syst. 100 (2019) 58–69.

ARTICLE IN PRESS

[4] I. Eyal, E.G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, Commun. ACM 61 (7) (2018) 95–102.

[5] I. Eyal, The miner's dilemma, in: Security and Privacy (SP), 2015 IEEE Symposium on, IEEE, 2015, pp. 89–103.

[6] M. Mirkin, Y. Ji, J. Pang, A. Klages-Mundt, I. Eyal, A. Juels, Bdos: Blockchain denial-of-service, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 601–619.

[7] A. Miller, A. Kosba, J. Katz, E. Shi, Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015, pp. 680–691.

[8] S. Solat, M. Potop-Butucaru, ZeroBlock: TImestamp-free prevention of block-withholding attack in bitcoin, 2016, arXiv preprint arXiv:1605.02435.

[9] R. Zhang, B. Preneel, Publish or perish: A backward-compatible defense against selfish mining in bitcoin, in: Cryptographers' Track at the RSA Conference, Springer, 2017, pp. 277–292.

[10] R. Pass, E. Shi, Fruitchains: A fair blockchain, in: Proceedings of the ACM Symposium on Principles of Distributed Computing, 2017, pp. 315–324.

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

[12] I. Eyal, E.G. Sirer, Blog post: How to disincentivize large bitcoin mining pools, 2014, Available from: http://hackingdistributed.com/2014/06/18/how-to-disincentivize-large-bitcoin-mining-pools.

[13] S. Bag, S. Ruj, K. Sakurai, Bitcoin block withholding attack: Analysis and mitigation, IEEE Trans. Inf. Forensics Secur. 12 (8) (2017) 1967–1978.

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

[15] M.S. Palatinus, Stratum mining potocol, 2019, Available from: https://slushpool.com/help/stratum-protocol/. [Online; accessed 01-August-2021].

[16] P.B. Andrea Lanfranch, M.V. Der, EIP 1571: EthereumStratum/2.0.0, 2018, Available from: https://eips.ethereum.org/EIPS/eip-1571. [Online; accessed 01-August-2021].

[17] J.G. Daira Hopwood, ZIP 301: Zcash Stratum Protocol, 2016, Available from: Online; accessed 01-August-2021.

[18] S. Lee, S. Kim, Countering block withholding attack efficiently, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2019, pp. 330–335.

[19] P. Szalachowski, D. Reijsbergen, I. Homoliak, S. Sun, Strongchain: Transparent and collaborative proof-of-work consensus, in: 28th {USENIX} Security Symposium ({USENIX} Security 19), 2019, pp. 819–836.

[20] Y. Sompolinsky, A. Zohar, Secure high-rate transaction processing in bitcoin, in: International Conference on Financial Cryptography and Data Security, Springer, 2015, pp. 507–527.

[21] R. Zhang, B. Preneel, Broadcasting intermediate blocks as a defense mechanism against selfish-mine in bitcoin, 2015, Cryptology ePrint Archive, Report 2015/518. https://eprint.iacr.org/2015/518.

[22] E. Heilman, One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner, in: International Conference on Financial Cryptography and Data Security, Springer, 2014, pp. 161–162.

[23] F. Ritz, A. Zugenmaier, The impact of uncle rewards on selfish mining in ethereum, in: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2018, pp. 50–57.

[24] J. Niu, C. Feng, Selfish mining in ethereum, 2019, arXiv preprint arXiv:1901.04620.

[25] A. Sapirshtein, Y. Sompolinsky, A. Zohar, Optimal selfish mining strategies in bitcoin, in: International Conference on Financial Cryptography and Data Security, Springer, 2016, pp. 515–532.

[26] K. Nayak, S. Kumar, A. Miller, E. Shi, Stubborn mining: Generalizing selfish mining and combining with an eclipse attack, in: Security and Privacy (EuroS&P), 2016 IEEE European Symposium on, IEEE, 2016, pp. 305–320.

[27] T. Leelavimolsilp, L. Tran-Thanh, S. Stein, On the preliminary investigation of selfish mining strategy with multiple selfish miners, 2018, arXiv preprint arXiv:1802.02218.

[28] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, Q. Kong, A deep dive into blockchain selfish mining, 2018, arXiv preprint arXiv:1811.08263.