Short Paper: Anti-Withholding Reward System to Secure Blockchain Mining Pools

Arijet Sarker Simeon Wuthier Sang-Yoon Chang
University of Colorado Colorado Springs
{asarker, swuthier, schang2}@uccs.edu

***Abstract*—In decentralized blockchain systems such as Bitcoin, the miners are rewarded for generating new blocks and processing the transactions. In order to reduce the reward variance, the miners join mining pools where the miners aggregate their computational resources and divide the reward according to their contributions, measured by shares. The miner-based attacks such as Block Withholding (BWH) and Fork After Withholding (FAW) yield unfair reward advantages to the attackers at the expense of the other honest miners by having the attackers withhold blocks while still submitting the shares and pretending to contribute to the victim pool. This paper introduces Anti-Withholding Reward System (AWRS) to prevent FAW and BWH attacks. Implemented at the pool manager (reducing the adoption overhead and supporting backward-compatibility), AWRS deprives the incentive for FAW and BWH attacks by providing greater rewards to the block submissions. AWRS completely disincentivizes the FAW attack and reduces the optimal attacker's behavior to honest mining, making the withholding-based threats irrelevant for rational miners.**

I. Introduction

Since Bitcoin's release in 2008 [1], blockchain technology has flourished and enabled financial and other applications without relying on the centralized authority. Blockchain, comprised of an unalterable linear chain of transactions put into blocks, is synchronized across the different users using a distributed consensus protocol based on proof of work (PoW). In PoW, the participants are called *miners* as they mine/produce new currencies or earn the transaction fees by solving a hash-based probabilistic computational puzzle, which generate new blocks. Mining is computationally fair (the greater the computational power the more likely to mine a valid PoW and generate the new block) and, in order to lower the reward variance and provide a more stable reward, miners join mining pools. Mining pool is popular, for example, in May 2019, just seven mining pools comprise of more than 85% of the entire network's computational power [12]. However, mining pools are susceptible to various attacks in Block Withholding (BWH) and Fork After Withholding (FAW). These attacks are based on withholding blocks on the victim pool and are practical threats for rational attackers driven by reward incentives.

We propose Anti-Withholding Reward System (AWRS) to defend against BWH and FAW attacks and deprive of the incentives of the attacks. AWRS introduces asymmetry between blocks and shares by distributing a portion of the pool reward to the block submissions. AWRS is implemented at the mining pool manager and does not incur any changes in the miners' implementations because it only changes the reward distribution controlled by the pool manager; therefore, it presents a solution to BWH and FAW attacks which is easy to implement and get adopted for practice.

We study AWRS in theoretical analyses and in simulations to show its effectiveness against an attacker which is rational (optimizing its strategies for greater reward) and uncooperative (operating and diverging from the protocol at the expense of the other honest/protocol-complying miners). We focus on the FAW attack because it builds on BWH and is the stronger attack; our performances is even better against BWH attack. Our analyses results show that AWRS reduces the rational attacker strategy to honest mining (no withholding of blocks), completely depriving of the incentives to launch the attacks, regardless of the attacker's computational power capability and its infiltration strategy.

II. Background: Mining, Fork, and Mining Pool

Each block in a blockchain contains a unique block header, where the hash of this header is used to identify the block. The mining process based on the computational-puzzle-based proof of work (PoW) generates new blocks on the distributed blockchain, which results in the miner of the new block earning financial rewards. For a miner to find the next block in the blockchain, the miner needs to solve a computational puzzle which involves hashing the block header with varying nonces; the solution (and the new block) corresponds to finding a nonce input such that the hash of the block header is less than a 256-bit threshold value, called the mining difficulty. The blockchain protocol adjusts the mining difficulty to scale with the advancement in computing, so that the mining of a new block occurs regularly in expectation, e.g., every ten minutes for Bitcoin. In blockchain, when two miners find and submit valid blocks almost simultaneously, the blockchain network can disagree in the latest block because of the delay in networking and synchronization, which results in a *fork*. The fork is resolved when a new block is mined on top of either of the forked blocks, making one chain longer than the other; according to the protocol, the miners (including those which were mining on the other forked block) select the longest chain.

Since solo-mining has high reward variance, miners form a *mining pool* to pool/aggregate their computational resources and the corresponding reward winnings. Because the mining pool distributes the reward according to the individual members' contributions in the PoW, it requires a mechanisms to keep track of the mining powers and contributions of each miner. For such mechanisms, the mining pool uses *shares*, which corresponds to the same PoW computational puzzle but with lower difficulty (higher threshold value) than that of

the mining difficulty. Therefore, a share solution occurs more frequently than a block solution (providing a more accurate estimation of each miners' computation contributions), and a block solution is implicitly a share solution (a nonce solution for a block is also a solution for a share).

III. The Mining Threat

A. Related Work in Mining Threats

In blockchain mining, there are attacks which violate the mining protocol. More specifically, we review the attacks which specifically withhold the blocks to gain unfair reward advantage in selfish mining and Block Withholding (BWH) attack in this section; we discuss about the more recent Fork After Withholding (FAW) attack (which combines selfish mining and BWH) [4] in Section III-C. In selfish mining [8], if an attacker finds a block, the attacker will not broadcast the block immediately but continue to mine on it; the attacker will broadcast the block only if an honest miner's chain catches up to the attacker's. As a result, the attacker can earn extra reward by invalidating honest miner's blocks, waste the honest miner's computations, and launch double-spending attack. However, selfish mining requires great computational power, e.g., more than 25% of the network's computational power, which can limit its practicality [8], [10].

In BWH [2], the attacker pretends to contribute to the victim pool by submitting only shares and permanently withholding the block. As a result, the attacker increases the expected reward in its main pool (from which the attacker does not need to share the reward with others) while still sharing the rewards from the victim pool (due to the submitted shares). In 2014, there was a BWH attack launched against Eligius mining pool, where the pool suffered a loss of 300 BTC. Eyal [5] modeled a BWH attack game between two mining pools and showed that the BWH attack creates mutual loss for both the mining pools, providing incentives for cooperations between the mining pools and to refrain from attacking each other (the *Miner's Dilemma*). Also, Bag et al. [3] propose a special reward to the block submitter[1] to further incentivize block submissions (by introducing asymmetry to share submissions) and prevent BWH attack; our work builds on such approach but defends against a stronger attack in FAW attack.

B. Threat Model

Our threat model is that of BWH and FAW attack, which has been realized in real-world attacks as described in Section III-A. The attacker which acts as a miner has a *main pool* and a compromised *victim pool*. The two pools are different in the attacker's strategy because the attacker does not share its reward in the main pool, e.g., solo mining or closed pool with attacker-controlled miners, while it does share the reward with other miners in the victim pool. The attacker can freely split its computational power resources across the two mining pools; we call the mining used on the victim pool *infiltration mining* and allocate τ fraction of the attacker's power capability to infiltration mining. Our threat model only requires the compromise of the victim pool, as opposed to compromising the mining pool manager.

The attacker is also rational and driven by incentives. If the attack yields reward lower than honest mining (following the protocol and submitting the block when they are found), then the attacker will switch to honest mining. (In fact, AWRS aims to achieve this, which would make the FAW and BWH attacks irrelevant to the rational miners.)

Mining process is computation-fair, as opposed to identity-fair. Given computational power, the attacker's capability is the same regardless of the number of accounts/addresses the attacker has. Therefore, our work does not rely on detection- or identification-based approaches for defending against mining threats (e.g., blacklisting); such approach will be ineffective because blockchain-based cryptocurrencies are designed for anonymity and an attacker can switch to another address if blacklisted. Rather, our scheme controls the reward distribution by the mining pool manager to prevent the attack by depriving of the incentives for conducting the mining threats.

C. FAW Attack

Fork After Withholding (FAW) advances BWH by conditionally releasing the withheld blocks [4]. The block is only submitted when there is a block submitted by a third-party miner (outside of the main and the victim pool), which causes fork and hence the name Fork After Withholding (FAW). This introduces an additional reward to the attacker beyond BWH because the intentionally forked block can win the forking race[2]. Kwon et al. [4] introduce FAW and analyze its reward performance, establishing that the FAW attack is practical and provides a real incentive to the rational miners since it increases the reward while forgoing Miner's Dilemma (depriving of the incentive for inter-pool cooperations). We adapt the following results from their work in this section.

Assuming that α is the attacker's power capability normalized by the entire power network, β is the victim pool's power capability normalized by the entire power network (e.g., $0 \leq \alpha + \beta \leq 1$ and $\alpha + \beta < 1$ if there are other active miners outside of the two attacker-involved pools), c is the probability that the attacker wins the forking race for the withheld block given that the attacker-intentional fork has occurred, and τ is the attacker's infiltration power as defined in Section III-B (e.g., the attacker's power for infiltration mining is $\alpha\tau$), the attacker's expected reward for launching FAW is the following:

$$R_{\text{FAW}} = \frac{(1-\tau)\alpha}{1-\tau\alpha} + \left(\frac{\beta}{1-\tau\alpha} + c\tau\alpha\left(\frac{1-\alpha-\beta}{1-\tau\alpha}\right)\right)\frac{\tau\alpha}{\beta+\tau\alpha} \quad (1)$$

If the attacker follows the protocol and behaves honestly, then $R_{\text{Honest}} = \alpha$, because the reward is proportional to the attacker's mining power by design of the PoW mining process. If the attacker launches BWH attack, then the corresponding reward R_{BWH} is the same as R_{FAW} in Equation 1 when

[1] This special reward to distinguish blocks vs. shares has also been studied in a non-security context [13], [7], but our focus is on defeating uncooperative miners launching withholding attacks.

[2] A more recent attack in Uncle-Block Attack (UBA) [14] further advances FAW by providing rewards even in the case where the attacker loses the forking race and by exploiting the uncle rewards given to the forked blocks losing the forking race. Such uncle-reward mechanism, e.g., implemented by Ethereum, is to improve fairness/decentralization with respect to the networking conditions of the miners. However, in this work, we focus on FAW because FAW is more generally applicable across cryptocurrency implementations than UBA.

$c = 0$, which is the worst case with respect to c, since not submitting the block in BWH is equivalent to the attacker never winning the forking race in FAW. In other words, FAW is more powerful and yields greater reward to the attacker than BWH, and the reward difference depends on c.

The attacker's optimal τ control (which maximizes the attacker's reward) can also be computed:

$$\hat{\tau}_{\text{FAW}} = \frac{(1-\alpha)(1-c)\beta + \beta^2 c - \beta\sqrt{f}}{\alpha(1-\alpha-\beta)(c(1-\beta)-1)} \tag{2}$$

where f is a function of α, β, and c, i.e.,

$$f = (1-\alpha-\beta)^2c^2 + ((1-\alpha-\beta)(\alpha\beta+\alpha-2))c \\ -\alpha(1+\beta)+1$$

IV. Anti-Withholding Reward System

Our scheme called Anti-Withholding Reward System (AWRS) introduces greater incentives for block submission than the share submissions by introducing Γ to control the pool's reward to the block submitter. In other words, Γ fraction of the pool's winnings goes to the block submitter while $1-\Gamma$ is distributed according to the share submissions, e.g., proportional to the number of shares for each miners. Previously, Γ is fixed at $\Gamma = 0$ without AWRS. $\Gamma = 1$ yields solo mining because the block submitter receives all the pool reward and the share submissions do not affect the reward distributions.

AWRS is effective against both FAW and BWH attacks. To show the effectiveness of our scheme, however, we focus on FAW attack rather than BWH attack because FAW attack is the more advanced attack with greater attack/reward impact and can generalize BWH attack, as described in Section III-C.

AWRS is implemented on the mining pool manager because it only controls the reward distribution given out by the pool manager. Since it does not require any changes in the miner implementations, it has a low implementation barrier and is backward-compatible, which has been identified as a major challenge in the practicality of implementing the security defenses on blockchain [4], [11].

V. Theoretical Analyses

The mining pool manager of the victim pool implements AWRS in order to protect the integrity of its reward distribution (so that the reward is proportional to each miner's computation power for block finding) and incentivize honest mining for the rational miners. We assume the worst-case attacker capable of correctly estimating the relevant parameters and achieving $\tau = \hat{\tau}$ to maximize its reward; AWRS works even better against weaker attackers lacking such capabilities.

Against AWRS, there are three distinct events when the attacker wins positive rewards: i) when the attacker finds the block in its main pool; ii) when another miner from the victim pool, not the attacker, finds the block (the FAW attacker submits the shares and reaps the benefits from those without actually contributing to the block finding); and iii) a third-party miner finds a block and the attacker submits the withheld block. The event i) has a probability of $\frac{(1-\tau)\alpha}{1-\tau\alpha}$ because the attacker spends $1-\tau$ while there is $1-\tau\alpha$ power to finding the block across the entire network (because the attacker uses $\tau\alpha$ only for share submissions); the attacker's reward given event i) is 1 since he does not share the reward with other miners in the main pool. The event ii) occurs with a probability of $\frac{\beta}{1-\alpha\tau}$ because the victim pool, excluding the attacker, mines with a power of β; if event ii) occurs, then the attacker wins a reward of $\frac{\alpha\tau}{\beta+\alpha\tau}\cdot(1-\Gamma)$ due to its share submissions and the $(1-\Gamma)$ scale from AWRS. The event iii) occurs with a probability of $c\tau\alpha\left(\frac{1-\alpha-\beta}{1-\tau\alpha}\right)$ where the $\tau\alpha\left(\frac{1-\alpha-\beta}{1-\tau\alpha}\right)$ is the probability that the attacker found a block (withheld) and so did a third-party miner (submitting its own block) and c is the probability of the attacker winning given that the fork occurred; if this event occurs, the attacker has a reward of Γ for being the block submitter and additional $(1-\Gamma)\frac{\tau\alpha}{\beta+\tau\alpha}$ from the share submissions. Summing them together, the FAW attacker's expected reward becomes:

$$R_{\text{AWRS}} = \frac{(1-\tau)\alpha}{1-\tau\alpha} + \left(\frac{\beta}{1-\tau\alpha}\right)\left(\frac{\tau\alpha}{\beta+\tau\alpha}\right)(1-\Gamma) \\ +c\tau\alpha\left(\frac{1-\alpha-\beta}{1-\tau\alpha}\right)\left(\Gamma+(1-\Gamma)\left(\frac{\tau\alpha}{\beta+\tau\alpha}\right)\right) \tag{3}$$

Against AWRS, the attacker chooses the following τ to maximize the reward:

$$\hat{\tau}_{\text{AWRS}} = \frac{\alpha\beta(\alpha-1-c(\alpha+\beta-1))+\sqrt{g}}{\alpha^2(1-\alpha+\beta(\Gamma-1)+c(\alpha+\beta-1)(1+\beta(\Gamma-1)))} \tag{4}$$

where g is a function of α, β, Γ, and c, i.e.,

$$g = -(\alpha^2)\beta^2(\Gamma-1+c(\alpha+\beta-1)(\Gamma-1))(1-\alpha(1+\beta) \\ +\beta\Gamma+c(\alpha+\beta-1)(1+\beta\Gamma))$$

We introduce the break-even Γ, Γ_{BE}, which satisfies $R_{\text{AWRS}} = \alpha$. Increasing Γ beyond $\Gamma = \Gamma_{\text{BE}}$ ($\Gamma > \Gamma_{\text{BE}}$) monotonically decreases the attacker's reward R_{AWRS} and therefore further disincentivizes the rational attacker from launching FAW attack. Solving for Γ_{BE} yields:

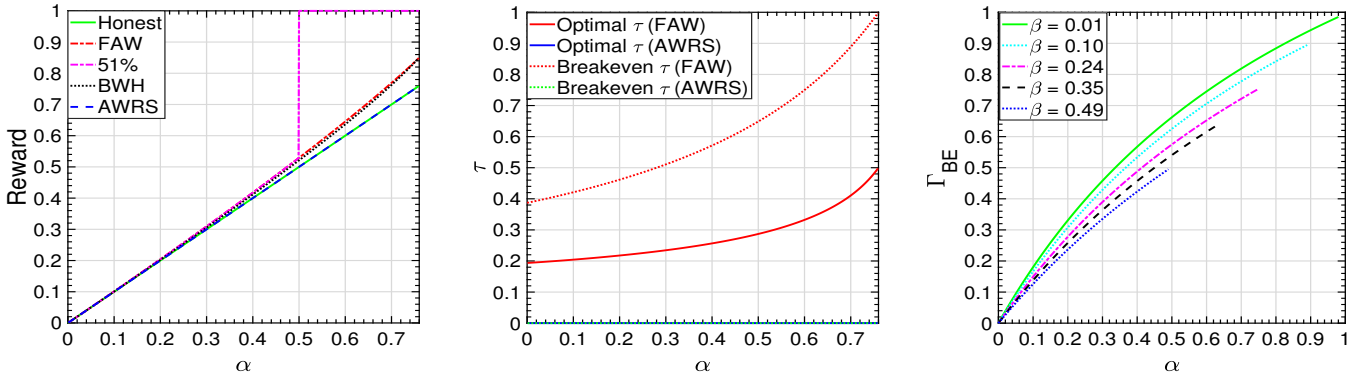
$$\begin{aligned}\Gamma_{\text{BE}} &= \frac{\alpha(\beta+(-1+\alpha-c(\alpha+\beta-1))\tau)}{\beta+\beta c(\alpha+\beta-1)} \\ &= \frac{\alpha}{1+c(\alpha+\beta-1)}\end{aligned} \tag{5}$$

The second equality assumes that $\tau = 0$, i.e., the attacker chooses the optimal strategy for maximizing its reward since he is rational and reward-incentive-driven; the attacker choosing a larger τ will provide smaller reward than honest mining.

For the rest of our analyses, we choose Γ_{BE}, which is the lower bound on the Γ which rids of the incentives of FAW attack (making the attacker's optimal strategy to be honest mining). Reducing Γ lowers the reward variance of the miners in the pool and is therefore desirable, and Γ_{BE} is the lowest for depriving the FAW incentives and making the FAW attack suboptimal compared to honest mining.

VI. Simulations Analyses

In this section, we simulate the game between the attacker and the victim mining pool and study the effectiveness of introducing AWRS against FAW attack. β = 0.24 (which value is derived from the real-world Bitcoin mining in July 2018 and corresponds to the largest pool's computational cap in Bitcoin),



(a) Attacker's reward with respect to α

(b) τ_{BE} and $\hat{\tau}$ with and without AWRS

(c) Γ_{BE} with respect to α

Fig. 1: Simulations analyses results

c = 0.5 (equal probabilities for the FAW attacker to win and lose the forking race), and $\Gamma = \Gamma_{\text{BE}}$. While this section uses these parameter values for simulations, the section builds on the analyses in Section V and holds true in general, i.e., while varying the aforementioned parameters.

Fig. 1a studies the attacker's reward performance using different strategies while varying the attacker's computational power capability α. When AWRS is disabled, FAW attack outperforms BWH attack and honest mining. A 51% attacker also has the complete control over the reward if it is capable of launching the 51% attack, i.e., has the majority of the computational power of the network; the attacker can revoke any blocks that has been mined by others since it controls more power than the rest of the network and can produce a longer chain forcing the protocol-compliant miners to accept the attacker's chain. If AWRS is enabled, then the attacker's performance becomes that of the honest mining and there are no incentives for launching FAW or BWH attacks.

We study the attacker's τ control between the attacker's main pool and the victim pool, i.e., $0 \leq \tau \leq 1$ where the larger the τ the greater the infiltration power on the victim pool. In Fig. 1b, if the victim pool does not enable AWRS, both the $\hat{\tau}$ (discussed in Section III-C) and the τ_{BE} (yielding $R_{\text{FAW}} = \alpha$, i.e., the FAW attack has the same reward as the honest mining) is positive and grows with the attacker's computational capability α. In contrast, with AWRS enabled, the $\hat{\tau}$ and the τ_{BE} become zero and the attacker's optimal strategy is not to use any power on the infiltration of the victim pool, spending all its power on the honest-mining main pool.

Fig. 1c studies the dependency of Γ_{BE} (which is the lower bound on the pool's reward portion to the block submitter for AWRS to deprive the incentives of the FAW attack) on the attacker's computational power (α) and the victim pool's computational power (β). As α grows, the bound in Γ_{BE} grows (in fact, $\Gamma_{\text{BE}} \geq \alpha$ in general) and, as β grows, Γ_{BE} reduces (providing greater flexibility in Γ control in AWRS).

VII. Conclusion

AWRS prevents FAW attack and BWH attack by depriving their incentives; the attacker is better left off mining honestly. We analyze the critical parameter control in Γ, which is the pool's reward to the block submitter, which forces the optimal attacker to mine honestly. In AWRS, the Γ control is independent to the attacker's infiltration control in τ because the attacker's diverging from honest mining and using greater τ only reduces its reward. In addition to AWRS being oblivious to the attacker's control, AWRS only incurs changes in the mining pool manager, providing a practical solution for implementation and adoption.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

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

[3] S. Bag and K. Sakurai, "Yet another note on block withholding attack on bitcoin mining pools," in International Conference on Information Security. Springer, 2016, pp. 167-180.

[4] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "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. ACM, 2017, pp. 195-209.

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

[6] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, "On power splitting games in distributed computation: The case of bitcoin pooled mining," in Computer Security Foundations Symposium (CSF), 2015 IEEE 28th. IEEE, 2015, pp. 397-411.

[7] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden, "Incentive compatibility of bitcoin mining pool reward functions," in International Conference on Financial Cryptography and Data Security. Springer, 2016, pp. 477-498.

[8] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," Communications of the ACM, vol. 61, no. 7, pp. 95-102, 2018.

[9] K. Nayak, S. Kumar, A. Miller, and 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.

[10] V. Buterin. Selfish Mining: A 25% Attack Against the Bitcoin Network. [Online]. Available: https://bitcoinmagazine.com/articles/selfish-mining-a-25-attack-against-the-bitcoin-network-1383578440/

[11] R. Zhang and 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.

[12] Blockchain Hashrate Distribution (Bitcoin). [Online]. Available: https://www.blockchain.com/en/pools.

[13] A. Fisch, B., Pass, R. and Shelat, A. "Socially Optimal Mining Pools," in International Conference on Web and Internet Economics. Springer, 2017, pp. 205-218.

[14] S.Y. Chang, Y. Park, S. Wuthier, and C. Chen, "Uncle-block attack: Blockchain mining threat beyond block withholding for rational and uncooperative miners," in International Conference on Applied Cryptography and Network Security (ACNS), 2019.