The Tragedy of the Miners

Vijay Banerjee, Ryan Rabinowitz, Mark Stidd, Rory Lewis, Philip N. Brown, Gedare Bloom
University of Colorado Colorado Springs
1420 Austin Bluffs Pkwy, Colorado Springs, CO, 80918, USA
vbanerje@uccs.edu, rrabinow@uccs.edu, mstidd2@uccs.edu, rlewis5@uccs.edu, pbrown2@uccs.edu, gbloom@uccs.edu

***Abstract*—In a network of mining pools that secure Bitcoin-like blockchains, it is known that a self-interested mining pool can dishonestly siphon off another pool's mining rewards by executing a block withholding (BWH) attack. In this paper, we show that a BWH attack is always unprofitable for an initial startup period which is at least one difficulty retarget interval (approximately 14 days for Bitcoin). Furthermore, we prove that the payback period to recoup this initial startup cost is always at least as long as the initial unprofitable startup interval, and we show numerically that it can be substantially longer. Thus, the decision of whether or not to execute a BWH attack is not a dominant strategy, and the so called Miner's Dilemma is not in fact a dilemma.**

I. Introduction

Open blockchain systems such as the Bitcoin network are extremely competitive with rewards for each participant (miner) that are directly related to their computational power. Here, the reward for miners is calculated based on a proof-of-work (PoW) mechanism [1] in which the computational resource expended by a miner is used to solve a computationally difficult problem. When a miner submits a solution to this problem (called a *full PoW*), the Bitcoin protocol rewards the miner with a *block reward*. The Bitcoin protocol adjusts the difficulty of the problem in proportion to the total computational power of miners, making it challenging for individual miners to earn profits. To increase the chance of earning a reward, it is common for individuals to join their computational hashing power in a group known as a mining pool. The pool allows each miner to receive rewards more regularly than the sporadic rewards they would earn mining on their own. With several mining pools competing for the reward, the pool managers prefer to stay open to any interested miner to increase the total hashing power of the pool.

Since generating a complete (full) PoW is difficult for individual miners, the mining pools use *partial* PoW to assess the hashing power of an individual miner. A partial PoW is identical to a full PoW except that it has an easier difficulty value than the network's requirements. By counting and verifying the partial PoW a miner submits, the pool can estimate how much mining power (hashrate) the miner is contributing to the pool and pay them accordingly. However, open mining pools are exposed to malicious miners who might join the pool and, while appearing to contribute to its total hashrate, withhold any valid full PoW they find. Such block withholding (BWH) attacks have been well-studied [2], [3]. In a BWH attack, a miner generates valid partial PoW but withholds any full PoW it might create [4]. This attack allows malicious miners to steal from a pool's legitimate miners, lowering the profitability of a pool. The malicious miners appear to be legitimate members of the pool and therefore get a share of the total rewards generated by the pool.

This work is supported in part by NSF grants OAC-2001789, CNS-2046705, ECCS-2013779, and Colorado State Bill 18-086.

The BWH vulnerability poses an interesting question to managers of mining pools: should they conduct an attack on other pools to increase their revenue, or should they stay honest and keep earning from their own computational resources? This question has been framed as the *Miner's Dilemma*, and it has been suggested that a dominant-strategy equilibrium exists in which all pools attack each other, despite all being better-off under no attacks [5].

However, the dominant-strategy equilibrium claim of [5] holds only when each pool manager can dynamically choose whether to comply and stay honest, or to defect by attacking another pool. Also, their rewards must immediately increase with the choice they make. However, in the case of the Bitcoin network, a pool's revenue does not immediately increase when the pool initiates an attack. Since the attacking miners do not submit full PoW, the Bitcoin protocol is unaware of their existence and eventually reduces the *difficulty* target in response to the apparent decrease in total network hashrate. The attack is only potentially profitable *after* this difficulty adjustment takes place.

In Bitcoin's distributed network, the difficulty-target value is adjusted every 2016 blocks (or published full proofs of work) in such a way that it takes approximately 10 minutes for the network to find a new full PoW [6]. Accordingly, it takes approximately two weeks after a BWH attack is initiated before the difficulty is adjusted to account for the reduction in effective hashing power.

In this paper, we show that the attacker suffers a loss relative to honest mining during at least the initial two weeks of an attack within the Bitcoin network, and that this loss is recouped slowly. Due to the high variance in total network hashrate and block discovery, this high initial cost and slow climb to profitability acts as a natural deterrent to mining pools launching a BWH attack. In summary, our contributions are:

1) It is costly to launch a BWH attack. Proposition 1 proves that for at least the first 14 days of an attack, an attacker suffers a revenue loss relative to honest mining.

978-1-6654-3161-3/22/$31.00 ©2022 IEEE

2) These initial costs are recouped more slowly than they are accrued. When the attack becomes profitable after a difficulty adjustment, Proposition 2 proves that the attacker's paypack period is at least as long as (and often several times longer than) the initial costly interval. That is, the daily losses prior to the adjustment are always greater than the daily profits after the adjustment.
3) We derive the revenue generated by each pool when an arbitrary number of pools are executing a BWH attack, both before and after a difficulty adjustment.

II. Related Work

Rosenfeld [4] analyzes different reward systems mining pools can use and introduces the concept of BWH to disrupt mining pool reward systems. Courtois and Bahack [7] formalize attacker strategies where the attacker would gain revenue at the expense of other miners, which includes BWH attacks. The authors discuss how such attackers can optimize their profits, but do not indicate that honest mining is a more profitable solution than withholding. Eyal [5] formulates BWH as a game and shows that (i) a *simplified version* of the two-pool game is an instance of the iterated prisoner's dilemma, (ii) any number of identical pools mutually attacking each other is a tragedy of the commons equilibrium, and (iii) not attacking is not a Nash equilibrium for the general case with any number of pools. Subsequent work further examines the game-theoretic formulation of BWH [8]–[15] and proposes variations of BWH strategies and countermeasures [2], [16]–[18]. The key difference between our work and the prior work is that we consider the delay between commencing an attack and when the difficulty adjustment occurs; indeed, prior formulations explicitly assume the network difficulty is fixed and stable, despite demonstration by Eyal [5] that the attacker's reward is not improved until after the difficulty is adjusted.

III. Analyzing BWH Attack Profitability

In this section we analyze attack revenues in different cases and compare these to honest mining revenues. We first show the potential revenue from honest mining and then how the revenue changes when one pool attacks a victim pool both before and after a difficulty adjustment. We further analyze the situation of mining pools attacking each other and generalize it into N mining pools.

The notation used in the following sections is summarized in Table I. We suppose that mining pool i has hashing power h_i, the fraction of its hashing power that pool i sends to pool j for a BWH attack is $x_{i,j}$, and the total hashing power of the whole Bitcoin network is $H \coloneqq \sum_{i=1}^{N} h_i$. Furthermore, we assume that attacking pools initiate attacks at the start of a difficulty adjustment interval so as to maximize the attack's effect on the adjustment algorithm.

TABLE I: Table of notation

Variable	Definition
i	Pool ID
$x_{i,j}$	Fraction of hashing power used by pool i to attack pool j
H	Total hashrate of the whole network
H'	Total hashrate of the whole network after BWH attack
h_i	Hashing power of pool i
R	Average daily revenue of the whole network
R_i^h	Average daily revenue of pool i if pool i is honest
R_i	Average daily revenue of attacking pool i before difficulty adjustment
R_i'	Average daily revenue of attacking pool i after difficulty adjustment
A	Adjustment factor for average daily revenue before difficulty adjustment
B	Block reward
T	Number of days to find 2016 blocks
P_i	Payback period of attacking pool i

A. Honest mining

In the case of honest mining (i.e., $x_{i,j} = 0$ for all i, j), mining pools generate revenue based on the blocks that they find. We denote a block reward by B, and—at the time of writing—each block yields a reward of $B = 6.25$ Bitcoins (BTC). When the difficulty target is exactly proportionate to the network hash rate, it takes 10 minutes on average for a single block to be found on the network. Thus, the fractional rate at which rewards are found on the network is $0.1B$/minute. Taking B as 6.25, the total expected revenue R for 24 hours is

$$\begin{aligned} R &= 0.1 \times B \times 60 \times 24 \\ &= 144 \times B \qquad (1) \\ &= 900 \text{ BTC}. \qquad (2) \end{aligned}$$

Using Eq. (2), pool i's average daily revenue from honest mining, denoted R_i^{h}, can be derived as:

$$R_i^{\mathrm{h}} = \frac{h_i}{H} \times R. \qquad (3)$$

B. When one mining pool is attacking another mining pool

In this case, we have to consider the fact that the difficulty of the blockchain is updated every 2016 blocks. With each block being found at an average target rate of 1 block per 10 minutes, 2016 blocks will be found after approximately 2 weeks. However, in a BWH attack scenario, the time to find 2016 blocks will be longer due to the reduced total effective hashrate, as the attacking miners will be withholding full PoW. Hence, the daily average reward will be earned at a lesser rate. To account for the reduced reward, we will multiply R with an *adjustment factor* A, to reach the effective rate of reward before difficulty adjustment.

In a set up of two mining pools, with pool 1 attacking pool 2, adjustment factor A is calculated as follows:

$$A \coloneqq 1 - \frac{x_{1,2}h_1}{H}. \qquad (4)$$

The average revenues generated per day prior to a difficulty adjustment are thus:

$$R_1 = R \times A \times \left\{ \frac{h_1(1 - x_{1,2})}{H} + \left(\frac{x_{1,2}h_1}{x_{1,2}h_1 + h_2} \right) \left(\frac{h_2}{H} \right) \right\} \qquad (5)$$

$$R_2 = R \times A \times \left\{ \frac{h_2}{H} - \left(\frac{x_{1,2}h_1}{x_{1,2}h_1 + h_2} \right) \left(\frac{h_2}{H} \right) \right\}. \quad (6)$$

Since $x_{1,2} \times h_1$ miners are not actively contributing to the Blockchain network, the expected time in days to find 2016 blocks will be:

$$T \coloneqq \frac{2016 \times 10}{60 \times 24} \times \frac{1}{A} = \frac{14}{A}. \quad (7)$$

After the difficulty adjustment the total hashing power of the Bitcoin network will be:

$$H' = HA. \quad (8)$$

Using (8) we then calculate the generated revenues of each of the pools after the difficulty adjustment.

$$R_1' = R \left\{ \frac{h_1(1 - x_{1,2})}{H'} + \left(\frac{x_{1,2}h_1}{x_{1,2}h_1 + h_2} \right) \left(\frac{h_2}{H'} \right) \right\} \quad (9)$$

$$R_2' = R \left\{ \frac{h_2}{H'} - \left(\frac{x_{1,2}h_1}{x_{1,2}h_1 + h_2} \right) \left(\frac{h_2}{H'} \right) \right\}. \quad (10)$$

We note that (9) and (10) are reported in [5], where it is demonstrated numerically that by carefully selecting attack rate $x_{1,2} > 0$, after a difficulty adjustment an attacker can profit relative to honest mining; i.e., can ensure that $R_1' > R_1^{\mathrm{h}}$.

However, in this paper we explicitly consider the cost an attacker incurs by launching such an attack. First, we show that prior to the difficulty adjustment, the attacker's average daily revenue is lower than that of honest mining, and fully characterize the set of profitable attack rates *following* the difficulty adjustment:

Proposition 1. *If a pool with hashrate h_1 initiates a BWH attack against a pool with hashrate h_2 with attack rate $x_{1,2} > 0$, then prior to the next blockchain difficulty adjustment, the attacker's daily revenue is strictly decreased relative to honest mining:*

$$R_1 < R_1^{\mathrm{h}}. \quad (11)$$

After the difficulty adjustment, an attacking pool is profitable if and only if:

$$x_{1,2} < \frac{h_2}{H - h_1}. \quad (12)$$

Proof. First we show (11). From (3) and (5) we obtain

$$\begin{aligned} R_1^{\mathrm{h}} - R_1 &= \frac{R}{H} \left(h_1 - A \left(h_1 + \frac{x_{1,2}h_1h_2}{x_{1,2}h_1 + h_2} - x_{1,2}h_1 \right) \right) \\ &= \frac{Rh_1}{H} \left(1 - A \left(1 - x_{1,2} \left(1 - \frac{h_2}{x_{1,2}h_1 + h_2} \right) \right) \right) \\ &\geq 0, \end{aligned} \quad (13)$$

where the inequality holds because $A \leq 1$, $x_{1,2} \leq 1$, and $h_2 \geq 0$.

Now we show that an attack is profitable after the difficulty adjustment if and only if (12) holds by manipulating the desired inequality on (9) and (3):

$$\begin{aligned} & R_1' > R_1^h \\ \iff & R \left\{ \frac{h_1(1 - x_{1,2})}{H'} + \frac{x_{1,2}h_1}{x_{1,2}h_1 + h_2} \left(\frac{h_2}{H'} \right) \right\} > \frac{h_1}{H} \times R \\ \iff & \frac{1 - x_{1,2}}{AH} + \frac{x_{1,2}h_2}{(AH)(x_{1,2}h_1 + h_2)} > \frac{1}{H} \\ \iff & \frac{h_2}{H - h_1} > x_{1,2}, \end{aligned} \quad (14)$$

where the last step is obtained by substituting (4) for A. □

Thus, an attacker must sustain an average daily loss of $R_1^{\mathrm{h}} - R_1 > 0$ for T days before realizing an average daily profit of $R_1' - R_1^{\mathrm{h}}$. Hence, if the condition in (12) is not satisfied, the attacking pool will never be able to *break even*. If condition (12) is satisfied, then how long after the difficulty adjustment must the attacker continue the attack to cover this initial cost? Pool 1's *payback period*, defined as the number of profitable days required to recoup the attacker's initial losses, is given by

$$P_1 \coloneqq \frac{R_1^{\mathrm{h}} - R_1}{R_1' - R_1^{\mathrm{h}}} \times T. \quad (15)$$

We are now ready to present our main result reporting and lower-bounding the payback period for a lone BWH attacker.

Proposition 2. *If a pool with hashrate[1] $h_1 \leq H/2$ initiates a BWH attack against a pool with hashrate h_2 and the initial difficulty adjustment interval is T, then the attacker's payback period is*

$$P_1 = \frac{(h_2 + x_{1,2}(H + h_1) - x_{1,2}^2h_1)(H - x_{1,2}h_1)}{(x_{1,2}(h_1 - H) + h_2)H} T \quad (16)$$

$$\geq T. \quad (17)$$

Before we present the proof, we note that while the payback period is always at least T (that is, an attacking pool's costs accumulate at least as fast as their profits), the payback period can be considerably longer. For instance, if $H = 1$, $h_1 = h_2 = 0.4$, the attack fraction to yield an optimal revenue is $x_{1,2} \approx 1/3$ and the payback period is approximately $3.56T$, or 57.5 days as $T = 16.1538$ (from Eq. 7). This example indicates that for substantial attacks, the attacker may need to sustain the attack for multiple months before any profit is realized. Figure 1 depicts the payback period from a profitable attack by pool 1. Substituting values in (12), we get that the attack will be profitable if $x < 0.667$. Note that the payback period goes over 4000 days when the attacking fraction is 0.66. Moreover, the total attack period is $P_1 + T$ as shown in Figure 2, meaning the total attack period is at least $2T$.

Proof. Expression (16) can be derived by substituting (3), (4), (5), (8), and (9) into (15). The derivation

[1]Note that $h_1 \leq H/2$ is a standard security assumption in Bitcoin; this means that pool 1 is not powerful enough to launch a 51% attack on the Bitcoin network.

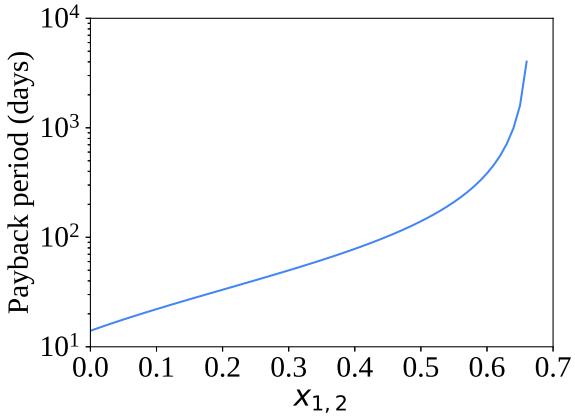


Fig. 1: Payback period (in days) as a function of the attacking fraction $x_{1,2}$, illustrating the results of Proposition 2, with $h_1 = h_2 = 0.4$ and $x_{1,2} < \frac{h_2}{H-h_1} = 0.667$.

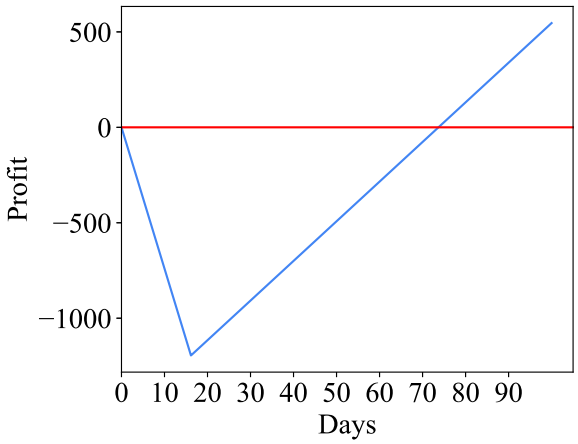


Fig. 2: Break-even point for an attack scenario where $H = 1$, $h_1 = h_2 = 0.4$, and $x_{1,2} = 1/3$. Note, the total time an attack must execute is $T+P$, as the attack is not yet profitable during initial period of T days

is straightforward and tedious, so we omit it here. Henceforth in the proof, we write $x = x_{1,2}$ to simplify the notation. To show the lower bound (17), we first expand and simplify (16) to obtain

$$P_1/T = \frac{H(h_2 + xh_1) + H^2x - 2Hx^2h_1 - xh_1h_2 - (xh_1)^2(1-x)}{H(h_2 + xh_1) - xH^2}. \tag{18}$$

Where we denote the numerator and denominator of (18) as N and D respectively. It can be shown that in a profitable attack, $D \geq 0$; thus, to show the desired bound, it suffices to show that $N \geq D$:

$$\begin{aligned} N - D &= x\left(2H(H - xh_1) - xh_1h_2 - (xh_1)^2(1-x)\right) \\ &\geq x\left(2H(H - xh_1) - xh_1h_2 - (xh_1)^2\right) && (19) \\ &\geq x(Hh_2 + Hh_1 - h_1h_2 - xh_1^2) && (20) \\ &= x(h_2(H - h_1) + h_1(H - xh_1) \geq 0. && (21) \end{aligned}$$

Inequality (19) holds since $x \geq 0$ and (20) holds since $H - xh_1 \geq H - h_1 \geq h_2$ and $h_1 \leq H/2$ (implying $H - h_1 \geq h_1$). Having shown $N \geq D$, this concludes the proof. □

C. A network of N mining pools

In this section we generalize the equations for N pools where all pools can be attacking and victim pools. The average daily revenue generated before difficulty adjustment is:

$$R_i = R \times A \times \left\{ \frac{h_i(1 - \sum_{j\neq i} x_{i,j})}{H} + \sum_{j\neq i}\left[\left(\frac{x_{i,j}h_i}{x_{i,j}h_i + h_j}\right)\left(\frac{h_j}{H}\right)\right] - \sum_{j\neq i}\left[\left(\frac{x_{j,i}h_j}{x_{j,i}h_j + h_i}\right)\left(\frac{h_i}{H}\right)\right]\right\}, \tag{22}$$

where

$$A := 1 - \frac{\sum_{j\neq i} x_{i,j}}{H}. \tag{23}$$

Similarly, the revenue after difficulty adjustment will be:

$$R_i' = R\left\{ \frac{h_i(1 - \sum_{j\neq i} x_{i,j})}{H'} + \sum_{j\neq i}\left[\left(\frac{x_{i,j}h_i}{x_{i,j}h_i + h_j}\right)\left(\frac{h_j}{H'}\right)\right] - \sum_{j\neq i}\left[\left(\frac{x_{j,i}h_j}{x_{j,i}h_j + h_i}\right)\left(\frac{h_i}{H'}\right)\right]\right\}, \tag{24}$$

where H' is:

$$H' := H - \sum_{i=0}^{N}\sum_{j=0}^{N} x_{i,j}. \tag{25}$$

D. A network of N mining pools with fees

Here, we extend (22) and (24) to account for pool fees. Some notable pools such as BTC.com [19] and Slush [20] charge a percentage-based fee on the profits of their miners. Accordingly, the infiltration revenue of a BWH attack is subject to this fee. Let f_i be the fractional fee of pool i where $f_i \in [0, 1]$, (22) becomes:

$$R_i = R \times A \times \left\{ \frac{h_i(1 - \sum_{j\neq i} x_{i,j})}{H} + \sum_{j\neq i}\left[\left(\frac{x_{i,j}h_i}{x_{i,j}h_i + h_j}\right)\left(\frac{h_j}{H}\right)(1 - f_j)\right] - \sum_{j\neq i}\left[\left(\frac{x_{j,i}h_j}{x_{j,i}h_j + h_i}\right)\left(\frac{h_i}{H}\right)(1 - f_i)\right]\right\}. \tag{26}$$

Similarly, the revenue after difficulty adjustment (24) becomes:

$$R'_i = R\Bigg\{\frac{h_i(1-\sum_{j\neq i} x_{i,j})}{H'} + \sum_{j\neq i}\left[\left(\frac{x_{i,j}h_i}{x_{i,j}h_i+h_j}\right)\left(\frac{h_j}{H'}\right)(1-f_j)\right] - \sum_{j\neq i}\left[\left(\frac{x_{j,i}h_j}{x_{j,i}h_j+h_i}\right)\left(\frac{h_i}{H'}\right)(1-f_i)\right]\Bigg\}. \quad (27)$$

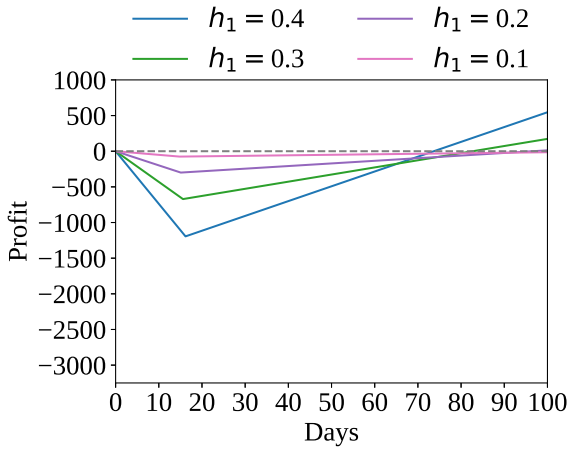
IV. Discussion

In this section we analyze the revenue outcome and the break-even points for arbitrary values of attacking pool hashrate, and attacking fractions. Figure 3a shows the effect of varying the attacker's share of the network hashrate h_1 while keeping h_2 and $x_{1,2}$ constant. As h_1 varies from $0.1-0.4$, the total number of days to reach difficulty adjustment T shifts from 14.48 days to 16.15 days. Interestingly, when h_1 increases past the value of h_2, the break-even point reduces from past 100 days to 73.71 days. These trends indicate that although the cost of the attack is proportional to the ratio h_1/h_2, the attacker can recoup their losses quicker.

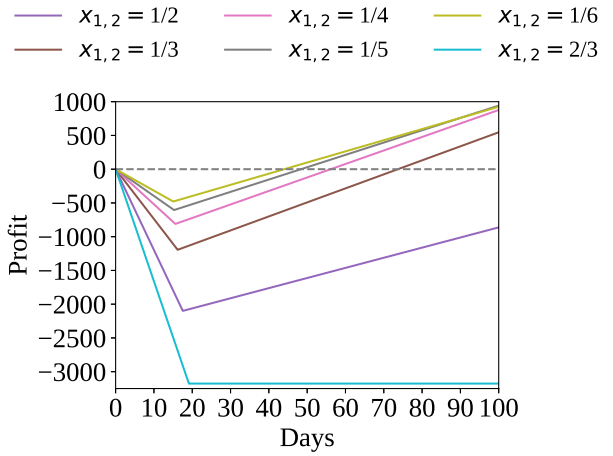
Figure 3b shows the consequences of an attacker devoting various portions of its hashing power to an attack. As $x_{1,2}$ increases from $1/6-2/3$, the initial cost of the attack increases as the slope of the attacker's profitability decreases after T days. An important exception to this trend is when $x_{1,2}=1/6$, as opposed to $x_{1,2}=1/5$, the slope of the profitability line after T days for $x_{1,2}=1/5$ is greater than when $x_{1,2}=1/6$. This implies there is a maximum optimal value of $x_{1,2}$ for this attack. From the plot it is clear the optimal value for $x_{1,2}$ is above $1/6$, but the only visually discernible maximum limit for an optimal $x_{1,2}$ value is below $1/2$ as its slope after T is less than the slope of $x_{1,2}=1/3$. Additionally, the slope for $x_{1,2}=2/3$ demonstrates the right hand bound of the attacking fraction as presented in Proposition 2. We observe that when $x_{1,2}=2/3$, the slope of the line is 0 meaning an attacker will never recover the initial cost of an attack.

Figure 3c demonstrates a situation where the hashrate of the victim pool decreases by 50% of its hashing power at T days into being attacked. Such a reduction in victim hashrate could be caused by factors which are not in the hands of the attacking pool, such as miners in the victim's pool realizing they are not being paid accordingly to their hashrate, as the victim pool is less profitable when attacked. If the victim pool continues to operate with 50% of its original hashing power, the attacker will never turn a profit as the slope of the profitability line is ≤ 0. Accordingly, if the attacker terminates the attack after T days, they will never recoup their loss by mining honestly. This is an example of the uncertainty that can make a BWH attack unprofitable, even if the attacking pool had an initially promising attack scenario.

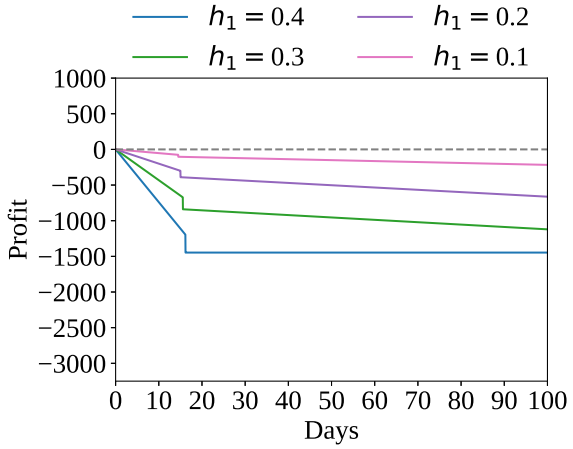
The observations from the plots show that an attack is not always profitable, and the break-even point can greatly vary depending on factors such as the size of the attacking pool and



(a) The profitability varies based on the size of the attacking pool. Here, $h_1 = \{0.1, 0.2, 0.3, 0.4\}, h_2 = 0.4$, and $x_{1,2} = 1/3$. The attack is not profitable until $T + P$ days.



(b) The attack profitability depends on the attack fraction. In this plot, $h_1 = h_2 = 0.4$, and $x_{1,2} = \{2/3, 1/3, 1/4, 1/5, 1/6\}$. The attack becomes unprofitable when $x_{1,2} > 1/3$.



(c) With $h_1 = \{0.1, 0.2, 0.3, 0.4\}, h_2 = 0.4$, and $x_{1,2} = 1/3$, $h2$ loses 50% of its hashing rate T days after the attack begins, and attacking is not profitable.

Fig. 3: Break-even points for attack scenarios with $H = 1$ and varying h, x.

the attacking fraction. When the profitability of the attack is taken into consideration, an attack is a non-dominant strategy, because the total attack period to break even, i.e., $P + T$, is at least four weeks—as proved in Proposition 2. Hence, the decision to attack another pool is not a dilemma, rather it is always in the favor of the mining pools to avoid attacking another pool with a BWH attack due to the uncertainties involved with the high recovery period for the attack.

V. Conclusion

We have analyzed the transient effects of a BWH attack taking into account the difficulty adjustment, which makes a profitable attack a gradual long-term process. We showed that the time to recover from the cost incurred in two weeks is always at least two additional weeks of constant attacking, and is often considerably more. We further show that the initial time to difficulty adjustment from the start of the attack is affected by increasing h_1 as well as increasing $x_{1,2}$. Our analysis shows that the two-player equilibrium of the Miner's Dilemma does not hold when revenue density of a pool does not immediately increase with an attack. Future work could extend our analyses to incorporate more dynamic and transient effects among pools, such as miner migrations, transitive and reflexive attack relationships (mutually assured destruction), and global network losses.

References

[1] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Advances in Cryptology — CRYPTO' 92*, E. F. Brickell, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 139–147.

[2] S. Bag, S. Ruj, and K. Sakurai, "Bitcoin block withholding attack: Analysis and mitigation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1967–1978, Aug 2017.

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

[4] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," *CoRR*, vol. abs/1112.4980, 2011.

[5] I. Eyal, "The miner's dilemma," in *2015 IEEE Symposium on Security and Privacy*, May 2015, pp. 89–103.

[6] A. Lamiri, K. Gueraoui, and G. Zeggwagh, "Bitcoin difficulty, a security feature," in *Information Systems and Technologies to Support Learning*, Á. Rocha and M. Serrhini, Eds. Cham: Springer International Publishing, 2019, pp. 367–372.

[7] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," *CoRR*, vol. abs/1402.1718, 2014.

[8] 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 *2015 IEEE 28th Computer Security Foundations Symposium*, July 2015, pp. 397–411.

[9] A. Laszka, B. Johnson, and J. Grossklags, "When bitcoin mining pools run dry," in *Financial Cryptography and Data Security*, M. Brenner, N. Christin, B. Johnson, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 63–77.

[10] N. Karpova, "Game-theoretic analysis of a block withholding attack on the bitcoin consensus protocol," 2019.

[11] A. Toroghi Haghighat and M. Shajari, "Block withholding game among bitcoin mining pools," *Future Generation Computer Systems*, vol. 97, pp. 482–491, 2019.

[12] S. Kim and S.-G. Hahn, "Mining pool manipulation in blockchain network over evolutionary block withholding attack," *IEEE Access*, vol. 7, pp. 144 230–144 244, 2019.

[13] Y. Wang, C. Tang, F. Lin, Z. Zheng, and Z. Chen, "Pool strategies selection in pow-based blockchain networks: Game-theoretic analysis," *IEEE Access*, vol. 7, pp. 8427–8436, 2019.

[14] D. Wu, X. dong Liu, X. bin Yan, R. Peng, and G. Li, "Equilibrium analysis of bitcoin block withholding attack: A generalized model," *Reliability Engineering & System Safety*, vol. 185, pp. 318–328, 2019.

[15] W. Li, M. Cao, Y. Wang, C. Tang, and F. Lin, "Mining pool game model and nash equilibrium analysis for pow-based blockchain networks," *IEEE Access*, vol. 8, pp. 101 049–101 060, 2020.

[16] S. Bag and K. Sakurai, "Yet another note on block withholding attack on bitcoin mining pools," in *Information Security*, M. Bishop and A. C. A. Nascimento, Eds. Cham: Springer International Publishing, 2016, pp. 167–180.

[17] Y. Velner, J. Teutsch, and L. Luu, "Smart contracts make bitcoin mining pools vulnerable," in *Financial Cryptography and Data Security*, M. Brenner, K. Rohloff, J. Bonneau, A. Miller, P. Y. Ryan, V. Teague, A. Bracciali, M. Sala, F. Pintore, and M. Jakobsson, Eds. Springer International Publishing, pp. 298–316.

[18] R. Qin, Y. Yuan, and F.-Y. Wang, "Optimal block withholding strategies for blockchain mining pools," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 709–717, June 2020.

[19] "The fee, settlement mode and payment threshold in btc.com pool," 2021.

[20] "Slush rewards, payout and fees," 2021.