Mind the Mining

GUY GOREN*, Technion IIT
ALEXANDER SPIEGELMAN, VMware Research

In this paper we revisit the mining strategies in Proof of Work based cryptocurrencies and propose two strategies, which we call *smart* and *smarter mining*, that in many cases strictly dominate honest mining. In contrast to other known attacks, such as selfish mining, which induce zero-sum games among the miners, the strategies proposed in this paper increase miners' profit by reducing their variable costs (i.e., electricity). Moreover, the proposed strategies are viable for much smaller miners than previously known attacks and, surprisingly, an attack launched by one miner can be profitable for all other miners as well.

While saving electricity is very encouraging for the environment, it may affect the coin's security. The smart and smarter mining strategies expose the coin to under 50% attacks, and this vulnerability might only grow when new miners join the coin in response to the increased profit margins induced by these strategies.

CCS Concepts: • **Security and privacy** → **Distributed systems security**; **Economics of security and privacy**; • **Theory of computation** → **Algorithmic game theory and mechanism design**.

Additional Key Words and Phrases: blockchain, cryptocurrency, proof of work, mining strategies, game theory.

ACM Reference Format:
Guy Goren and Alexander Spiegelman. 2019. Mind the Mining. In *The 20th ACM conference on Economics and Computation (EC '19), June 24–28, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3328526.3329566

1 INTRODUCTION

As of the end of 2018, the total cryptocurrency market cap is above 100 billion US Dollars. Dozens of new coins emerge every month and the industry of digital mining is blooming. According to [2], the vast majority of the coins are based on the PoW technology [5], which received a lot of attention with the introduction of Bitcoin [15]. The main idea is that a lot of power has to be wasted in order to change the coin's state. This makes Sybil attacks impossible, and enables reaching consensus in an anonymous open network. The drawback of this technology is the huge amount of electricity it consumes. As of 2018, Bitcoin mining alone consumes more electricity than 159 countries including Nigeria and Morocco [1]. Apart from not being environmentally friendly, the huge waste of power induces very high costs on coin maintenance.

The main entities in a cryptocurrency system are the miners. They maintain the state and preserve the security by performing work that requires a lot of power, and in return they get to mint new coins. For economical and security reasons, cryptocurrencies try to enforce a fixed rate of new minted coins.[1] This is done by determining how much power has to be invested by a miner in order to

*The work of Guy Goran was supported in part by a grant from the Technion Hiroshi Fujiwara cyber security research center and the Israel cyber bureau.

[1]Presently, for example, the Bitcoin systems [15] tries to enforce miners to collectively mint 75 new coins every hour.

EC '19, June 24–28, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6792-9/19/06...$15.00
https://doi.org/10.1145/3328526.3329566

mint one coin, which is usually called the mining *difficulty*. Ideally, if the total mining power (by all miners) invested in a coin would be known at any given time, then the difficulty could be accurately calculated and a fixed minting rate could be enforced. However, this is not the case. Miners can freely join and leave the system at any time, and are free to stop mining when it is not profitable for them.

Therefore, the best that cryptocurrency systems can do is to predict the future based on an estimation of the total mining power in the past. This is done by dividing executions into epochs, where each epoch consists of a fixed number of minted coins. (For example, in Bitcoin [15], each epoch consists of 2016 blocks, and each block mints 12.5 coins - as of 2018.) The minting difficulty for each epoch is calculated based on the estimation of the total mining power in the previous epoch. This means that coins can adjust to changes in the total mining power only during epoch changes, which makes them vulnerable to sudden changes in total mining power. This vulnerability has been already noticed before as one that can lead to a problem called "blockchain death spiral" [3], in which miners suddenly leave a coin (possibly because its value suddenly dropped), leaving it with high difficulty and forcing a long epoch. This can lead to serious throughput decrease in strong coins, and to a total death of small ones. In this paper we further explore this vulnerability and show how miners can exploit it for their benefit. In particular, we show that the "desired equilibrium" in which miners always mine with their full power is not an equilibrium. Surprisingly, we show that in many cases stopping to mine and being idle is a strictly better strategy. The basic concept is based on the fact that the total mining power in each epoch determines the difficulty, and thus also the revenue of miners, in the next epoch. Therefore, if a miner does not mine during an epoch, it loses the revenue of this epoch, but it saves the cost of the power in this epoch and may gain more revenue in the next epoch due to the difficulty adjustment.[2] We call the strategy in which a miner alternately mines in one epoch and then remains idle in the following epoch a *smart mining* strategy, where epochs in which the miner mines and epochs in which it does not mine are called *high revenue epochs (HRE)* and *low revenue epochs (LRE)*, respectively. See Figure 1 for an illustration. Interestingly, the benefit of the smart miner strategy does not come at the expense of other miners. On the contrary, other miners benefit from it even more since they lose nothing in low revenue epochs and gain in high revenue epochs. With this in mind, we show that in some cases miners benefit the most from mining with only part of their mining power. We call this strategy *smarter mining*. The smarter mining strategy can be seen as an optimization of smart mining.

Note that while smart and smarter mining are a win-win for all miners and the environment, the coin security is compromised during low revenue epochs. Not only that a smart miner does not mine in low revenue epochs, it might be the case that other miners gain from joining him. This can leave the coin exposed to attacks (during low revenue epochs) by a malicious miner that controls less than 50% of the total mining power. The smart and smarter mining strategies increase the profit margins in high revenue epochs, which might bring new miners to the coin, which in turn could be expected to restore the coin's security. However, surprisingly, we get exactly the opposite effect. When new miners join the high revenue epochs, the difficulty in the low revenue epoch goes higher, and thus the revenue per time unit in the low revenue epochs decreases, which might cause miners to leave these epochs and expose the coin to additional attacks.

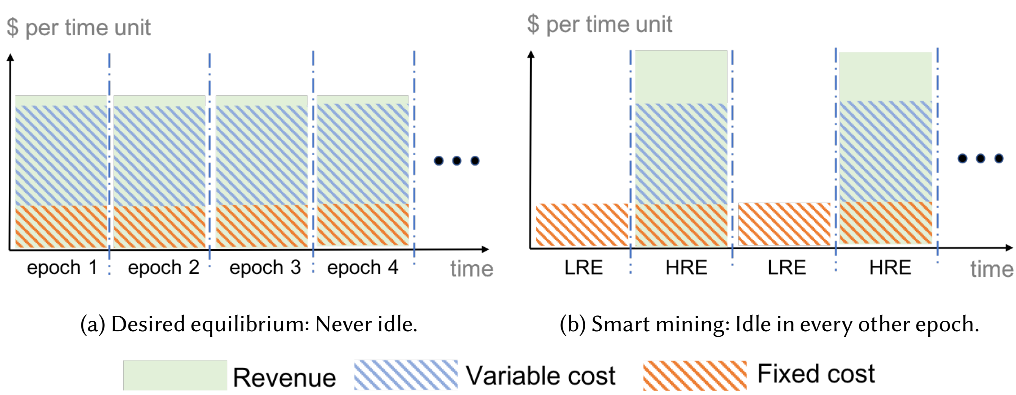
Analysing the strategic behavior of miners in cryptocurrency systems has received considerable attention in the last few years [6, 7, 9–14, 16, 18, 19, 21, 22]. The pioneering "Selfish-Mining" attack strategy demonstrated that deviating from the mining protocol can be beneficial even without a majority of the mining power [10]. Their strategy, however, requires 25% or more of the mining power, which is relatively high. Smart mining, on the other hand, is relevant for smaller miners as

[2]In practise there is a maximum factor by which the difficulty can change between two consecutive epochs. However, it does not invalidate our attack, it only adds another parameter for our analysis, which for simplicity and readability we choose to omit in this paper.

well. For example, Section 4 shows that if the fixed costs represent 10% of the miner's total costs, having just 12% of the mining power suffices. Moreover, the long term effect of allowing new players to join the system exposes significant differences between our strategies and selfish mining. While in selfish mining the joining of new players restores the system's security, in smart mining the opposite happens. Players that join the coin for economical reasons, unintentionally further damage the coin's security and increase its vulnerability to under 50% attacks.

In [7], Carlsten et al. showed that selfish mining can be made profitable for a miner with a low hash power share in a model in which miners get paid by transaction fees rather than by minted coins. In the same model, Tsabary and Eyal [22], showed that miners can increase their profit by not mining (being idle), and thus reduce their electricity costs, when the total amount of available transaction fees is low. In this paper we consider the more standard model that is currently used in practice, in which miners are paid in newly minted coins. To the best of our knowledge, we are the first to propose a dominating strategy that is beneficial to all miners (i.e., the attacker and honest miners), but decreases the coin security.

The rest of the paper is organized as follows: In section 2 we give an overview on PoW-based cryptocurrencies and in section 3 we define our model. Section 4 introduces and analyse the smart and smarter mining strategies, and in Section 5 we analyse the other miners best response and discuss the implication on the coin security. Section 6 provides a discussion of our results and considers future work.



(a) Desired equilibrium: Never idle.

(b) Smart mining: Idle in every other epoch.

Fig. 1. A miner's profit in an epoch is the revenue minus the costs, which is the uncovered green area. In the desired equilibrium, the profit is the same in every epoch. In smart mining, the miner pays the fixed cost and gains no profit in LRE epochs, but gains large profits in HRE epochs. If the profit bonus in HRE epochs is bigger than the fixed cost plus the epoch profit in the desired equilibrium strategy, then the smart mining strategy is more profitable for the miner.

2 PROOF OF WORK OVERVIEW

In the next section we define a model that aims to capture the core of the Proof of Work (PoW) based cryptocurrencies (e.g., Bitcoin [15] and Ethereum [4]) mechanism that we need to demonstrate our attack. In this section we give a short and simplified description of how it works. PoW, which in the context of cryptocurrencies was first introduced in Bitcoin [15], is a novel approach to solve randomized synchronous anonymous byzantine leader election. The idea is that there is a puzzle known to all that all parties try to solve by performing hash operations, where the difficulty of the puzzle determines the probability for a single hash operation to solve the puzzle. The first party that manages to solve the puzzle broadcasts the solution, and as a result it is elected as the leader. Parties are usually called miners, and the number of hash functions that a miner can perform in a time unit is called her mining power. An important property of PoW, which makes it useful for cryptocurrencies, is the fact that the probability of a miner to be the first to solve the puzzle and become the leader is equal to the ratio of the miner's mining power out of the total mining power of all miners.

Most of the PoW-based cryptocurrency systems use PoW in the following (simplified) way. They start from a commonly known genesis block that determines the first puzzle. Once a solution to the k^{th} puzzle is found, the chosen leader (the party that found the solution) broadcasts it to all parties. The solution forms a block that is added to the block-chain, which in turn determines the next puzzle. However, this basic idea raises several challenges that needed to be addressed:

- First, the system must give incentives for the miners in order to encourage them to participate in the protocol (e.g., solve the puzzles). To this end, miners get paid, by new coins they mint, when they find solutions.
- Second, a crucial requirement from a PoW-based cryptocurrency is that solutions are broadcast faster than they are found [8]. This is essential in order to reduce the possibility of disagreement on the leader's identity (i.e., forks) - if one miner finds a solution first, but another miner finds a different solution before the first solution was broadcasted, then the identity of the leader may be ambiguous. To overcome this problem (by reducing the probability of such an event), cryptocurrencies try to control the expected rate at which solutions are found. Recall that the difficulty determines the probability of a single hash to solve the puzzle, and since the probability of every hash is independent from the other hashes, if the total mining power is known, then the difficulty can be set to determine the expected rate of solutions.
- The third challenge is how to estimate the total mining power. Recall that in public cryptocurrencies, miners can leave and join the system whenever they want, so there must be a dynamic mechanism to track these changes and adjust the difficulty accordingly. In most of the PoW-based cryptocurrencies this is done in the following way. The execution is divided into epochs, each of which consists of a fixed number B of blocks (puzzle solutions), e.g., $B = 2016$ in Bitcoin. When epoch ep_i is over, the system uses the real time it took for epoch ep_i to complete in order to estimate the total mining power used during this epoch. Then, this estimation is used to calculate a difficulty level for the next epoch.

Note that since the difficulty and the reward (number of new minted coins) for finding a solution are always known, a miner can calculate the expected revenue for every hash it performs, and by taking into account its costs, the miner can estimate the expected profit per hash. The desired equilibrium in PoW-based cryptocurrencies, which is important in order to reason about security, is that all miners always (during all epochs) mine with their total mining power. In this paper we consider the standard demand and supply economic assumption in the desired equilibrium, in which the profits of the miners are negligible, and show that miners have strictly better strategies that compromise the coin security. It is important to note that the better strategies we demonstrate in this paper exist also for an arbitrary ϵ profit, with only minor changes in the numerical results.

3 MODEL AND DEFINITIONS

For simplicity of analysis, we define a deterministic model that captures the core of PoW-based cryptocurrencies. Our model does not use puzzles and a difficulty to determine the probability that a single hash will solve the puzzle. Instead, we deterministically define the revenue each miner gets from a single hash operation and how many hashes are needed in total to complete an epoch. Note that by defining a deterministic model we give the system more control and thus our results apply for the real probabilistic case as well. Our model consists of a single coin C and a set of miners $\Pi = \{p_1, \ldots, p_n\}$, which mine for C by performing hash operations. Each miner p_i possesses a hashing power that enables it to perform up to m_i hashes per time unit, and we allow miners to choose when to hash. We denote by $M \triangleq \sum_{i=1}^{n} m_i$ the total hash power all miners collectively possess. We assume that miners have fixed and variable costs. That is, a miner p_i pays a fixed price FC_i for every time unit regardless of how many hashes it performs, and an additional variable price VC_i per every hash.

Recall that a PoW-based cryptocurrency progresses in epochs, where each epoch consists of a fixed number B of blocks, and the system sets the difficulty in order to control the expected number of time units T that the epoch will last. This is done by choosing the difficulty of the puzzles in a way that requires $T \cdot M$ total hash operations in *expectation* in order to solve B puzzles. In our model we straighten the coin by allowing to define the required number of hash operations in every epoch deterministically. An execution in our model is a sequence of epochs $ep_1, ep_2, \ldots$, where each epoch ep_k consists of a total of H_k hashes performed by all miners. That is, epoch ep_k, $k > 1$, starts immediately after H_{k-1} hashes were collectively performed during epoch ep_{k-1}. We denote by t_k the number of time units it took for epoch ep_k to complete. Initially, $H_1 = M\tau$, where τ is a system parameter. Intuitively, τ is the desired duration of time that an epoch should take. Note that if all miners mine during the first epoch, then the epoch duration is exactly τ time units, i.e., $t_1 = \tau$. As for the next epochs, for every $k > 1$, we set $H_k = \frac{H_{k-1}}{t_{k-1}}\tau$. Similar to a real system, $\frac{H_{k-1}}{t_{k-1}}$ estimates the total mining power used during epoch $k-1$, and H_k is calculated so that if the mining power stays the same during epoch k, its duration will be the desired τ.

In a real system, a miner can estimate, by the difficulty, her expected revenue and profit from every hash she performs. Here we define it deterministically. Recall that in a real system a miner gets to mint a fixed number of coins for every solution, and thus the system "pays" a fixed total of rewards in every epoch. Here, since we deterministically define the number of hash operations in every epoch, we can deterministically define the revenue a miner gets for every performed hash. Let w be the total reward the coin C pays during an epoch. (Again, for example, Bitcoin [15] pays 12.5 Bitcoins per solution, so for an epoch of 2016 blocks, Bitcoin [15] pays 25200 Bitcoins.) For every $k \geq 1$, the revenue per hash in epoch ep_k is $RpH_k \triangleq \frac{w}{H_k}$ for every miner.

The miners in our model are rational in that they try to maximize their profit over time. Therefore, they may choose not to utilize there full mining capabilities at all times. However, for simplicity, we assume that miners do not change their mining effort in the course of an epoch. We denote by $\hat{m}_i[k] \leq m_i$ the number of hash operations per time unit that miner p_i performs during epoch ep_k, and the cost per-time-unit of miner p_i in epoch ep_k by $C_i[k] \triangleq FC_i + VC_i\hat{m}_i[k]$. The revenue per time unit of miner p_i during epoch ep_k is denoted by $R_i[k] \triangleq \frac{w}{H_i}\hat{m}_i[k]$, and the profit per-time-unit by $P_i[k] \triangleq R_i[k] - C_i[k]$. The utility function of a miner p_i is defined as the average profit per unit of time over an unbounded execution:

$$u_i \triangleq \lim_{K\to\infty} \frac{\Sigma_{k=1}^{K} P_i[k] \cdot t_k}{\Sigma_{k=1}^{K} t_k}$$

We assume that miners are economical beings that would not mine for a loss, but would join mining if it is profitable. Thus, by the classical model of supply and demand [17, 20], we assume that the profit of miners in the desired equilibrium, in which miners mine with there full capacity, is some $\epsilon \geq 0$. Figure 2 uses the "desired equilibrium" to demonstrate our definitions. To capture this in our model we set $w = \tau\Sigma_{i=1}^{n}(FC_i + VC_i m_i + \epsilon)$. For simplicity and readability we set $\epsilon = 0$.[3] Intuitively, w, the total reward the coin divides during an epoch, is set to be equal to the total cost the miners pay if they mine in full capacity during the epoch.

[3]In classical economic theories of free markets [17, 20], the revenues and costs strive for equality, resulting in a negligible ϵ. This is not fundamental for the smart mining strategy, and only slightly alters the numerical results.

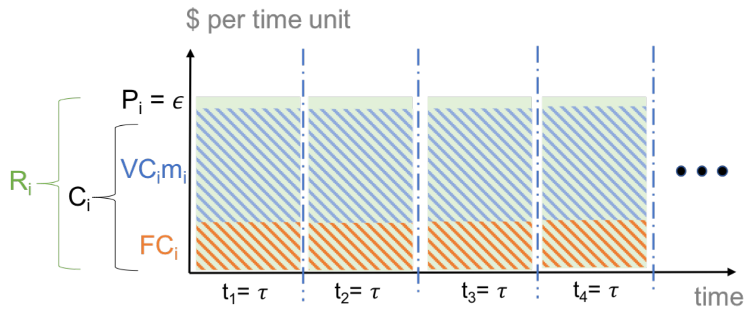


Fig. 2. Demonstration of the definitions over the "desired equilibrium" strategy. The epoch parameter k is omitted.

4 RATIONAL MINING

Recall that the "desired equilibrium" in a PoW-based cryptocurrency mining system, is such that all miners mine with full capacity in all epochs. However, due to the difficulty update mechanism this "desired equilibrium" is actually not an equilibrium in many cases, and a miner p_i has a strongly dominant strategy diverting from the protocol.

As we shall show, p_i can sometimes increase her revenues in the next epoch by mining less in the current epoch. This results in p_i losing revenues in the current epoch but also in p_i reducing her costs, therby opening a possibility for increased profits. We call the resulting mining strategy *smart mining*.

4.1 Smart Mining

Assume that all miners are mining and the system has achieved its "desired equilibrium" and remains stable during all epochs $k' < k$. Hence, according to the protocol design (and intention), for every $k' < k$ the values of $t_{k'}, H_{k'}$ and $\text{RpH}_{k'}$ are set to $\tau, M\tau$ and $\frac{w}{M\tau}$ respectively. We now show a strategy by which a miner p_i can benefit by deviating from the protocol. The strategy is for p_i to stay idle (not mine) during epochs $\{ep_k, ep_{k+2}, ep_{k+4}, \dots\}$ and to mine with full power during epochs $\{ep_{k+1}, ep_{k+3}, ep_{k+5}, \dots\}$. Let's analyze p_i's profits.

Epoch k. Since $t_{k-1} = \tau$, we have in ep_k that $H_k = M\tau$ as well. If p_i remains idle during ep_k, then

$$t_k = \frac{H_k}{M - m_i} = \frac{\tau}{1 - \frac{m_i}{M}}.$$

Consequently, p_i's profit for ep_k is $(-\text{FC}_i \cdot t_k)$, which is negative if there are any fixed costs.

Epoch k + 1. Since $t_k = \frac{\tau}{1-\frac{m_i}{M}} > \tau$, the difficulty adjustment mechanism reduces the difficulty which in turn increases the reward per hash, and correspondingly also the profit per time unit. The resulting values are:

$$H_{k+1} = \frac{\tau}{t_k} \cdot H_k = \frac{M - m_i}{M} \cdot M\tau$$

$$\text{RpH}_{k+1} = \frac{w}{H_{k+1}} = \frac{w}{(M - m_i)\tau}$$

$$t_{k+1} = \frac{H_{k+1}}{M} = \frac{M - m_i}{M} \cdot \tau$$

$$P_i[k+1] = m_i \text{RpH}_{k+1} - (\text{VC}_i m_i + \text{FC}_i) = \frac{m_i w}{(M - m_i)\tau} - (\text{VC}_i m_i + \text{FC}_i)$$

Epochs $\geq k+2$. Since $t_{k+1} = \frac{M-m_i}{M} \cdot \tau$ and $H_{k+1} = \frac{M-m_i}{M} \cdot M\tau$, we have in ep_{k+2} that $H_{k+2} = M\tau$ as in ep_k. The rest is identical to ep_k. Inductively, from here on epochs $\{ep_{k+2}, ep_{k+4}, \ldots\}$ result in the same values as in ep_k, and epochs $\{ep_{k+3}, ep_{k+5}, \ldots\}$ result in the same values as in ep_{k+1}. Figure 3 illustrates the smart mining strategy.

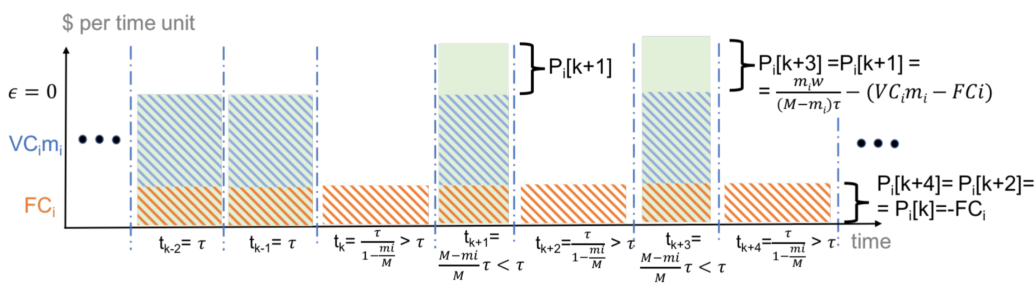


Fig. 3. Smart mining strategy for a party p_i. Profitable when $P_i[k+1] \cdot t_{k+1} > P_i[k] \cdot t_k$.

We are now ready to calculate i's profit averaged over time as defined by the utility function.

$$
\begin{aligned}
u_i &\triangleq \lim_{K \to \infty} \frac{\Sigma_{k'=1}^{K} P_i[k'] \cdot t_{k'}}{\Sigma_{k'=1}^{K} t_{k'}} \\
&= \lim_{K \to \infty} \left[\frac{\Sigma_{k'=1}^{K} (P_i[2k'-1] \cdot t_{2k'-1}) + \Sigma_{k'=1}^{K} (P_i[2k'] \cdot t_{2k'})}{\Sigma_{k'=1}^{K} t_{k'}} \right] \\
&= \frac{P_i[k+1] \cdot t_{k+1} + P_i[k] \cdot t_k}{t_{k+1} + t_k} \\
&= \frac{\left(\frac{m_i w}{(M-m_i)\tau} - (\mathrm{VC}_i m_i + \mathrm{FC}_i) \right) t_{k+1} - \mathrm{FC}_i t_k}{t_{k+1} + t_k} \\
&= \frac{\left(\frac{m_i w}{(M-m_i)\tau} - (\mathrm{VC}_i m_i + \mathrm{FC}_i) \right) \frac{M-m_i}{M} - \frac{M}{M-m_i} \mathrm{FC}_i}{\frac{M-m_i}{M} + \frac{M}{M-m_i}} \\
&= \frac{\frac{m_i w}{M\tau} - (\mathrm{VC}_i m_i + \mathrm{FC}_i) \frac{M-m_i}{M} - \frac{M}{M-m_i} \mathrm{FC}_i}{\frac{M-m_i}{M} + \frac{M}{M-m_i}}
\end{aligned}
$$

In the "desired equilibrium" of the protocol $u_i = \epsilon$. Therefore, the smart mining strategy strictly dominates the protocol whenever $u_i > \epsilon$. Moreover, recall that (1) the revenue per hash given by the coin in the stable state is $\frac{w}{M\tau}$, (2) miner p_i's revenue per unit time is $\frac{w}{M\tau} m_i$, and (3) her costs are $\mathrm{VC}_i m_i + \mathrm{FC}_i$. Since honest miners don't mine for a loss, under our demand and supply assumption [17, 20], the market powers establish that $\frac{w}{M\tau} m_i = (\mathrm{VC}_i m_i + \mathrm{FC}_i) + \epsilon$. Assuming for simplicity $\epsilon \leftarrow 0$, we calculate below when $u_i > 0$.

$$
\begin{aligned}
& u_i > 0 \\
& \iff \\
& \frac{w}{M\tau} m_i > \frac{M - m_i}{M} (\mathrm{VC}_i m_i + \mathrm{FC}_i) + \frac{M}{M - m_i} \mathrm{FC}_i \\
& \iff \\
& \mathrm{VC}_i m_i + \mathrm{FC}_i > \frac{M - m_i}{M} (\mathrm{VC}_i m_i + \mathrm{FC}_i) + \frac{M}{M - m_i} \mathrm{FC}_i \\
& \iff \\
& \frac{m_i}{M} > \frac{\mathrm{FC}_i}{\mathrm{VC}_i m_i + \mathrm{FC}_i} \cdot \frac{M}{M - m_i} \\
& \iff \\
& \frac{m_i}{M} \cdot \frac{M - m_i}{M} > \frac{\mathrm{FC}_i}{\mathrm{VC}_i m_i + \mathrm{FC}_i}
\end{aligned}
$$

Denoting the percentage of the fixed cost out of the total costs as $y \triangleq \frac{\mathrm{FC}_i}{\mathrm{VC}_i \cdot m_i + \mathrm{FC}_i}$, and the percentage of i's mining power as $x \triangleq \frac{m_i}{M}$, we get that our smart mining attack strictly dominates the protocol whenever $x \cdot (1 - x) > y$ for $(x, y) \in (0, 1) \times (0, 1)$. Figure 4 illustrates in which cost structures the smart mining attack dominates honest mining. As an example, when the fixed costs are 10% of the total costs, having 12% of the mining power suffices to create excess profit using smart mining.

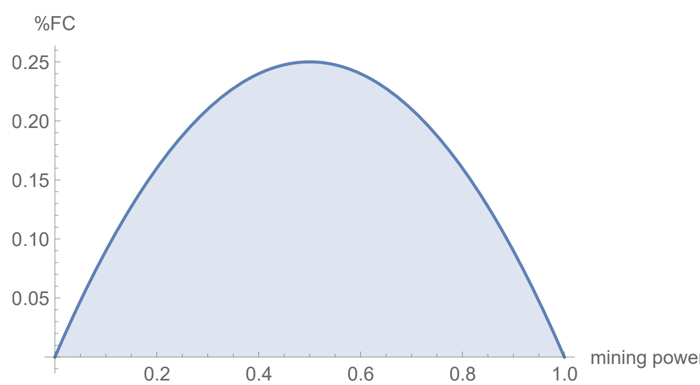


Fig. 4. Smart mining strictly dominates honest mining in the area under the curve. x-axis: $\frac{m_i}{M}$, y-axis: $\frac{\mathrm{FC}_i}{\mathrm{VC}_i \cdot m_i + \mathrm{FC}_i}$.

4.2 Smarter Mining

Having explained the basic smart mining strategy we can now improve it by fine tuning. When we expand the miner's strategy space to include strategies where $0 \leq \hat{m}_i[k] \leq m_i$ (that is, not an all or nothing but a hybrid strategy), the miner can achieve even greater profits by deviating from honest mining. Moreover, a miner profits from the attack in many more scenarios, both with higher fixed costs percentages and with less mining power required. For these reasons we call the fine-tuned attack *smarter mining*. Smarter mining stems from the observation that all mining operations are profitable in the high revenue epochs regardless of whether the miner participated in the attack, and only the idle miners bare a loss in the low revenue epochs. In particular, an honest miner profits from the attack without incurring any costs. Thus, a smarter attacking miner might choose to optimize her profits by slightly reducing her excess profits in an HRE in exchange for a higher reduction of her LRE losses. A miner i that optimizes her $\hat{m}_i[k]$ in a smarter way than by choosing $\hat{m}_i[k] \in \{0, m_i\}$ may therefore enjoy higher average profits. An illustration of the intuition behind smarter mining appears in Figure 5.

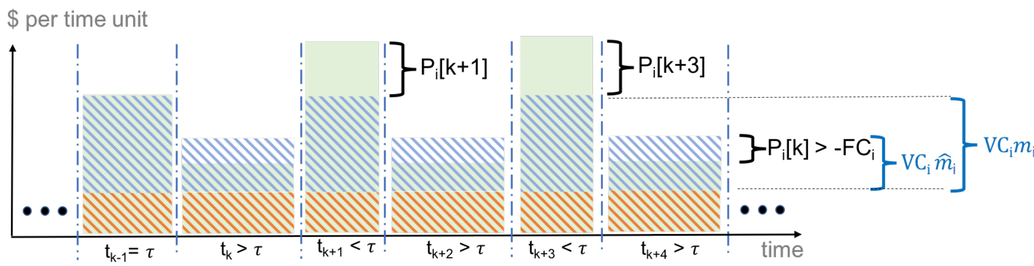


Fig. 5. The smarter mining strategy for a party p_i. By mining with part of her mining power ($0 < \hat{m}_i < m_i$) during low revenue epochs, p_i is able to maximize her profits by optimizing the balance between her losses in low revenue epochs and her gains in high revenue epochs.

Expected Profit Analysis. The analysis is similar to Section 4.1, except that in epochs $\{ep_k, ep_{k+2}, \ldots\}$, miner p_i mines with $\hat{m}_i[k]$ mining power and in epochs $\{ep_{k+1}, ep_{k+3}, \ldots\}$, she mines with her full power m_i. Denote the amount of p_i's idle mining power during epoch ep_k by $\Delta m_i \triangleq m_i - \hat{m}_i[k]$. We obtain the following:

Epoch k.

$$\begin{aligned}
t_{k-1} &= \tau \\
H_k &= M\tau \\
t_k &= \frac{H_k}{M - \Delta m_i} = \frac{\tau}{1 - \frac{\Delta m_i}{M}} \\
\text{RpH}_k &= \frac{w}{H_k} = \frac{w}{M \cdot \tau} = \frac{1}{m_i}(\text{VC}_i m_i + \text{FC}_i) \\
P_i[k] &= \text{RpH}_k \hat{m}_i[k] - (\text{VC}_i \hat{m}_i[k] + \text{FC}_i) = \left(\frac{\hat{m}_i[k]}{m_i} - 1\right)\text{FC}_i
\end{aligned}$$

Epoch k + 1. Since $t_k = \frac{\tau}{1 - \frac{\Delta m_i}{M}}$, the difficulty adjustment mechanism reduces the difficulty, which in turn increases the reward per hash and correspondingly the profit per time unit as well. The resulting values are:

$$\begin{aligned}
H_{k+1} &= \frac{\tau}{t_k} H_k = (M - \Delta m_i)\tau \\
t_{k+1} &= \frac{H_{k+1}}{M} = \frac{M - \Delta m_i}{M}\tau \\
\text{RpH}_{k+1} &= \frac{w}{H_{k+1}} = \frac{w}{(M - \Delta m_i)\tau} = \frac{M}{M - \Delta m_i}\text{RpH}_k = \frac{1}{m_i} \cdot \frac{M}{M - \Delta m_i}(\text{VC}_i m_i + \text{FC}_i) \\
P_i[k+1] &= m_i \text{RpH}_{k+1} - (\text{VC}_i m_i + \text{FC}_i) = \frac{\Delta m_i}{M - \Delta m_i}(\text{VC}_i m_i + \text{FC}_i)
\end{aligned}$$

Epochs $\geq k+2$. Since $t_{k+1} = \frac{M-\Delta m_i}{M}\tau$ and $H_{k+1} = (M-\Delta m_i)\tau$, we have in ep_{k+2} that $H_{k+2} = M\tau$ as in ep_k. The rest is identical to ep_k. Inductively, from here on epochs $\{ep_{k+2}, ep_{k+4}, \ldots\}$ result in the same values as in ep_k, and epochs $\{ep_{k+3}, ep_{k+5}, \ldots\}$ result in the same values as in ep_{k+1}.

. As a result, if p_i employs the smarter mining strategy, her profit averaged over time as defined by the utility function would be:

$$u_i \triangleq \lim_{K\to\infty} \frac{\Sigma_{k'=1}^{K} P_i[k'] \cdot t_{k'}}{\Sigma_{k'=1}^{K} t_{k'}}$$

$$= \lim_{K\to\infty} \left[\frac{\Sigma_{k'=1}^{K}(P_i[2k'-1] \cdot t_{2k'-1}) + \Sigma_{k'=1}^{K}(P_i[2k'] \cdot t_{2k'})}{\Sigma_{k'=1}^{K} t_{k'}} \right]$$

$$= \frac{P_i[k+1] \cdot t_{k+1} + P_i[k] \cdot t_k}{t_{k+1} + t_k}$$

$$= \frac{\overbrace{\frac{\Delta m_i}{M-\Delta m_i}(\mathrm{VC}_i m_i + \mathrm{FC}_i)}^{P_i[k+1]} \cdot \overbrace{\frac{M-\Delta m_i}{M}\tau}^{t_{k+1}} + \overbrace{\left(\frac{\hat{m}_i[k]}{m_i} - 1\right)\mathrm{FC}_i}^{P_i[k]} \cdot \overbrace{\frac{\tau}{1-\frac{\Delta m_i}{M}}}^{t_k}}{\frac{M-\Delta m_i}{M} \cdot \tau + \frac{\tau}{1-\frac{\Delta m_i}{M}}}$$

$$= \frac{\frac{\Delta m_i}{M}(\mathrm{VC}_i m_i + \mathrm{FC}_i) - \frac{\Delta m_i}{m_i} \cdot \frac{M}{M-\Delta m_i}\mathrm{FC}_i}{\frac{M-\Delta m_i}{M} + \frac{M}{M-\Delta m_i}}$$

Figure 6 shows the strong potential of the proposed strategy. The colors represent profits as a percentage of the total cost ($\frac{u_i}{\mathrm{VC}_i m_i + \mathrm{FC}_i} - 1$), the x axis represents the mining power ($\frac{m_i}{M}$), and the y axis the costs structure ($\frac{\mathrm{FC}_i}{\mathrm{VC}_i m_i + \mathrm{FC}_i}$). Unlike in selfish mining [10, 18], we can see that there are many scenarios in which smarter mining is profitable even for a very small miner. Furthermore, the expected profits yield a reasonable return on investment that makes smarter mining a viable economic strategy.

5 COIN SECURITY AND OTHER PARTIES BEST RESPONSE

As we have shown in Section 4, the desired equilibrium of the coin in which the participants invest all of their power to secure the coin (mining with full capacity), is actually not an equilibrium in many cases. In an abundance of realistic scenarios, a miner p_i has a strongly dominating strategy over honest mining. One might hope, however, that the response of the other participants in the system may somehow balance the negative effect of p_i deviating from the protocol. Unfortunately, this does not happen with smart/smarter mining. Unlike other strategies (e.g. selfish mining [10, 18]), in our strategy p_i's deviation from the protocol does not harm the rest of the miners; On the contrary, it benefits them.

Prior to the attack, a miner $p_j \neq p_i$ that acts honestly receives the income $m_j RpH_k$ per unit time. After p_i starts the attack, p_j receives the income per unit time $m_j RpH_k$ (which she is satisfied with) in the low revenue epochs, and she receives $m_j RpH_{k+1} > m_j RpH_k$ in the high revenue epochs. Thereby, p_j's utility (u_j) is increased due to p_i's attack. Thus, even if p_j is rational and considers deviating from honest mining, she has no incentive to obstruct p_i's attack. In fact, an analysis similar to the one above shows that p_j will not resist p_i's attack in any way. Indeed, the only possible strategy for p_j deviating from honest mining that might be more profitable in specific cases is to join the attack and increase both of their profits.

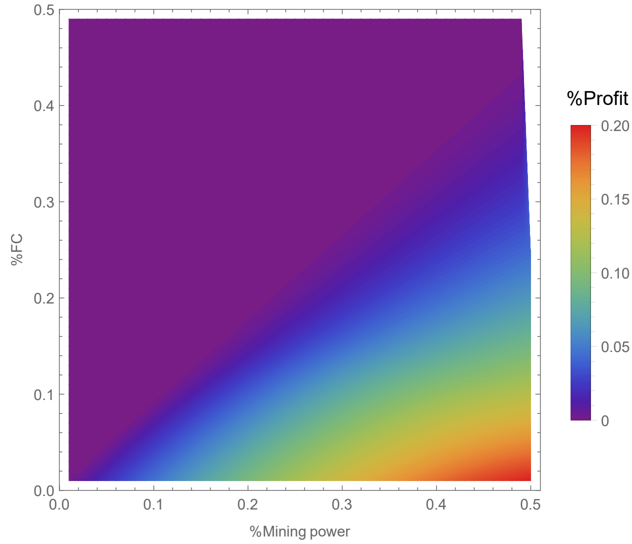


Fig. 6. The Return On Investment in smarter mining. For every ϵ, if **%Profit** $> \frac{\epsilon}{VC_i m_i + FC_i}$, than smarter mining strictly dominates honest mining.

Coin security. Obviously, since less mining power is invested during low revenue epochs in smart/smarter mining, these strategies expose the coin to under 50% attacks. For example, the smart mining strategy is profitable for a miner that controls 20% of the total mining power with fixed costs that constitutes up to 15% of her total costs. Therefore, if such a miner chooses to adopt the smart mining strategy, a malicious miner will be able to attack the coin with only 41% of the total mining power. This vulnerability increases even more when we consider the other miners. As mentioned above, in some cases it is profitable for other miners to join the smart/smarter mining strategies, and leave the low revenue epochs with even less honest mining power.

Surprisingly, the joining of new miners to the coin only makes the security problem worse in smart/smarter mining strategies. For comparison, in long term analysis of selfish mining [10], we get that new miners benefit from joining the coin due to the drop in difficulty, which leads to higher RpH. These new miners mitigate the loss in honest mining power due to forks created by selfish mining. However, this is not the case in smart/smarter mining. Obviously joining miners will join the epochs that are profitable for them, which are the high revenue epochs (smart/smarter mining does not change the RpH in low level epochs, so if these epochs were not profitable for new miners before, they remain unprofitable after the smart/smarter mining attack is performed). This, in turn, will lead to a drop in the RpH in the low revenue epochs due the difficulty adjustment, and force more miners to abandon these epochs, further reducing the honest mining power.

6 DISCUSSION

The recent drop in the Bitcoin [15] price makes mining much less profitable than in the past, and forces miners to revisit their mining strategies. In this paper we propose two strategies, smart and smarter mining, that strictly dominate honest mining in many cases. In contrast to other known attacks that induce zero-sum games among miners [10], the strategies proposed in this paper increase miners' profits by reducing their variable costs (i.e., electricity). However, while saving electricity power is very encouraging for the environment, it negatively affects the coin's security. The smart/smarter mining strategies expose the coin to under 50% attacks during low revenue epochs, and this vulnerability only escalates when new miners join the coin as a response to the increase in profit margins induced by these strategies.

Interestingly, the smart/smarter mining can be profitable even for miners that possess a relatively small amount of mining power, and in case the fixed cost is negligible even less than 1% of the mining power can be enough. Another point that is worth mentioning is the potential of smart/smarter mining in a system with many PoW-based coins. Although this paper deals with a single coin system, the extension into multiple coins only strengthens the attack strategies, since it allows attackers to mine for other coins during low revenue epochs and thus significantly reduce the loss in these epochs or even make profit by joining other coins in their high revenue epochs.[4]

In this work we have exploited the difficulty adjustment mechanisms of PoW-based cryptocurrencies by manipulating it to our benefit. In general, this is but a single case of trying to emulate a continuous process (supply and demand adjustment) by a discrete process (difficulty update in epochs). Similar issues are likely to arise in other mechanisms and systems, as the transformation from continuous to discrete is not trivial, but does not always receive the appropriate attention during the design of a system. We believe that areas such as blockchains, where real world systems progress faster than rigorous analysis, provide many opportunities for theoretical research to make meaningful contributions.

7 ACKNOWLEDGMENTS

We would like to thank Yoram Moses and Gal Assa for revising earlier versions of this paper, and Itay Tsabary for his helpful feedback.

REFERENCES

[1] Bitcoin power compare. https://powercompare.co.uk/bitcoin/, Accessed: 2019-01-28.
[2] Cryptocurrency market state visualization. https://coin360.io/, Accessed: 2019-01-28.
[3] Bitcoin power compare. https://www.theblockcrypto.com/2018/12/04/the-bitcoin-mining-death-spiral-debate-explained/, Accessed: 2019-01-30.
[4] Ethereum foundation. https://www.ethereum.org/, Accessed: 2019-02-07.
[5] Adam Back et al. Hashcash-a denial of service counter-measure, 2002.
[6] Joseph Bonneau. Why buy when you can rent? In *International Conference on Financial Cryptography and Data Security*, pages 19–26. Springer, 2016.
[7] Miles Carlsten, Harry Kalodner, S Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 154–167. ACM, 2016.
[8] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE, 2013.
[9] Ittay Eyal. The miner's dilemma. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 89–103. IEEE, 2015.
[10] Ittay Eyal and Emin Gun Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.
[11] Tim Hellemans, Benny Van Houdt, Daniel S Menasche, Mandar Datar, Swapnil Dhamal, and Corinne Touati. Mining competition in a multi-cryptocurrency ecosystem at the network edge: a congestion game approach. *ACM SIGMETRICS Performance Evaluation Review*, 46(3):114–117, 2019.
[12] Benjamin Johnson, Aron Laszka, Jens Grossklags, Marie Vasek, and Tyler Moore. Game-theoretic analysis of ddos attacks against bitcoin mining pools. In *International Conference on Financial Cryptography and Data Security*, pages 72–86. Springer, 2014.
[13] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 365–382. ACM, 2016.
[14] Kevin Liao and Jonathan Katz. Incentivizing blockchain forks via whale transactions. In *International Conference on Financial Cryptography and Data Security*, pages 264–279. Springer, 2017.
[15] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[4]This trivial property makes the attack somewhat unbeaten in the practical world and thus poses a significant threat to PoW-based cryptocurrencies.

[16] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 305–320. IEEE, 2016.
[17] David Ricardo. On the principles of political economy and taxation. 1817.
[18] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2016.
[19] Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden. Incentive compatibility of bitcoin mining pool reward functions. In *International Conference on Financial Cryptography and Data Security*, pages 477–498. Springer, 2016.
[20] Adam Smith. An inquiry into the nature and causes of the wealth of nations, 1776.
[21] Alexander Spiegelman, Idit Keidar, and Moshe Tennenholtz. Game of coins. *arXiv preprint arXiv:1805.08979*, 2018.
[22] Itay Tsabary and Ittay Eyal. The gap game. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 713–728, New York, NY, USA, 2018. ACM.