# Incentivizing Double-Spend Collusion in Bitcoin

Kevin Liao[1] and Jonathan Katz[2]

[1] Arizona State University
kevinliao@asu.edu
[2] University of Maryland
jkatz@cs.umd.edu

**Abstract.** Bitcoin's core innovation is its solution to double-spending, called Nakamoto consensus. This provides a probabilistic guarantee that transactions will not be reversed or redirected, provided that it is improbable for an attacker to obtain a majority of mining power in the network. While this may be true in the traditional sense, this assumption becomes tenuous when miners are assumed to be rational and hence venal. Accordingly, we present the *whale attack*, in which a minority attacker increases her chances of double-spending by incentivizing miners to subvert the consensus protocol and to collude via *whale transactions*, or transactions carrying anomalously large fees. We analyze the expected cost to carry out the attack, and simulate the attack under realistic system parameters. Our results show that double-spend attacks, conventionally thought to be impractical for minority attackers, can actually be financially feasible and worthwhile under the whale attack. Perhaps more importantly, this work demonstrates that rationality should not underestimated when evaluating Bitcoin's security.

## 1 Introduction

Decentralized cryptocurrencies have precipitated considerable interest in recent years. Bitcoin [1], the first empirical success of its kind, has laid the foundation for subsequent decentralized cryptocurrencies through its innovative solution to double-spending, a long-standing failure mode of digital currencies that allows an attacker to spend a given set of coins more than once. This solution, known as Nakamoto consensus, provides a high assurance that coins will not be double spent, barring if an attacker obtains an improbable amount of resources. However, this tenuous assumption has induced notions of a looming crisis in the Bitcoin community, which casts serious doubt on the security of cryptocurrencies as currently prescribed.

In general, the security of a digital currency is congruous with the irreversibility of its transactions. More concretely, when users send coins to vendors in exchange for merchandise, vendors expect that once the purchased merchandise has been disbursed, the transaction will not be reversed or redirected elsewhere. Double-spending undermines this desideratum, in that if an attacker issues two conflicting transactions using the same set of coins, say, one to the vendor and one to herself, eventually one of these transactions will be invalidated. If the

vendor unknowingly disburses the merchandise, under the impression that it has been paid for, and the paying transaction is invalidated, then the vendor is left empty-handed.

In this regard, Nakamoto consensus offers a probabilistic guarantee that a transaction will not be reversed. The protocol is as follows. Participants in the Bitcoin network, known as miners, compete to solve a computationally expensive proof-of-work puzzle. The miner who solves this puzzle is permitted to add a block of newly confirmed transactions to the blockchain, a distributed public ledger serializing all transactions ever issued. In remuneration, the miner is rewarded with newly minted bitcoins and (more importantly for this work), any embedded transaction fees, which are gratuities left by payers. The new block and its proof-of-work are then broadcast to the network, and upon verification, miners will add the block to their corresponding blockchains and repeat the mining process atop their updated ledgers. Since mining is performed concurrently, it may be the case that conflicting versions of the blockchain form, known as branches. In the prescribed protocol, miners resolve this by mining on the longest branch, as measured by the total expense of mining power. During this process, the shorter branch will be orphaned and any conflicting transactions will be invalidated.

Although transactions invalidated during branch selection enables the possibility of double-spending, as transactions gain more confirmations, in other words, when new blocks are added atop their respective blocks, the probability that a conflicting longer branch forms decreases exponentially. Thus, a transaction with six confirmations is well-accepted by the community to be secure against double-spending.

The main caveat of this probabilistic guarantee is that it assumes no single mining entity wields a majority of mining power in the network. Otherwise the system ceases to be decentralized—a majority miner can unilaterally control the blockchain and can thus double-spend at will. Bitcoin's security guarantees have been proven [2] only in accordance with this assumption, namely that a majority of miners (as measured by their mining powers) behave honestly by adhering to the prescribed protocol. The question then arises of whether or not these security guarantees hold when miners are, instead, assumed to behave rationally, in other words, they are incentivized by maximizing profit.

**Our contributions.** We consider a minority attacker, henceforth referred to as Alice, who attempts to double-spend against a vendor, henceforth referred to as Bob, within a purely rational network. From a cost perspective, double-spend attacks require a large proportion of mining power that may be improbable to attain singly. Auspiciously, Alice can amplify her own mining power by incentivizing, or "bribing," rational miners into subverting the prescribed protocol.

Accordingly, we present a novel double-spending strategy called the *whale attack*, which was inspired by a peculiar (perhaps erroneous) bitcoin transaction,[1] in which the payer issued a transaction carrying an exorbitant transaction fee of

[1] `https://blockchain.info/tx/143707654`

291 bitcoins.[2] We henceforth generalize transactions carrying anomalously large transaction fees as *whale transactions*, and we study the implications of these transactions on mining consensus. In particular, we are interested in the capabilities of whale transactions to incentivize rational and transaction-fee sensitive miners into colluding towards a double-spend attack.

The main contributions of this work are the following:

1. We introduce and formalize the whale attack, which demonstrates that rationality should not be underestimated when evaluating Bitcoin's security.
2. We establish informal upper bounds on the expected cost to carry out the whale attack with success probability 1.
3. We simulate the whale attack, mirroring the actual Bitcoin network, as a proof of concept for the feasibility of the attack, even when the attacker wields a modest amount of mining power and capital.

### 1.1 The Whale Attack

We begin by informally introducing the whale attack. Suppose a minority attacker Alice attempts to launch a double-spend attack against a vendor Bob. Alice initiates the attack by mining a block, but instead of broadcasting this block to the network, she surreptitiously mines atop this block by herself, thereby forming her private branch. She then uses the same set of bitcoins to pay Bob on the original branch, while issuing a conflicting transaction to herself on her private branch. Upon receiving Alice's transaction, Bob will wait six confirmations, as per conventional wisdom, before sending the purchased merchandise.

For the attack to succeed, Alice's private branch must keep up with and overtake the original branch after at least six confirmations have been reached. Consider that in a traditional double-spending attack, that is, without bribing other miners into colluding, the probability that Alice succeeds is quite low since she wields a minority of mining power in the network. Alternatively, suppose Alice proceeds as before and she mines on her private branch while waiting for six confirmations. Then, once six confirmations have been reached, if Alice's private branch is shorter than the original branch, she publishes her branch to the network and issues whale transactions, which are redeemable only by mining on her branch.

Assuming miners in the network are rational, they will choose to *whale mine* on Alice's branch if doing so is more profitable than honest mining. Whether whale mining is more profitable depends on the whale transactions' fees given the risk of mining on a shorter branch and the forgone block rewards should the attack fail. As more miners switch to whale mining, the probability that the double-spend succeeds increases. If a majority of mining power comes to whale mine, then the attack is guaranteed to succeed. Finally, once Alice's branch

[2] The current recommended transaction fee for a no-delay transaction is $6.0 \times 10^{-7}$ bitcoins per byte. At the current median transaction size of 257 bytes, this would only amount to a transaction fee of $1.542 \times 10^{-4}$ bitcoins (from `https://bitcoinfees.21.co`, accessed September 1, 2016).

overtakes the original branch in length, Alice's transaction to Bob is invalidated, and Bob is left empty-handed.

### 1.2 Related Work

There is a growing body of research examining incentive compatibility in Bitcoin. A number of recent works study the implications of block withholding, that is, delaying the broadcast of newly mined blocks. Rosenfeld [3] and Eyal [4] analyze block withholding attacks, in which an infiltrating miner discards full proofs-of-work, thereby sabotaging the victim pool's expected rewards. Eyal and Sirer [5] develop the selfish mining attack, in which an attacker surreptitiously forks the blockchain and withholds blocks in attempt to later orphan the original branch, thereby wasting computations by honest miners. Nayak *et al.* [6] and Sapirshtein *et al.* [7] further analyze and optimize the space of selfish mining strategies.

More closely related to this work are bribery attacks. Bonneau [8] presents various bribery attacks, in which an attacker temporarily rents mining power rather than traditionally buying mining hardware. For example, an attacker pays miners out-of-band, in other words, outside of Bitcoin, to mine on a chosen branch. Alternatively, an attacker sends bribery money to a set of scripted addresses, located in-band on the attacker's branch, that can be claimed by mining the next block(s). Our attack differs from the former, in that whale transactions are trustless and can be issued anonymously, and compares to the latter, but instead disburses bribery money through transaction fees, which are inherent to the protocol.

Teutsch *et al.* [9] present another bribery attack, in which an attacker casts proof-of-work puzzles as Ethereum smart contracts, called script puzzles, to serve as an additional mining revenue source. Thus, rational miners may increase their profits by apportioning their mining powers between puzzle-solving and Bitcoin mining, thereby reducing mining power in the Bitcoin network. Our attack differs from the script puzzle attack, in that miners are purveyed a single source of revenue, namely the block rewards on the longest branch.

More broadly, there is also a growing interest in the interrelationship between transaction fees and Bitcoin's long-term health. Möser and Böhme [10] perform a longitudinal analysis of transaction fees and examine the externalities that influence these fees. Kroll *et al.* [11], Houy [12], and Kaşkaloğlu [13] consider the economics of Bitcoin mining and discuss potential changes to transaction fees and their policies in the long-term. Carlsten *et al.* [14] develop a new attack strategy and revisit the selfish mining attack in the context of a transaction-fee regime.

## 2 Model

We adapt the model used by Rosenfeld [15] and updated by Sompolinsky and Zohar [16] to consider double-spending under the whale attack.

We assume that the distribution of mining power in the network remains constant. An attacker Alice controls a fraction $\alpha$ of the mining power, where $\alpha \leq 0.5$, since otherwise she could double-spend by herself at will. The remaining network consists of $k$ mining entities controlling a fraction $\beta = 1-\alpha$ of the mining power. Thus, each mining entity $i$ controls a fraction $\beta_i$ such that $\sum_{i=1}^{k} \beta_i = \beta$.

Miners mine blocks according to a Poisson process with rate $\lambda$, which also remains constant. Further, the propagation of new blocks to the network is instantaneous. Thus, the passage of time is a discrete-time process marked by block creation events on either of the original branch or Alice's branch. The reward for mining a block on the original branch is 1; the reward for mining a block on Alice's branch is $\delta + 1$.

Following each block creation event, each mining entity, including Alice, makes a new rational decision that will be pursued until the next block creation event. More specifically, Alice makes a rational decision for whether to continue the attack or reset the attack. Similarly, once whale transactions are underway underway, each mining entity $i$ makes a binary rational decision for $\gamma_i \in \{0, 1\}$ of whether to honest mine ($\gamma_i = 0$) or whale mine ($\gamma_i = 1$).

At this point, the remaining network $\beta$ can be further divided into two partitions: whale miners and honest miners. More formally, a fraction $q = \alpha + \sum_{i=1}^{k} \gamma_i \cdot \beta_i$ of the mining power is devoted to whale mining, in other words, extending Alice's branch. On the other hand, a fraction $p = \sum_{i=1}^{k} (1-\gamma_i) \cdot \beta_i = 1-q$ of the mining power is devoted to honest mining, in other words, extending the original branch.

### 2.1 Attack Strategy

The whale attack is carried out in two phases: the *pre-mining* phase and the *race* phase. An algorithm for the attack is fully specified in Algorithm 1.

**Pre-mining phase.** In this phase, Alice surreptitiously forks the blockchain, issues a pair of conflicting double-spend transactions ($tx_B$ to Bob and $tx_A$ to herself), and then singly mines on her private branch until $tx_B$ has reached $n$ confirmations, at which point Bob will disburse the merchandise. Note that Alice with neither reveal her private branch nor issue whale transactions before Bob disburses the merchandise, since either action could dissuade Bob from doing so.

To initiate the attack, while Alice need only mine one block to begin her private branch, Sompolinsky and Zohar present the *pre-mining* [16] strategy, by which Alice could, in theory, mine $n + 1$ blocks prior to double-spending. Thus, the attack succeeds with probability 1. Since this may take a long time to achieve (depending on $n$), the underlying premise is that Alice can freely choose when to purchase merchandise from Bob. We aver that, in practice, this is a plausible assumption. Nonetheless, Alice can alternatively pre-mine fewer blocks to carry out the attack with a lower success probability. Regardless, Sompolinsky and Zohar also point out that Alice can employ selfish mining strategies [5] to gain while pre-mining.

**Algorithm 1** Whale Attack

```
1: procedure RESET
2:     original_branch ← longest branch
3:     Alice_branch ← longest branch
4:     l_count ← 0                          ▷ len(Alice_branch) − len(original_branch).
5:     Issue tx_A on Alice_branch.
6:     Mine at head of Alice_branch.

7: procedure PRE-MINE(l, n)
8:     RESET
9:     while l_count < l do
10:        new_block ← LISTEN                ▷ LISTEN for block creation event.
11:        if new_block on Alice_branch then
12:            l_count ← l_count + 1
13:        else if l_count = 0 then          ▷ len(Alice_branch) < len(original_branch).
14:            RESET
15:        else                              ▷ len(Alice_branch) ≥ len(original_branch).
16:            l_count ← l_count − 1
17:    Issue tx_B on original_branch.
18:    n_count ← 0
19:    m ← 0
20:    while n_count < n do
21:        new_block ← LISTEN
22:        if new_block on Alice_branch then
23:            m ← m + 1
24:        else
25:            n_count ← n_count + 1
26:    Publish Alice_branch.
27:    if m + l ≤ n then                     ▷ len(Alice_branch) ≤ len(original_branch).
28:        RACE(n − (m + l))

29: procedure RACE(z)
30:    Issue new tx_W on Alice_branch.
31:    while z > −1 do
32:        new_block ← LISTEN
33:        if new_block on Alice_branch then
34:            z ← z − 1
35:            Issue tx_W j on Alice_branch.
36:        else if z = z_lim − 1 then                    ▷ Cut off attack.
37:            RESET
38:        else
39:            z ← z + 1
```

Suppose Alice aims to pre-mine $l \in \mathbb{N} : 1 \leq l \leq n+1$ blocks more than the original branch before issuing $tx_B$. Alice embeds $tx_A$ in the first block she mines ahead of the original branch, which marks the start of a new "attempt." In any attempt, if the original branch overtakes Alice's branch in length, she accepts the original branch and resets to a new attempt. Otherwise, if Alice successfully pre-mines $l$ blocks more than the original branch, she issues $tx_B$ on the original branch.

Then, overloading Sompolinsky's and Zohar's definition of pre-mining, Alice also singly mines $m$ blocks on her private branch while waiting for $tx_B$ to reach $n$ confirmations. In accordance with Rosenfeld's analysis [15], the probability for a given value of $m$ is

$$P(m) = \binom{m+n-1}{m} \alpha^m \beta^n. \tag{1}$$

Finally, once Bob disburses the merchandise, Alice publishes her heretofore private branch containing $m+l$ pre-mined blocks. If $m+l \leq n$, in other words, Alice's branch is shorter than the original branch, then the attack transitions to the race phase.

**Race phase.** In this phase, Alice's branch and the original branch enter into a race. However, instead of continuing to singly mine on her branch, Alice issues whale transactions ($tx_W$) on her branch, which offer a $\delta$ percentage increase over the normal block reward. Throughout the rest of this paper, when we refer to the value of whale transactions, we are referring to the value of its transaction fee. Although Alice can choose from several payout structures, we assume that she issues a new whale transaction in each block on her branch until the attack succeeds. This allows for a more consistent proportion of whale mining power in the network, since mining entities persistently contend for $tx_W$ fees throughout the race phase (see Section 3 for more details about our assumptions).

The race phase can be modeled as a biased random walk. The initial state is $z = n-(m+l)$, where $z$ is the lead of the original branch. In each block creation step, $z$ increases by 1 with probability $p$ and $z$ decreases by 1 with probability $q$, where $p$ and $q$ are the mining powers devoted to honest mining and whale mining, respectively. Again, in accordance with Rosenfeld's analysis [15], the probability that $z$ reaches the absorbing state $-1$, in other words, Alice's branch becomes longer than the original branch, as a function of $p$, $q$,and $z$ is

$$a_z = min(q/p, 1)^{max(z+1,0)} = \begin{cases} 1 & \text{if } z < 0 \text{ or } q > p \\ (q/p)^{z+1} & \text{if } z \geq 0 \text{ and } q \leq p. \end{cases} \tag{2}$$

As $z$ increases, the probability that the attack succeeds decreases and the attack may become intractable. For this reason, Alice can choose to cut off the attack when $z$ reaches $z_{lim}$. This is then analagous to the Gambler's Ruin problem.

In accordance with Sompolinsky's and Zohar's analysis [16], the probability that the whale attack fully succeeds in a given attempt is

$$f(n,\alpha) = \sum_{l=0}^{\infty} \frac{1-2\alpha}{\beta} \cdot \left(\frac{\alpha}{\beta}\right)^{l} \cdot \left( \sum_{m=0}^{n-l} \binom{m+n-1}{m} \alpha^{m}\beta^{n} \cdot \left(\frac{q}{p}\right)^{n+1-m-l} + \sum_{m=n-l+1}^{\infty} \binom{m+n-1}{m} \alpha^{m}\beta^{n} \right). \quad (3)$$

While it would be interesting to analyze the success probabilities of the attack further, we are more interested in the expected cost to carry out the whale attack with success probability 1. This allows us to determine if the whale attack is worthwhile, without having to make any assumptions about Alice's risk tolerance or the liquidity of the purhased merchandise should the attack fail.

## 3 Analysis

We now establish informal upper bounds on the expected cost to carry out the whale attack with success probability 1. Since Alice's profit is contingent on the value of the double-spend being greater than the sum of the whale transactions, the main questions we are trying to answer are "How large do whale transactions need to be?" and "How many whale transactions are needed?"

Before we begin, we make a number of simplifying assumptions and explain their rationales here, as well as en route of the analysis. Granted, these assumptions may differ in practice and could dramatically change (in most cases dramatically reduce) the cost of the attack. However, these simplifications are meant to make the analysis more tractable, and to determine if the attack is even practically worthwhile. We leave these as points of discussion in Section 4.

1. *Mining entities consider at least their own mining power and Alice's mining power when making rational decisions.* We make minimal assumptions about the sophistication of mining entities in evaluating their profits. We simply assume that each mining entity considers its own mining power and Alice's mining power.
2. *Mining entities are not "sticky."* When mining entities mine a whale block, they will not simply "stick" to whale mining for the remainder of the attack. Instead, they continue to make new rational decisions following each block event, without taking into consideration their prior earnings.
3. *Mining entities will choose the more profitable (even marginally) mining strategy.* We later determine appropriate values for whale transactions $\delta$, which, if even marginally sufficient, will incentivize mining entities to whale mine.
4. *Whale mining power is kept constant throughout the race phase.* Instead of keeping the values of whale transactions constant throughout the attack, Alice keeps whale mining power constant by issuing appropriate whale transactions in each block on her branch.

5. *Alice issues whale transactions in every block on her branch until the attack succeeds.* Since mining entities always make new rational decisions following each block event, Alice issues whale transactions until her branch is longer than the original branch. This also means that she never cuts off the attack ($z_{lim} = \infty$).

Assumptions 1, 2, and 5 are meant to induce an upper bound on the cost of the attack. To prevent any confusion, we call $\delta$ an upper bound, by virtue of our assumptions, but it is defined as the minimum whale transaction necessary for whale mining to be profitable. Thus, Assumption 3 establishes that whale transactions need only be marginally more than $\delta$ for a given block event, or the sum of $\delta$s, for the cost of the entire attack. Finally, Assumption 4 simply makes the analysis more tractable, since the race phase can then be modeled as a steady state stochastic process.

### 3.1 How large do whale transactions need to be?

The first step in evaluating the cost of the whale attack is to determine what values of whale transactions $\delta$ are appropriate for incentivizing a desired proportion of the network to whale mine. To do this, we examine the decision problem faced by a rational mining entity $m$ in accordance with our assumptions, particularly Assumption 1.

Suppose $m$ has mining power $\beta_m$ and decides to honest mine ($\gamma_m = 0$). This means that $m$ receives block rewards only if the whale attack fails. From Equation 2, the probability that the whale attack fails is equal to $1 - a_z = 1 - min(q/p, 1)^{max(z+1,0)}$. Recall that whale mining power $q = \alpha + \sum_{i=1}^{k} \gamma_i \cdot \beta_i$. Since Alice singly whale mines, $q = \alpha$. Then, honest mining power $p$ is simply equal to $1 - q = \beta$. Conditioned on the whale attack failing, $m$ receives block rewards with probability $\beta_m/p$. It follows that $m$'s profit when honest mining is given by

$$\pi_m(\alpha, \beta_m, \gamma_m = 0, \delta = 0, z) = \frac{(1 - a_z) \cdot \beta_m}{p} = \frac{\left(1 - \left(\frac{\alpha}{\beta}\right)^{z+1}\right) \cdot \beta_m}{\beta}. \quad (4)$$

On the other hand, suppose $m$ decides to whale mine ($\gamma_m = 1$). This means that $m$ receives block rewards only if the whale attack succeeds. The probability that the whale attack succeeds is $a_z = min(q/p, 1)^{max(z+1,0)}$, where $q = \alpha + \beta_m$ and $p = 1 - q = \beta - \beta_m$. Conditioned on the whale attack succeeding, $m$ receives block rewards with probability $\beta_m/q$. Recall that the normal block reward is 1 and the whale block reward is $\delta + 1$. It follows that $m$'s profit when whale mining is given by

$$\pi_m(\alpha, \beta_m, \gamma_m = 1, \delta, z) = \frac{a_z \cdot \beta_m}{q} \cdot (\delta + 1) = \frac{\left(\frac{\alpha+\beta_m}{\beta-\beta_m}\right)^{z+1} \cdot \beta_m}{\alpha + \beta_m} \cdot (\delta + 1). \quad (5)$$

More generally, $m$'s profit is given by

$$\pi_m(\alpha, \beta_m, \gamma_m, \delta, z) = \frac{\gamma_m \cdot \left(\frac{\alpha+\beta_m}{\beta-\beta_m}\right)^{z+1} \cdot \beta_m}{\alpha+\beta_m} \cdot (\delta+1) + \frac{(1-\gamma_m) \cdot \left(1-\left(\frac{\alpha}{\beta}\right)^{z+1}\right) \cdot \beta_m}{\beta}. \tag{6}$$

By rationality, $m$ will choose $\gamma_i \in \{0,1\}$ that maximizes its profit $\pi_m$. Clearly, as long as $\pi_m(\alpha, \beta_m, \gamma_m = 1, \delta, z) > \pi_m(\alpha, \beta_m, \gamma_m = 0, \delta = 0, z)$, in other words, Equation 5 is greater than Equation 4, then $m$ will choose to whale mine. We can then solve for $\delta$ to determine what values of whale transactions make whale mining more profitable.

$$\delta > \frac{\left(1-\left(\frac{\alpha}{\beta}\right)^{z+1}\right)}{\beta} \cdot \frac{\alpha+\beta_m}{\left(\frac{\alpha+\beta_m}{\beta-\beta_m}\right)^{z+1}} - 1, \tag{7}$$

which is equivalent to

$$\delta > \frac{\Pr[\text{whale attack fails} \mid \gamma_m = 0]}{\Pr[\text{honest block} \mid \gamma_m = 0]} \cdot \frac{\Pr[\text{whale block} \mid \gamma_m = 1]}{\Pr[\text{whale attack succeeds} \mid \gamma_m = 1]} - 1.$$

Table 1 provides values for $\delta$, as functions of $\alpha$ and $\beta_m$.

We now point out several insights from Equation 7 and Table 1. First, we see that, in terms of cost, larger mining entities are more easily bribed into whale mining. In fact, as $z$ approaches $-1$, $m$ may choose to whale mine regardless of whether or not there are whale transactions on Alice's branch. An intuitive explanation for this is that, from $m$'s perspective in accordance with Assumption 1, it earns a larger proportion of the block rewards on Alice's branch as long as honest mining power is greater than whale mining power. Thus, as the whale attack becomes more likely to succeed, the expected profit in whale mining for a larger proportion of the block reward becomes greater than that of honest mining and being left empty-handed should the whale attack succeed.

Second, if we convert Equation 7 into a function $f(\alpha)$ and we differentiate with respect to $\alpha$, we see that $f(\alpha)$ is strictly decreasing in the interval $\alpha \in [0, 0.5)$. This insight is rather straightforward and tells us that increasing Alice's mining power $\alpha$ will decrease the cost of the whale attack. Similarly, if we convert Equation 7 into a function $f(\beta_m)$, and differentiate with respect to $\beta_m$, we see that $f(\beta_m)$ is strictly decreasing in the interval $\beta_m \in [0, 0.5)$. This means that if whale mining is profitable for $m$, then whale mining is profitable for all mining entities with mining power greater than or equal to $\beta_m$.

Now, it becomes more clear why Assumption 1 induces an "upper bound" on the cost. By the latter insights, if it is profitable for $m$ to whale mine, then mining entities larger than $m$ will also whale mine. From $m$'s perspective, considering that larger entities will whale mine has the same effect as if Alice were to increase

**Table 1.** The value of $\delta$ that makes whale mining more profitable than honest mining, as a function of the lead of the original branch at the start of the race phase $z$, Alice's mining power $\alpha$, and $m$'s mining power $\beta_m$. For $z = 0$, $\delta$ is always equal to 0.

$z = 6$

| $\alpha$ | 0.00 | 0.05 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 47045879.95 | 503469.42 | 29624.20 | 3448.26 | 574.53 | 117.91 | 27.07 | 6.19 | 0.93 | 0 |
| 0.10 | 531439.89 | 31270.04 | 3639.89 | 606.50 | 124.52 | 28.63 | 6.59 | 1.04 | 0 | 0 |
| 0.15 | 33109.34 | 3854.04 | 642.23 | 131.90 | 30.37 | 7.04 | 1.16 | 0 | 0 | 0 |
| 0.20 | 4094.75 | 682.40 | 140.20 | 32.33 | 7.54 | 1.29 | 0 | 0 | 0 | 0 |
| 0.25 | 727.76 | 149.56 | 34.54 | 8.11 | 1.44 | 0 | 0 | 0 | 0 | 0 |
| 0.30 | 159.96 | 37.00 | 8.74 | 1.61 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.35 | 39.49 | 9.38 | 1.78 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.40 | 9.72 | 1.88 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.45 | 1.51 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$z = 3$

| $\alpha$ | 0.00 | 0.05 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 6857.95 | 689.63 | 161.81 | 52.89 | 20.32 | 8.36 | 3.38 | 1.13 | 0.06 | 0 |
| 0.10 | 727.89 | 170.83 | 55.88 | 21.50 | 8.88 | 3.63 | 1.25 | 0.12 | 0 | 0 |
| 0.15 | 180.79 | 59.18 | 22.80 | 9.45 | 3.89 | 1.38 | 0.18 | 0 | 0 | 0 |
| 0.20 | 62.75 | 24.21 | 10.07 | 4.18 | 1.52 | 0.25 | 0 | 0 | 0 | 0 |
| 0.25 | 25.67 | 10.71 | 4.48 | 1.67 | 0.32 | 0 | 0 | 0 | 0 | 0 |
| 0.30 | 11.28 | 4.75 | 1.80 | 0.39 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.35 | 4.87 | 1.85 | 0.42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.40 | 1.71 | 0.34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.45 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$z = 1$

| $\alpha$ | 0.00 | 0.05 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 17.95 | 7.50 | 4.06 | 2.36 | 1.36 | 0.71 | 0.27 | 0 | 0 | 0 |
| 0.10 | 7.89 | 4.29 | 2.51 | 1.47 | 0.79 | 0.32 | 0 | 0 | 0 | 0 |
| 0.15 | 4.49 | 2.65 | 1.56 | 0.86 | 0.38 | 0.03 | 0 | 0 | 0 | 0 |
| 0.20 | 2.75 | 1.64 | 0.91 | 0.41 | 0.05 | 0 | 0 | 0 | 0 | 0 |
| 0.25 | 1.67 | 0.94 | 0.43 | 0.07 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.30 | 0.90 | 0.41 | 0.05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.35 | 0.32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

her mining power, which we already know decreases $\delta$. Regardless, this does not affect $m$'s decision, since whale mining remains the rational strategy.

### 3.2 How many whale transactions are needed?

The next step in evaluating the cost of the whale attack is to determine how many whale transactions are expected to guarantee that the attack succeeds. Referring back to our assumptions, Alice will keep whale mining power constant by issuing appropriate whale transactions $\delta$ in each block, and she will continue doing so until her branch is longer than the original branch. Setting aside the assumption that Alice never cuts off the attack for a moment (Assumption 6), the race phase we propose in Section 2.1, in which Alice chooses a finite cutoff for the attack $z_{lim}$, is analagous to the Gambler's Ruin problem.

To recap, the initial state in the race phase is the lead of the original branch over Alice's branch $z$. Then, $z$ decreases by 1 with probability $q$, which is the proportion of whale mining power, and increases by 1 with probability $p = 1-q$, which is the proportion of honest mining power. Alice's goal is to reach the absorbing state $z = -1$, before reaching the absorbing state $z = z_{lim}$, at which point she becomes ruined. Although, if Alice becomes ruined, the only costs incurred are the forgone block rewards she could have received mining honestly, not the whale transactions.

Alternatively, we can define the initial state as $z_{lim}$ and the absorbing states as 0 and $S = z_{lim} + z + 1$. Thus, we can calculate the expected number of steps (block creation events) before we hit an absorbing state using

$$E(z_{lim}, z) = \begin{cases} \frac{z_{lim}}{1-2q} - \frac{S}{1-2q} \cdot \frac{(\frac{p}{q})^{z_{lim}}-1}{(\frac{p}{q})^{S}-1} & \text{if } p \neq 0.5 \\ z_{lim} \cdot (z+1) & \text{if } p = 0.5. \end{cases} \tag{8}$$

THen, extending this back to Assumption 6, which stipulates that Alice never cuts off the attack until it succeeds, is simple.

$$\lim_{z_{lim} \to \infty} E(z_{lim}, z) = \begin{cases} \frac{z+1}{2q-1} & \text{if } p \neq 0.5 \\ \infty & \text{if } p = 0.5. \end{cases} \tag{9}$$

The expected number of whale transactions is then

$$\frac{\lim_{z_{lim} \to \infty} E(z_{lim}, z)}{2} + z + 1, \tag{10}$$

since Alice only issues whale transactions in blocks on her own branch.

Now that we have established an informal upper bound on appropriate values for whale transactions and have calculated the number of whale transactions expected, the ultimate question we are trying to answer is "How much does the whale attack cost?" Given the complexity of posing an analytical result for this question, we determine the cost of the attack by simulation. Before we detail our simulations, here are a number of considerations on the cost of the attack.

First, consider that Alice reclaims her own whale transactions with probability $\frac{\alpha}{q}$, which is reflected in our simulations. Second, to interpret our results, recall that $\delta$ is a lower bound on the value of whale transactions for whale mining to be more profitable. Thus, the cost of the attack is marginally more than the sum of the whale transactions in our simulations. Finally, recognize that the whale attack being profitable is different from it being rational. The whale attack is rationally worthwhile for Alice only if the difference between the double-spend $tx_B$ and the cost of the attack is greater than what Alice would have earned simply by honest mining. However, do consider that Alice reaps all of the block rewards from her $m + l$ pre-mined blocks.

### 3.3 Simulation

We model the snapshot of the Bitcoin network shown in Table 2 and we represent Alice by the largest pool in the network ($\alpha = 0.188$). As aforementioned, we are interested in the expected cost to carry out the attack with success probability 1, so we only consider cases in which the whale mining power $q > 0.5$. For example, we run simulations issuing appropriate $\delta$s, such that all pools as large as BTCC Pool will whale mine, to get $q = 0.532$. Similarly, we run simulations issuing appropriate $\delta$s, such that pools as large as BW.COM will whale mine, to get $q = 0.670$, and so on. Table 3 presents the cost of the whale attack in terms of $\delta$ under different parameters of $q$ and $z$.

**Table 2.** Distribution of mining power among the ten largest pools (95% of the network) from July 30-August 2, 2016 (Source: `https://blockchain.info/pools`).

| AntPool | F2Pool | BTCC Pool | BW.COM | BitFury |
|---|---|---|---|---|
| 18.8% | 18.2% | 16.2% | 13.8% | 9.4% |

| HaoBTC | SlushPool | ViaBTC | BitClub Net | Kano CKPool |
|---|---|---|---|---|
| 6.4% | 5.9% | 4.4% | 3.7% | 3.1% |

## 4 Discussion

Our simulations return a number of interesting results. Immediately, we can see the impact that pre-mining has on the cost of the whale attack. As Sompolinsky and Zohar have mentioned, while the $l$ blocks pre-mined before even issuing $tx_B$ may take a long time, as long as Alice controls the timing of the attack and employs selfish mining strategies, mining these $l$ blocks need not be costly [16]. Then, once $tx_B$ has been issued, Alice can mine $m$ more blocks on top of the $l$ guaranteed blocks before $tx_B$ reaches $n$ confirmations to further reduce costs.

Next, we see that centralization of mining increases the venality of the network. As shown in Section 3.1, larger pools are more easily bribed than smaller

**Table 3.** The simulated attack cost (sum of $\delta$s) under different parameters of the whale mining power $q$ and the lead of the original branch at the start of the race phase $z$. The values shown are averages across $10^6$ simulations for each pair of $q$ and $z$.

| $q$ | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0.532 | 2.93e+23 | 3.09e+22 | 8.03e+21 | 1.10e+22 | 2.57e+24 | 2.50e+21 | 4.40e+20 |
| 0.670 | 999.79 | 464.74 | 307.71 | 267.72 | 56.09 | 17.64 | 3.63 |
| 0.764 | 768.09 | 291.86 | 109.89 | 40.16 | 12.73 | 2.48 | 0 |
| 0.828 | 1265.14 | 417.85 | 135.80 | 42.32 | 11.60 | 1.65 | 0 |
| 0.887 | 1205.00 | 390.63 | 123.93 | 37.23 | 9.46 | 1.00 | 0 |
| 0.931 | 1806.67 | 540.75 | 159.34 | 44.66 | 10.69 | 1.12 | 0 |
| 0.968 | 2178.58 | 628.13 | 178.19 | 48.29 | 11.23 | 1.15 | 0 |
| 0.999 | 2598.64 | 723.92 | 198.92 | 52.33 | 11.89 | 1.22 | 0 |

pools. In our simulation, the three largest pools, which includes Alice, already combine for a majority of whale mining power. Since $q = 0.532$ is only slightly above a majority, the cost of the attack is exorbitant. However, simply adding the fourth largest pool for $q = 0.670$ dramatically reduces the cost of the attack. Observing Table 3 for $z = 6$, we see that Alice's cheapest option is to aim for $q = 0.764$. Attempting to bribe the smaller pools, which would allow $z$ to converge faster, would not be cost efficient. Now, consider if mining was completely decentralized, and the largest pools wielded less than 0.01 of the mining power—the whale attack would be incredibly costly in our model.

Finally, consider that Alice only wields $\alpha = 0.188$ of the mining power in our simulations. In the past, mining pools have enjoyed much larger shares of mining power, even exceeding a majority on several occasions. Observing Table 1, we can see that a larger attacker could dramatically reduce the cost of the attack. Thus, we aver that $\alpha = 0.188$ is modest in comparison, and even so, the whale attack need not require an intractable amount of capital. Taking this a step further, our assumptions from Section 3, already induce an upper bound on the cost. We address these assumptions below, and discuss how they might differ in practice.

*Assumption 1.* We briefly discussed this in Section 3, but a more sophisticated mining entity who considers the decisions of other mining entities could dramatically lower the necessary $\delta$ for whale mining to be rational. In practice, cooperative mining entities would achieve similar effects, since they could certainly account for each other's mining power when evaluating the profits.

*Assumption 2.* In practice, if a mining entity mines a large whale block, it would likely be in its best interest to "stick" to whale mining. Consider that it may even be rational to issue their own smaller whale transactions to ensure the success of Alice's branch. From Alice's perspective, the best case (other than if she were to reclaim every whale transaction) would be to have different mining entities each mine a single whale block. If these entities combine for a majority of whale mining power, it is probable that further whale transactions would not be needed at all. Our model assumes the worst case, in which some negligibly

sized mining entity miraculously receives $\frac{1-\alpha}{q}$ of the rewards, thus rendering the other mining entities "unsticky."

*Assumption 3.* In practice, a marginal profit for whale mining over honest mining may not be sufficient, and we would need to consider the "cost of deviation."

*Assumption 4.* In practice, it is not necessary for Alice to keep whale mining power consistent, especially if Alice does not require that the whale attack succeed with probability 1. Perhaps if the purchased merchandise is quite liquid, having the attack fail with nonzero probability would not be a tremendous setback.

*Assumption 5.* As we mentioned before in addressing Assumption 2, there are cases in which it would not be necessary to issue whale transactions until the attack is completed. Additionally, Alice might also choose a finite cutoff for $z_{lim}$, since continuing the whale attack would not be rational if the attack unluckily takes longer than expected.

Our work is primarily a proof-of-concept for the whale attack being feasible for a minority attacker, and we leave open the challenges of modeling the cost of the attack more precisely and exploring the strategy space when combining the whale attack with other mining attacks.

## 5 Conclusion

Cryptocurrencies fail to fit into established theoretical frameworks for secure distributed systems. Instead, their security relies on the assumption that a majority of miners, as measured by their computational resources, will behave honestly. In this regard, researchers have uncovered many deviant mining strategies, which reveal evident security gaps in a rational setting. In this work, we presented the whale attack, in which a minority attacker increases her chances of double-spending by incentivizing rational miners into colluding. Moreover, we demonstrated that such an attack is feasible, even when the attacker wields a modest amount of mining power and capital. While Nakamoto consensus has been a stopgap to the issue of double-spending, we showed that as currently prescribed, it is by no means a panacea.

## Acknowledgments

We thank Elijah Soriah and Andrew Miller for their valuable feedback, and the faculty and students of the CAAR REU program for the wonderful experience.

## References

1. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
2. Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

3. Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980*, 2011.
4. Ittay Eyal. The miner's dilemma. In *2015 IEEE Symposium on Security and Privacy*, pages 89–103. IEEE, 2015.
5. Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.
6. 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.
7. Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. *arXiv preprint arXiv:1507.06183*, 2015.
8. Joseph Bonneau. Why buy when you can rent? bribery attacks on bitcoin-style consensus. In *Proc. 3rd Workshop on Bitcoin and Blockchain Research*, 2016.
9. Jason Teutsch, Sanjay Jain, and Prateek Saxena. When cryptocurrencies mine their own business.
10. Malte Möser and Rainer Böhme. Trends, tips, tolls: A longitudinal study of bitcoin transaction fees. In *Financial Cryptography and Data Security*, pages 19–33. Springer, 2015.
11. Joshua A Kroll, Ian C Davey, and Edward W Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013. Citeseer, 2013.
12. Nicolas Houy. The economics of bitcoin transaction fees. *GATE WP*, 1407, 2014.
13. Kerem Kaskaloglu. Near zero bitcoin transaction fees cannot last forever. 2014.
14. Miles Carlsten, Harry Kalodner, S. Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *ACM Conference on Computer and Communications Security*, 2016.
15. Meni Rosenfeld. Analysis of hashrate-based double spending. *arXiv preprint arXiv:1402.2009*, 2014.
16. Yonatan Sompolinsky and Aviv Zohar. Bitcoin's security model revisited. *arXiv preprint arXiv:1605.09193*, 2016.