

An Analysis of Attacks on Blockchain Consensus (DRAFT)

George Bissas[†], Brian Levine[†], A. Pinar Ozisik[†], Gavin Andresen, Amir Houmansadr[†]
[†]College of Information and Computer Sciences, Univ. of Massachusetts Amherst

Abstract

We present and validate a novel mathematical model of the blockchain mining process and use it to conduct an economic evaluation of the double-spend attack, which is fundamental to all blockchain systems. Our analysis focuses on the *value* of transactions that can be secured under a conventional double-spend attack, both with and without a concurrent eclipse attack. Our model quantifies the importance of several factors that determine the attack’s success, including confirmation depth, attacker mining power, and any confirmation deadline set by the merchant. In general, the security of a transaction against a double-spend attack increases roughly logarithmically with the depth of the block, made easier by the increasing sum of coin turned-over (between individuals) in the blocks, but more difficult by the increasing proof of work required. In recent blockchain data, we observed a median block turnover value of 6 BTC. Based on this value, a merchant requiring a single confirmation is protected against only attackers that can increase the current mining power by 1% or less. However, similar analysis shows that a merchant that requires a much longer 72 confirmations (≈ 12 hours) will eliminate all potential profit for any double-spend attacker adding mining power less than 40% of the current mining power.

1 Introduction

Nakamoto’s blockchain-based distributed consensus algorithm [15] is the basis of Bitcoin and other digital currencies. Despite their widespread adoption, there exists little guidance on how merchants accepting blockchain-based currencies can avoid *double-spend attacks*. Double-spend attacks cannot be prevented in blockchain currencies because they are subject to the FLP impossibility result [10], which says, informally, that consensus cannot be reached in distributed systems that do not set a deadline for when messages (i.e., new blocks) can be received. The only mitigating defense against double-spends is for the merchant to wait for a transaction to receive z confirmations (i.e., $z - 1$ blocks are added after the block in which it originally appeared) before releasing goods to the attacker. Nakamoto derived the probability of success for that defense, but the result is limited because it considers neither the cost of the attack, nor the profit that the attacker stands to gain. Furthermore, Bitcoin’s underlying peer-to-peer network is prone to *eclipse attacks* [12] in which adversaries easily deny service to a targeted peer and occlude its view of the majority’s blockchain. Double-spend attacks can be combined with an eclipse attack for a greater probability of success [12].

This work’s copyright is owned by the authors. A non-exclusive license to distribute the pdf has been given to arxiv.org. Last revised 2016-10-24.

Contributions. We contribute a novel economic evaluation of double-spend attacks in Bitcoin that could easily be extended to similar blockchain currencies such as Litecoin (<https://litecoin.org>) or Zerocash [20]. We derive and validate equations for the value of transactions that can be secured against a double-spending adversary, who may add new mining power, e.g., stolen from a cloud provider or via a botnet [13], or who is a dishonest miner. We also evaluate the double-spend attack when conducted contemporaneously with the eclipse attack.

Our results quantify Bitcoin’s security as a currency. We show that the correct attacker model considers not just the depth of the block and the attacker’s mining power, but also the summed value of coin in the z blocks that is exchanged between individuals (turnover), as well as the coinbase reward and fees. The security of a transaction against double-spend attack increases roughly logarithmically with the depth of the block, made easier by the increasing turnover sum, but more difficult by the increasing proof of work required.

For example, based on the current median turnover value of 6 BTC per block, we determine that transactions with a single confirmation, are protected only against attackers that can increase the current mining power by less than 1%. However, waiting for significantly more confirmations increases a transaction’s security considerably. With 72 confirmations, and based on the median turnover value for 72 blocks, a transaction can be protected from double-spend attack as long as the attacker possesses less than 40% of the current mining power. We also demonstrate quantifiably that double-spend attacks are more effective if conducted concurrent with an eclipse attack whenever fewer than four confirmations are required and the attacker’s mining power exceeds 30% of the current mining power.

2 Background

Using a blockchain as a method for distributed consensus was first proposed by Nakamoto as part of his development of the Bitcoin digital currency [15]. Blockchains allow for an open group of peers to reach consensus, while mitigating Sybil attacks [7] and the limitations of the FLP impossibility result [10] through a *mining* process. The mining process places peers with resourceful computational power at an advantage, but overall, incentivizes miners to reach consensus. Bitcoin and follow-on currencies, such as Zerocash [20], Ethereum [8], and many others [4], also use the blockchain algorithm to manage an electronic payment system.

Unlike physical coins, Bitcoins do not exist as distinct items, but rather as an account balance in an *address*, and therefore are fungible, divisible, and recombinable. Addresses comprise a stored asymmetric cryptographic key and an associated balance. Bitcoin users exchange money through *transactions*, which are analogous to bank checks. A transaction is a message that is cryptographically signed by the payer, stating the exchange of some coins from the payer’s address to the recipient’s address. They are broadcast over Bitcoin’s *peer-to-peer* (p2p) network. Miners on the p2p network each independently agglomerate a set of transactions into a *block*, verify that the transactions are valid, and attempt to solve a predefined proof-of-work problem involving this block and all prior valid blocks. In Bitcoin, this process is dynamically calibrated to take approximately ten minutes per block. Under ordinary, non-adversarial, conditions, the first miner to solve the proof-of-work problem broadcasts her solution to the network, adding it to the ever-growing *blockchain*; the miners then start over, trying to add a new block containing the set of transactions that were not previously added. When transactions appear on a block, they are *confirmed*. If two miners discover a new block simultaneously, the blockchain will bifurcate. Miners will attempt to

add to only one of the equal length branches. All miners will switch to adding to the first branch that grows longer. In general, miners will switch to adding to a branch presented to them at any point that represents the largest proof of work.

As incentive, all miners insert, as the first item in their block, a *coinbase* transaction, which is the protocol-defined creation of new Bitcoins and a transfer of those coins into a address of their choosing. In doing so, they have *mined* those coins and made the chosen address and its balance valid in future transactions. Second, miners receive a small fraction of the face value of all transactions in the block that they successfully add to the blockchain; this transaction fee overhead serves to incentivize miners even after the last protocol-defined Bitcoin is mined. Miners are commonly organized into *mining pools*, which allow many miners to pool together their resources. In these pools, rewards are split equitably according to the amount of resources they contributed to creating a block.

2.1 Limitations

Double spending. A fundamental attack against Bitcoin is the *double-spend* attack [15], which works as follows. An attacker creates a transaction that moves funds to a merchant’s address. After the transaction appears in the newest block, the attacker takes possession of the purchased goods. Using his mining power, the attacker then immediately releases two blocks, with a transaction in the first that moves the funds to a second attacker-owned address. Now the attacker has the goods and his coin back. To defend against the attack, a merchant can refuse to release goods to a Bitcoin-paying customer until z blocks have been added to the blockchain including the first block containing a transaction moving money to the merchant’s address. Nakamoto calculated the probability of the attack succeeding assuming that the miner controlled a given fraction of the mining power [15]; for a given fraction, the probability of success decreases as z increases.

In general, a merchant may wait z blocks before releasing goods, which can thwart an attacker. But choosing the minimum value of z that secures a transaction is an unresolved issue. The core Bitcoin client shows that a transaction is unconfirmed until it is 6 blocks deep in the chain [3], and the advice from researchers to policymakers can be vague; e.g., “for very large transactions, coin owners might want to wait for a larger number of block confirmations” [5]. As we describe in Section 5, Rosenfeld [19] derived a simple model for calculating the value of transactions secure from the double-spend attack that is a great improvement over the 6-block rule. We develop a richer, continuous-time model that explicitly accounts for attacker cost as a function of mining duration. Our approach has the added benefit of being capable of modeling more sophisticated threats such as the eclipse attack, which is discussed presently. In Section 3, we also show that our model offers improved accuracy over Rosenfeld’s.

Eclipse Attacks. Heilman et al. showed that Bitcoin’s p2p network peer discovery mechanism is vulnerable to eclipse attacks [12], which occlude a victim peer’s view of the blockchain. For example, if an adversary controls a botnet, he can fill a peer’s table of possible neighbors, resulting in a very high chance the victim will connect only to the attacker. Alternatively, the eclipse can involve controlling a victim’s local connection to the Internet.

Heilman et al. also showed that eclipse attacks can be used as a tool to increase the effectiveness of the double-spend attack on a merchant. First, the attacker eclipses the merchant’s view of the blockchain. Then, he sends the merchant a seemingly honest transaction \mathcal{H} , which contains the payment for a good. Third, the attacker sends to the miners a faulty transaction \mathcal{F} that moves the funds elsewhere. Next, he creates and sends a

series of z blocks to the victim merchant such that \mathcal{H} is part of the first block. Finally, he continues the eclipse until the real blockchain has progressed by at least $z + 1$ blocks. At that point, he has both the goods and the Bitcoin that the merchant intended to keep.

Heilman et al. offered a detailed analysis of the mechanics of an eclipse attack, as well as several protocol-level defenses to the attack. But they offered no insight into the economic incentives for the attacker. As a result, it remains unclear what minimum number of confirmations, z , are sufficient to secure a given value of purchased goods. In this paper, we derive a model for the profit received by an adversary who launches an eclipse attack, and use it to determine the attacker’s breakeven point for various values of z .

3 Economic Analysis of Doublespend and Eclipse Attacks

In this section, we compute the value of transactions v that can be secured from an attacker given their mining power q and the number of waiting blocks z chosen by the merchant. Using the same techniques, we are able to determine the mining power q required to profitably double spend a transaction of value v that is z blocks deep in the longest chain. As we explain below, v is the sum amount of coins transferred between entities (turnover) by all transactions in a single block — particularly, the block in which the transaction moving funds to the merchant appears. In fact, as we discuss below, the most conservative evaluation of Bitcoin’s security is to calculate v as the *summed* turnover in the set of z blocks.

Our guiding principle is that *a resource is secure from an attack only if it is worth less than the attack’s cost*. We find that because of the well-behaved statistical properties of mining times and transparency of transaction values, Bitcoin is particularly amenable to such an analysis.

Attack Cases. Any analysis of the double-spend attack is particular to both the attack and defense strategies. For example, one double-spend strategy for the attacker is to first eclipse the merchant’s view of the blockchain [12]. Currently, merchants will wait without a time deadline for z blocks to arrive because inter-block delays are variable. In an analysis of blocks from 2016, we found that 20% of inter-block delays are greater than 15 minutes, and 5% are greater than 30 minutes. Unfortunately, in this scenario, the attacker will achieve 100% success in this case if she deploys an eclipse attack.

In this section, we analyze two double-spend attack strategies. The first case assumes that the attacker uses an eclipse attack to double spend. We show that if the merchant both (i) sets z proportional to the *value of the block the transaction appears in* and (ii) sets a *deadline* for receiving the z blocks, then, in order to expect success, the attacker will be forced to spend resources that increase proportionally with the current coinbase reward and z .

In the second case, we assume that the attacker elects not to employ an eclipse attack (or equivalently, fails in an attempt to do so). In this case, as we show below, if the attackers do not add more than about 1/3 of the existing mining power to the system, then preventing the eclipse attack increases the value of the transactions that can be secured from an attacker. However, when the attacker adds more than about 1/3 of the existing mining power to the system, eclipsing the merchant actually increases the breakeven point of an eclipse attack, thereby rendering it less-effective.

3.1 Attacker model

We assume the attacker **adds mining power** to the system to execute the attack, where the amount he adds is expressed as a fraction of the existing mining power ranging from $0 < q < 1$. When $q \geq 1$, the attacker holds at least 50% of the aggregate mining power, in which case Bitcoin cannot secure any transaction. With trivial adjustments, our equations can also be made to model double-spend attacks by existing miners.

We model attacker resources conservatively from the merchant’s perspective. In reality, attackers may find costs are higher. For example, we assume that the attacker is able to lease mining power for any exact duration of time and without fixed fees per attack. We assume the cost of leasing computational resources is equal to the opportunity cost from not using those resources for mining honestly: $qB/10$ per minute, where B is the coinbase reward for discovering a block. In reality, leasing resources for general computation may have higher costs than a miner’s custom ASICs; however, we expect that resources that cost less than B would be used for mining and increase the network’s *difficulty*¹ accordingly. The average value of the fees from transactions may be added to the value of B if desired, but for simplicity, we elide them from consideration. Currently, fees per block sum to about 0.5 BTC, while the coinbase is $B = 12.5$ BTC.

We assume that the network is correctly calibrated so that a block is produced about every 10 minutes given the current mining power, which is generally true in the real system. Bitcoin adjusts its difficulty once every 2,016 blocks (about every two weeks), and we assume these attacks have no effect on the difficulty while they are run.

We assume that the eclipse attack succeeds without fail. An eclipse attack is likely to incur some cost, but we do not include it because that cost is hard to estimate. For example, in the most general scenario, a botnet might be required [12]. On the other hand, if a merchant is physically accessible and has only a single, unsecured wireless link to the Internet, eclipse attacks are much simpler and much less costly. We also assume the attacker does not launch a denial-of-service attack on honest miners.

Finally, we assume that the attacker is capable of targeting multiple merchants simultaneously, which allows him to amortize the cost of mining blocks for the purpose of double-spending both with and without the eclipse attack. However, we will frequently refer to only a singular “merchant” with the understanding that the activity should be applied to *each* merchant being targeted.

Attacker Strategy: Case 1. The attacker launches an eclipse attack against one or more merchants, and using mining power q , he sends z blocks with a transaction moving coins to the merchant in the earliest block. Once the merchant releases the goods to the attacker and the honest miners reach $z + 1$ blocks (ensuring the attacker’s blocks will not be accepted on the main chain), the attacker can cease to eclipse the merchant and the attack ends. The merchant will refuse to hand over the goods if the z blocks are not received by her *deadline*. The attacker stops mining (and paying leasing costs) once he mines z blocks or the deadline has passed. Because the z blocks are never added to the main chain, the coinbase rewards earned are useless. (We do not consider the case where the attacker attempts to replace the main chain with his false chain.)

Attacker Strategy: Case 2. No eclipse attack is leveraged. The attacker uses mining power q and attempts to discover $z + 1$ blocks, which he temporarily holds. He releases the blocks once he has taken possession of the goods and the honest miners have produced z

¹Bitcoin’s *difficulty* determines the computation required to discover a block in about 10 minutes.

blocks. We assume the attacker stops if he cannot discover $z + 1$ blocks before the honest miners announce $z + 1$ blocks. The attacker in our model does not try to reach $z + 2$ or farther and continue the race (called the Gambler’s Ruin [9]). In general, more advanced strategies are possible, and they require a more advanced model than what we derive here. We leave discovering the optimal attacker strategy for future work.

Parallel Attacks. In either case above, an attacker can increase his profit by mining a block that contains multiple double-spends for simultaneous attacks on multiple merchants. Suppose that the merchant requires z confirmations before delivering the goods. It is possible that all other merchants under attack require only a single confirmation, which means the attacker could profit from transactions in all z confirmation blocks. To account for the worst-case attacker strategy, the merchant determines a value for z based on v , which is the sum amount of the coins within the z blocks that are transferred, or *turned over*, between distinct entities; several past works have evaluated metrics for estimating that value [14,18]. The merchant cannot know the turnover value for a block *a priori*, which makes it impossible to be confident in the value v . One solution is to derive v based on the historical average turnover per block. Another solution is for the merchant to calculate v progressively and adjust z based on the current cumulative value of v . This would of course require that the terms between merchant and customer be amended with regard to the required number of confirmations before the release of goods.

3.2 Case 1: Eclipse-Based Double-Spend

We wish to determine Bitcoin’s security with respect to an eclipse-based double-spend attack by quantifying a potential attacker’s economic breakeven point under this strategy. Breakeven occurs when the revenue R less the cost C is equal to zero, *i.e.*,

$$R - C = 0. \quad (1)$$

As described above, we assume the attacker adds to the system’s hash power equivalent to fraction q of the total honest mining power, and the merchant requires that the payment transaction receive at least z confirmations before she will release the goods. We further assume that the merchant imposes a deadline of d minutes to the attacker, *i.e.*, the sale is nullified if the z th confirmation block has not been mined after d minutes.

Let X_i^q be a random variable representing the time it takes for the attacker to mine the i th block using mining power q , and let

$$X = \sum_{i=1}^z X_i^q. \quad (2)$$

Here, X represents the time it takes the attacker to reach z blocks using mining power q . Define $C(x; z, d, q)$ to be the cost of an attack with duration x minutes and deadline d that uses mining power q . To calculate cost, we assume that the miner will stop mining once he mines the z blocks, but will continue to mine until the deadline if he is unsuccessful. We assume further that cost can be measured in terms of the opportunity cost for diverting the mining power from performing honest mining, which means that the attacker could alternatively earn the block reward of B for mining on the main chain. Since blocks are expected to be mined every 10 minutes on the main chain, it follows that

$$C(x; z, d, q) = \begin{cases} \frac{qx B}{10}, & x \leq d \\ \frac{qd B}{10}, & x > d \end{cases}. \quad (3)$$

Mining is an example of a Poisson process because under constant mining power, blocks are mined continuously and independently at a constant average rate. Therefore $X_i^q \sim \text{exponential}(\beta)$ with $\beta = 10/q$. It is well known that the exponential distribution is a special case of the gamma distribution with shape parameter $\alpha = 1$. Furthermore, the sum of z gamma distributions with shape $\alpha = 1$ and the same rate β is again gamma with rate β and shape z . Thus $X \sim \text{gamma}(z, 10/q)$. Let $g(x; \alpha, \beta)$ be the density function for the distribution $\text{gamma}(\alpha, \beta)$, and let $G(x; \alpha, \beta)$ be the CDF. It follows then that

$$\begin{aligned}
E[C(X, z, d, q)] &= \int_0^\infty C(x; z, d, q) g(x, z, \beta) dx \\
&= \int_0^\infty C(x; d, q) \frac{1}{\beta^z (z-1)!} x^{z-1} e^{-x/\beta} dx \\
&= \frac{qB}{10} \left[\int_0^d \frac{1}{\beta^z (z-1)!} x^z e^{-x/\beta} dx + d(1 - G(d, z, \beta)) \right] \\
&= \frac{qB}{10} \left[\beta z \int_0^d \frac{1}{\beta^{(z+1)} z!} x^z e^{-x/\beta} dx + d(1 - G(d, z, \beta)) \right] \\
&= \frac{qB}{10} \left[\frac{10z}{q} G\left(d; z+1, \frac{10}{q}\right) + d \left(1 - G\left(d; z, \frac{10}{q}\right)\right) \right] \\
&= \frac{qdB}{10} + zBG(d; z+1, 10/q) - \frac{qdB}{10} G(d; z, 10/q). \tag{4}
\end{aligned}$$

Consider now the attacker's revenue $R(x; d)$. If he succeeds in the attack, then he will earn revenue v and will earn nothing otherwise. Formally,

$$R(x; d) = \begin{cases} v, & x < d \\ 0, & x \geq d \end{cases}. \tag{5}$$

The probability of his success is

$$P(X \leq d) = G(d; z, 10/q). \tag{6}$$

Hence the expected revenue is given by

$$E[R(X; d)] = vG(d; z, 10/q). \tag{7}$$

Using the fact that expected breakeven occurs when $E[R(X; d)] - E[C(X; z, d, q)] = 0$ we have our **result for Case 1**:

$$\begin{aligned}
v &= \frac{E[R(X; d)]}{G(d; z, 10/q)} \\
&= \frac{E[C(X; z, d, q)]}{G(d; z, 10/q)} \\
&= \frac{\frac{qdB}{10} + zBG\left(d; z+1, \frac{10}{q}\right) - \frac{qdB}{10} G\left(d; z, \frac{10}{q}\right)}{G(d; z, 10/q)} \\
&= \frac{\frac{qdB}{10} + zBG\left(d; z+1, \frac{10}{q}\right)}{G(d; z, 10/q)} - \frac{qdB}{10}. \tag{8}
\end{aligned}$$

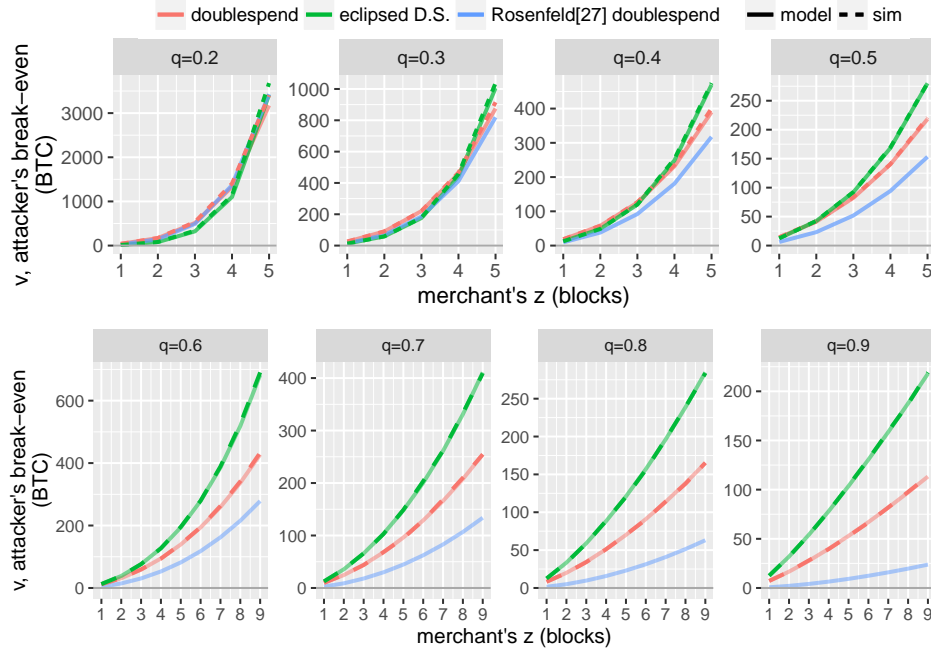


Figure 1: For an attacker that can add q mining power to the system, these plots show the value v required to expect to breakeven given a value z selected by the merchant. The result depends on whether the attacker performs a standard double-spend attack (Eq. 18) in red; or an eclipse-based double-spend attack (Eq. 8) in green. For the eclipse attack we set the deadline to $d = 10z$ minutes. Results from the model are shown as solid, transparent lines; results from an independent Monte Carlo simulation of the attacker are shown as a dashed line. Because our model matches the simulation exactly, the model and simulation curves coincide. Rosenfeld’s model [19] is also shown in blue; in places where it is accurate his curve coincides with the red curve.

3.3 Case 2: Double-Spend Without The Eclipse Attack

In this case, we assume that an eclipse attack is employed by the attacker. By comparing results to Case 1, we can determine the breakeven point under a double-spend attack where the attacker is better off not eclipsing the merchant.

For clarity, we name the chain that the attacker builds (holding the double-spend transaction) the *fraudulent chain* and the chain that the honest miners build (holding the payment transaction) the *main chain*. In order for the attack to succeed, the fraudulent chain must eventually be adopted by the community as the main chain. And so it does not make sense for the merchant to impose a confirmation deadline because it cannot increase the difficulty of the attack without adversely affecting honest customers. She does, however, still enforce an embargo on goods until the transaction transferring funds to her has been confirmed z times.

Bitcoin changes the expected total number of hashes computed by all miners required to find a new block only once every two weeks (and specifically not a sliding window of the last two weeks to avoid certain attacks). Therefore, an attacker who adds resources will

not change the expected number of hashes per block (*i.e.* the difficulty remains unchanged) unless the attack happens to straddle a difficulty update, which the attacker can most likely avoid.

Let Y_i^q be a random variable denoting the time it takes the attacker to mine the i th block if he controls fraction $q/(1+q)$ of the total mining power. Similarly, define M_i^q to be the time it takes the honest miners to mine block i given that they control $1/(1+q)$ of the mining power. Finally, define

$$Y = \sum_{i=1}^{z+1} Y_i^q \quad (9)$$

and

$$M = \sum_{i=1}^{z+1} M_i^q \quad (10)$$

to be the time it takes for the attacker and other miners, respectively, to mine $z+1$ blocks. In order for the attack to be a success, it must be the case that $Y < M$. We assume that the attacker will stop mining when he reaches $z+1$ blocks on the fraudulent chain or when the honest miners reach $z+1$ blocks on the main chain, whichever happens first. Analogous to Section 3.2 we can define $C(y, m; z, q)$ as the cost to the attacker when the merchant requires z confirmation blocks and the attacker possesses mining power q :

$$C(y, m; z, q) = \begin{cases} \frac{qyB}{10}, & y \leq m \\ \frac{qmB}{10}, & y > m \end{cases} \quad (11)$$

Just like X in Section 3.2, both Y and M have gamma distributions, this time with rate parameters $\beta_Y = 10/q$ and $\beta_M = 10$, respectively. Specifically, $Y \sim \text{gamma}(z+1, \beta_Y)$ and $M \sim \text{gamma}(z+1, \beta_M)$. Define $g(x; \alpha, \beta)$ and $G(x; \alpha, \beta)$ as in Section 3.2. It follows that

$$\begin{aligned} E[C(Y, M; z, q)] &= \int_0^\infty \int_0^\infty C(y, m; z, q) g(m; z+1, \beta_M) g(y; z+1, \beta_Y) dy dm \\ &= \int_0^\infty g(m; z+1, \beta_M) \left(\frac{qB}{10} \int_0^m yg(y; z+1, \beta_Y) dy + \right. \\ &\quad \left. \frac{qmB}{10} \int_m^\infty g(y; z+1, \beta_Y) dy \right) dm. \\ &= \frac{qB}{10} \left(\int_0^\infty g(m; z+1, \beta_M) \int_0^m yg(y; z+1, \beta_Y) dy dm + \right. \\ &\quad \left. \int_0^\infty mg(m; z+1, \beta_M) \int_m^\infty g(y; z+1, \beta_Y) dy dm \right) \end{aligned}$$

$$\begin{aligned}
&= \frac{qB}{10} \left(\int_0^\infty g(m; z+1, \beta_M) \int_0^m \beta_Y(z+1) g(y; z+2, \beta_Y) dy dm + \right. \\
&\quad \left. \int_0^\infty \beta_M(z+1) g(m; z+2, \beta_M) \int_m^\infty g(y; z+1, \beta_Y) dy dm \right) \\
&= \frac{qB(z+1)}{10} \left(\beta_Y \int_0^\infty g(m; z, \beta_M) G(m; z+2, \beta_Y) dm + \right. \\
&\quad \left. \beta_M \int_0^\infty g(m; z+1, \beta_M) (1 - G(m; z+1, \beta_Y)) dm \right) \\
&= \frac{qB(z+1)}{10} \left(\frac{10}{q} \int_0^\infty g(m; z, 10) G(m; z+2, 10/q) dm + \right. \\
&\quad \left. 10 \int_0^\infty g(m; z+1, 10) (1 - G(m; z+1, 10/q)) dm \right) \tag{12}
\end{aligned}$$

In order to determine the attacker's expected breakeven point, we must also calculate his expected revenue. For any given z , the attacker's revenue when it took him y minutes to mine $z+1$ blocks on the fraudulent chain while it took the other miners m minutes to mine $z+1$ blocks on the main chain is given by the following:

$$R(y, m; z) = \begin{cases} v + (z+1)B & y < m \\ 0 & y \geq m \end{cases} \tag{13}$$

Revenue differs from that collected from the eclipse attack in Section 3.2 because if the attack is successful, the fraudulent chain will become the main chain and the attacker will also earn the coinbase reward for each block he mines. The probability of attack success is equal to

$$\begin{aligned}
P[Y < M] &= \int_0^\infty \int_0^m g(m; z+1, 10) g(y; z+1, 10/q) dy dm \\
&= \int_0^\infty g(m; z+1, 10) G(m; z+1, 10/q) dm. \tag{14}
\end{aligned}$$

Therefore, the expected reward can be calculated as

$$\begin{aligned}
E[R(Y, M; z)] &= \int_0^\infty \int_0^m (v + (z+1)B) g(m; z+1, 10) g(y; z+1, 10/q) dy dm \\
&= (v + (z+1)B) \int_0^\infty g(m; z+1, 10) G(m; z+1, 10/q) dm \\
&= (v + (z+1)B) P[Y < M]. \tag{15}
\end{aligned}$$

Finally, we determine the expected breakeven point by finding the value of v for which revenue minus cost is zero.

$$E[R(Y, M; z)] - E[C(Y, M; z, q)] = 0. \tag{16}$$

Substituting for revenue,

$$(v + (z+1)B)P[Y < M] = E[C(Y, M; d, q)], \tag{17}$$

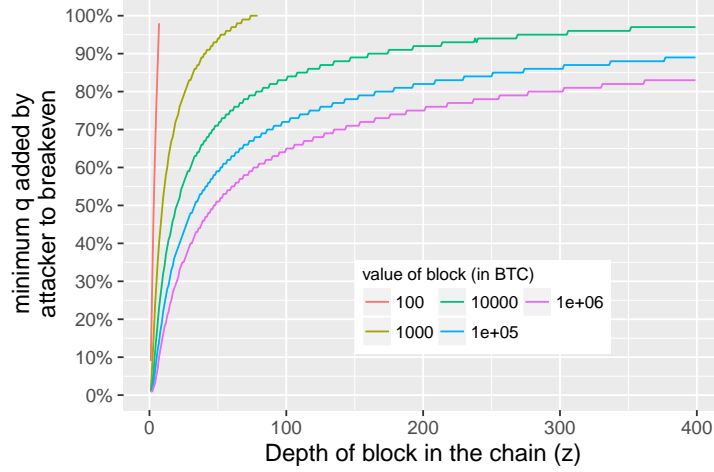


Figure 2: Minimum mining power values q added by an attacker required to breakeven for given merchant confirmation requirement z and potential attack revenue v . For each value of z and v , the most effective double-spend attack is used (eclipse or not-eclipse based) to find the lowest value q that allows the attacker to breakeven.

and rearranging, we have our **result for Case 2** as follows:

$$\begin{aligned}
 v &= \frac{E[C(Y, M; d, q)] - (z + 1)BP[Y < M]}{P[Y < M]}, \\
 &= \frac{E[C(Y, M; d, q)]}{P[Y < M]} - (z + 1)B
 \end{aligned} \tag{18}$$

Note that Eq. 18 is fully expressed by substituting in Eqs. 12 and 14.

4 Discussion

In this section, we use the derivations from the previous section to quantitatively compare attacker strategies and discuss implications for blockchain systems and their users.

Better to eclipse or not? Figure 1 compares Eqs. 8 and 18, the breakeven value for a rational attacker performing double-spend attack with and without eclipsing the merchant, respectively. For lower values of q , we limit $z \leq 5$ so that the eclipse attack's advantage can be distinguished. When the attacker adds mining power equal to $q = 0.2$, we see that the eclipse attack provides him with a slight advantage for values of $z \leq 4$. However, once the attacker's added mining power exceeds $q = 0.4$ the advantage disappears, and for values of $q \geq 0.5$, it becomes universally better to attack without eclipsing the merchant. An independent Monte Carlo simulation of both attacks, which is also plotted in the figure, agrees with the model perfectly.

Comparison to Rosenfeld [19]. Rosenfeld offers a model for v for what we call Case 2 (double-spend without the eclipse attack), but he does not address Case 1. In our terms,

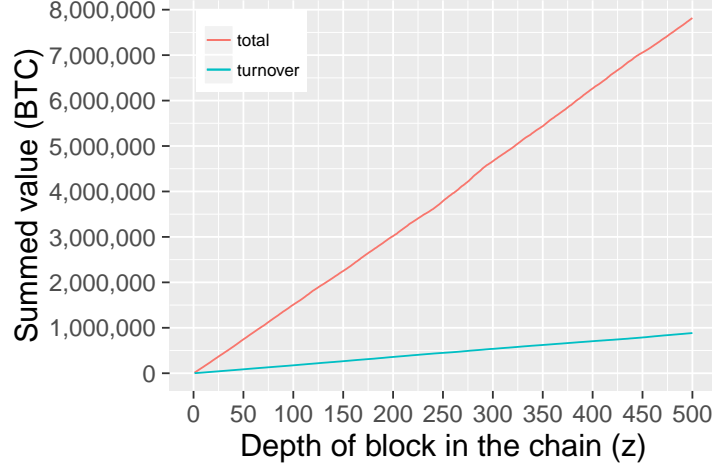


Figure 3: Comparison between total (red) and turnover (blue) block values. The plots are based on actual Bitcoin block data from July 2016. Each point, (z, v) , on either curve corresponds to the median aggregate block value v (either total or turnover) across a sliding window of z consecutive blocks throughout the month.

his equation is $v > (1 - r)zB/r$, where r is a discrete model² of the attacker’s probability of success given z and q . Figure 1 plots this model, showing that it is a good fit for lower values of q (where it matches Monte Carlo simulation results), but quickly falls away from the exact value that our analysis provides.

Revenue required to breakeven. In general, double-spending attacks are more efficient with higher mining power q because there is less risk of losing. As a result, the attacker cannot profit from some targets unless he procures a certain minimum amount of mining power. Figure 2 shows the mining power necessary for the attacker to breakeven for various revenue values v and merchant confirmation requirements z . The points are generated by means of Monte Carlo integration. For each pair of values z and v , the most effective double-spend attack strategy is reported (the strategy that breaks even for lowest value q). From the plot we can see that, for low values of z , nearly any potential revenue is suitable for an attacker with limited mining power. On the other hand, as the merchant increases z , the required mining power increases rapidly for low revenue attacks. For example, an attack that targets 1M BTC in revenue (purple points) can be successful with relatively low mining power, $q = 0.35$, as long as the merchant keeps z less than 25. But a lower revenue target, such as 1K BTC (yellow points), would require the attacker to possess mining power $q = 0.75$ for the same value z .

Parallel attack. The attacker’s potential profit has a strong impact on his breakeven point, and to be conservative, the merchant must assume the attacker is capable of profiting from all turnover in the blocks that confirm the sale of her goods. Figure 3 shows the actual median aggregate total and turnover values for consecutively mined blocks during the month of July, 2016. Each point should be thought of as the typical sum of all outputs for the transactions (red curve) or typical turnover value (blue curve) for the given number of

²Specifically, Eq. 1 in [19], where our q must be converted to $q/(1 + q)$ to match his notation.

consecutive blocks (shown on the independent axis). The plot shows that the merchant can significantly bound the attacker’s potential revenue by measuring turnover as opposed to using total block output value. The turnover values can also be used in conjunction with Figure 2 to determine the merchant’s security against an attacker with known mining power.

Attacks by miners. Miners have an advantage over third parties when launching the non-eclipse version of the double-spend attack: they take away mining power from the honest set of miners. In the model above, a third party that garners q resources has an effective mining power of $q/(1+q)$. Existing miners remove some of their own competition. A miner with $1/3$ of the current mining power would be matched only by a third party bringing $q = 1/2$ (since $\frac{1/2}{1+1/2} = 1/3$). Miners have no advantage in the eclipse attack since the deadline is fixed.

Attacker resources. In order to procure mining power, an attacker must either rent or purchase mining equipment and pay for the required electricity. There does not currently exist a large-scale market for renting mining equipment, and electricity prices vary widely by region. Here we quantify only the cost associated with purchasing the mining equipment necessary to launch an attack.

The Antminer S7 Bitcoin mining hardware produces 4.73 Terahashes/sec and costs US\$480, therefore an attacker would need to spend around US\$102 to purchase a single Th/s of mining equipment. On the other hand, the current hashrate of the network is about 1,500,000 Th/s. So the attacker must spend US\$153M to purchase enough mining equipment to reach the current network hashrate. However, when the merchant sets $z = 1$, attackers can breakeven with mining power $q = 0.01$. This amount of mining power would require only about US\$1.5M in mining equipment. In this case, the cost of purchasing the equipment could potentially be defrayed by a series of relatively small attacks on the order of hundreds of thousands of dollars. Once the attacks have completed, the attacker could potentially sell some of his mining hardware. There is an active secondary market for the Antminer S7. A recent listing on eBay³ reports an auction sale price of US\$400, suggesting that the Antminer retains more than 80% of its retail value.

It is also possible for the attacker to gain access to a botnet. The most valuable assets to a miner in a botnet are the GPUs. Currently the highest hashrate achievable with a GPU is approximately 2.1 Gh/s.⁴ Thus, in order for an attacker to muster a fraction $q = 0.01$ of the current network hashrate, he would need access to a botnet with at least 7.14 million computers. Small increments in q quickly increase this count.

Coinbase and Fees. Bitcoin’s security is critically related to the reward for mining. On July 9, 2016, the rewarded coinbase halved from $B = 25$ to $B = 12.5$. In 2020, Bitcoin’s security against double-spends will decrease further since the coinbase reward will halve again. A lower block reward absent higher fees or a significant increase in Bitcoin’s fiat-exchange value will make it cheaper for attackers to procure a higher percentage of the mining power. Hence, if conditions remain the same except for a decrease in coinbase reward, then merchant’s will need to wait for more confirmation before releasing goods.

Setting the deadline. The eclipse attack’s deadline d is a parameter in Eq. 8. Lower values of d increase the breakeven value for the attacker, increasing the security provided by the merchant’s choice of z . The downside to securing goods with a shorter confirmation deadline is that honest customers may not succeed in meeting the deadline due to the inherent randomness of block discovery. For example, when $d = 10z$ minutes, 60% of blocks

³<http://www.ebay.com>

⁴https://en.bitcoin.it/wiki/Non-specialized_hardware_comparison

in 2016 arrived within that timeframe for $z \in [1, 6]$. In cases where the deadline is missed, the merchant should return the transferred funds. A third-party escrow can enforce a fair exchange of coin before the goods are released [16].

Advice to Bitcoin merchants. Until the problem of eclipse attacks is solved, merchants are best off setting parameters to address one of two cases. (1) Merchants can assume miners are not carrying out double-spend attacks, and that no outside party can reasonably acquire $q > 1\%$ mining power to add to the system. In this case $z \geq 1$ is sufficient security for the median block turnover. (2) Assume miners are dishonestly carrying out eclipse-based double-spend attacks, which is harder to defend against. In this case, most of the benefit from the embargo is achieved by $z = 72$ (≈ 12 hours) which requires attacking miners have 40% of the *existing* mining power for the median block turnover.

Applications to off-chain and side-chain protocols. In its full generality, our analysis quantifies the security of an exchange of some off-blockchain quantity for Bitcoin. Thus far, we have imagined a merchant trading physical goods or services. But our results apply equally to many systems that rely on or assume a stable blockchain, including off-blockchain protocols such as Sidechains [1] and Zerocash [20], micropayment channels such as the Lightning Network [17] and TumbleBit [11], and protocols that rely on stability such as the XIM decentralized mix service [2].

All these alternate protocols require a certain transaction, \mathcal{T} , be confirmed in a Bitcoin block that locks or moves coin while in use by the other protocol. If the Bitcoin miners subsequently switch to a chain that doesn't include \mathcal{T} (or worse, includes a transaction that prohibits \mathcal{T} 's validity in future blocks), then a *reorganization* of the alternate protocol's blockchain or transaction is required, resulting in havoc. Unfortunately, due to the FLP impossibility result [10], it is always possible for an attacker with sufficient resources to force a reorganization. And so in all cases, these protocols vaguely recommend the block containing the transaction reach a sufficient depth. For example, Back et al [1] recommends that "a typical confirmation period would be on the order of a day or two." Sasson [20] recommends that users with "sensitive transactions only spend coins relative to blocks further back in the ledger."

Using our analysis, the confirmation depth required for \mathcal{T} can be more precisely calculated, and the risk of the reorganization can be quantified as follows. First, participants wait for \mathcal{T} to be confirmed in a block. Second, the sum turnover of the block is calculated. Over time, the depth of the block and summed turnover will both increase. At each depth, using essentially Figure 2 (which is an expression of equations 8 and 18), the transaction participants determine the mining power q required by an attacker to overwrite the block containing \mathcal{T} ; once the value of q is sufficiently large in the view of the transaction participants, they can proceed in the alternate protocol.

5 Related Work

In addition to the works listed in Sections 2, the following research is related to our contributions. Tschorsch et al. [22], Bonneau et al. [4], and Croman et al. [6] offer summaries of broader Bitcoin research issues.

Satoshi Nakamoto derived the double-spend attack's success probability in the original Bitcoin paper [15]. But the work most related to our economic analysis is Rosenfeld's whitepaper [19], shown in Fig. 2. He analyzed the value of transactions that can be secured from double spending with a simple model based on discrete distributions. Our model is

richer in several ways. Rosenfeld did not analyze the case where attackers rely on an eclipse attack; did not include the coinbase reward when attackers win; did not address parallel attacks on multiple merchants by considering the amount of coin changing hands in a block. Additionally, in his model, if the honest miners are lucky and reach $z + 1$ blocks before the attacker has leased computation for even 1 block, Rosenfeld’s model will overcharge the attacker for $z + 1$ blocks.

Sompolinsky and Zohar [21] extend Nakamoto’s original analysis to cases where the attacker is capable of *pre-mining* blocks on a secret branch at little or no opportunity cost. Such circumstances may arise when the attacker is a member of a selfish mining pool, which routinely mines blocks on secret branches. The idea is for the attacker to pre-mine a transaction that double-spends funds that he intends to use for payment to a merchant in the future. Once his double-spend is $z + 1$ blocks deep into a secret, unpublished branch, the attacker makes a purchase with an unsuspecting merchant who happens to require z confirmations. The attacker waits for the legitimate transaction to be confirmed z times and then releases his branch of length $z + 1$, thereby completing the double-spend. The paper quantifies the advantage that is afforded to the attacker through selfish mining in terms of the number of blocks required to carry out a double-spend attack. Under the assumption of a given selfish mining hash rate α and block acceptance rate γ , their results show how many confirmation c the attacker can begin ahead of the merchant. Such an advantage can easily be incorporated into our model by simply changing the attacker’s block target from z to $z - c$. We note that pre-mining in the context of the eclipse attack may not be feasible since an eclipse cannot generally be carried out for an indefinite period of time. Nevertheless, we intend to update both of our double-spend analyses to account for pre-mining in future work.

6 Conclusion

In this paper we have presented a novel economic model of Bitcoin double spend attacks that incorporates the depth of the block containing the transaction of interest, the attacker’s mining power, economic turnover, and coinbase reward. Based on this model, we have shown that a merchant can protect themselves from an attacker roughly logarithmically with the depth of the block, where an attacker benefits from the increasing turnover sum but is also throttled by the increasing proof of work required. Additionally, we have demonstrated that the eclipse attack is not particularly a desirable strategy for attackers who can add more than 30% of the current mining power or roughly for $z \geq 3$.

References

- [1] Back, A., Corallo, M., Dashjr, L., Mark, F., Maxwell, G., Miller, A., Poelstra, A., Timón, J., Wuille, P.: Enabling Blockchain Innovations with Pegged Sidechains. <http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains> (October 2014)
- [2] Bissias, G., Ozisik, A.P., Levine, B.N., Liberatore, M.: Sybil-Resistant Mixing for Bitcoin. In: Proc. ACM Workshop on Privacy in the Electronic Society (November 2014), <http://forensics.umass.edu/pubs/bissias.wpes.2014.pdf>
- [3] Confirmation. <https://en.bitcoin.it/wiki/Confirmation> (February 2015)

- [4] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J., Felten, E.: Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In: IEEE S&P. pp. 104–121 (May 2015), <http://doi.org/10.1109/SP.2015.14>
- [5] Bonneau, J.: How long does it take for a bitcoin transaction to be confirmed? <https://coincenter.org/2015/11/what-does-it-mean-for-a-bitcoin-transaction-to-be-confirmed/> (November 2015)
- [6] Croman, K., et al.: On Scaling Decentralized Blockchains . In: Workshop on Bitcoin and Blockchain Research (Feb 2016)
- [7] Douceur, J.: The Sybil Attack. In: Proc. Intl Wkshp on Peer-to-Peer Systems (IPTPS) (Mar 2002)
- [8] Ethereum Homestead Documentation. <http://ethdocs.org/en/latest/>
- [9] Feller, W.: An Introduction to Probability Theory And its Applications, vol. 1. Wiley (1957)
- [10] Fischer, M., Lynch, N., Paterson, M.: Impossibility of distributed consensus with one faulty process. JACM 32(2), 374–382 (1985)
- [11] Heilman, E., Alshenibr, L., Baldimtsi, F., Scafuro, A., Goldberg, S.: Tumblebit: An untrusted bitcoin-compatible anonymous payment hub. Cryptology ePrint Archive, Report 2016/575 (2016), <http://eprint.iacr.org/2016/575>
- [12] Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse Attacks on Bitcoin’s Peer-to-peer Network. In: USENIX Security (2015)
- [13] Huang, D., Dharmdasani, H., Meiklejohn, S., Dave, V., Grier, C., McCoy, D., Savage, S., Weaver, N., Snoeren, A., Levchenko, K.: Botcoin: Monetizing stolen cycles. In: NDSS (2014)
- [14] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G., Savage, S.: A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In: Proc. ACM IMC. pp. 127–140 (2013), <http://doi.acm.org/10.1145/2504730.2504747>
- [15] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf> (May 2009)
- [16] Pagnia, H., Vogt, H., Gaertner, F.: Fair Exchange. The Computer Journal, vol. 46, num. 1, p. 55, 2003. 46(1), 55–78 (2003)
- [17] Poon, J., Dryja, T.: The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. <http://www.lightning.network/lightning-network-paper.pdf> (November 2015)
- [18] Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Proc. Financial Crypto. pp. 6–24 (Apr 2013), http://doi.org/10.1007/978-3-642-39884-1_2
- [19] Rosenfeld, M.: Analysis of hashrate-based double-spending. <https://bitcoil.co.il/Doublespend.pdf> (December 2012)

- [20] Sasson, E.B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: IEEE S&P. pp. 459–474 (2014), <http://dx.doi.org/10.1109/SP.2014.36>
- [21] Sompolinsky, Y., Zohar, A.: Bitcoin’s Security Model Revisited. <https://arxiv.org/abs/1605.09193> (May 2016)
- [22] Tschorsch, F., Scheuermann, B.: Bitcoin and beyond: A technical survey on decentralized digital currencies. IEEE Communications Surveys Tutorials PP(99), 1–1 (2016)