Why buy when you can rent?
Bribery attacks on Bitcoin-style consensus

Joseph Bonneau

Stanford University & Electronic Frontier Foundation

Abstract. The Bitcoin cryptocurrency introduced a novel distributed consensus mechanism relying on economic incentives. While a coalition controlling a majority of computational power may undermine the system, for example by double-spending funds, it is often assumed it would be incentivized not to attack to protect its long-term stake in the health of the currency. We show how an attacker might purchase mining power (perhaps at a cost premium) for a short duration via bribery. Indeed, bribery can even be performed in-band with the system itself enforcing the bribe. A bribing attacker would not have the same concerns about the long-term health of the system, as their majority control is inherently short-lived. New modeling assumptions are needed to explain why such attacks have not been observed in practice. The need for all miners to avoid short-term profits by accepting bribes further suggests a potential tragedy of the commons which has not yet been analyzed.

1 Introduction

Bitcoin [6], launched as a cryptocurrency in 2009, has rocketed to popularity with a monetary base nominally worth over US$6 billion at the time of this writing. Any cryptocurrency must prevent double-spending. Bitcoin relies on a public, distributed ledger called the blockchain which logs all transactions to ensure that funds may only be spent once. Bitcoin uses a computational puzzle system (often called "proof-of-work"[1]) to maintain consensus on this ledger and continually add new *blocks* of transactions.

The scheme is frequently claimed to be *incentive-compatible* in that stability is maintained assuming miners behave "rationally", though this was not formally defined (let alone proved) in the system's original design [6] and does not have a consistently agreed-upon definition [1]. A key assumption, dating to Nakamoto's original white paper [6], is that any party controlling a majority of mining capacity is likely to maintain significant capacity and hence has a large expected future revenue stream. The risk of compromising this earning potential is believed to discourage any attacks which may harm Bitcoin's exchange rate. Our contribution is to show that this assumption might fail in the case that a miner *temporarily* obtains a majority of mining power through bribery. Such a miner would know this majority to be fleeting and hence would not have future earnings to protect. There are plausible assumptions under which this attack is still not feasible or at least not lucrative, but they are much stronger than those used thus far to argue that Bitcoin is incentive compatible.

[1] Bitcoin's mining puzzle is not a strict *proof-of-work* scheme but a probabilistic one.

2 Renting mining capacity

There are multiple ways in which an attacker might obtain a *temporary majority* of mining capacity not through the traditional route of buying and owning mining power, but by renting this capacity from the nominal owners. We will discuss three such scenarios in turn, some are known in Bitcoin folklore but none has been explicitly discussed in formal Bitcoin research. Note that in every scenario, the attacker will have to pay some premium ϵ to rent mining capacity; the attacker would expect to recoup this through double-spending profits.

2.1 Out-of-band payment

The simplest mechanism is to directly pay the owners of mining capacity to work on blocks of the attacker's choosing. This payment may be in bitcoins or any outside (state) currency. Multiple online "cloud mining exchange" services have arisen in the past year which allow exactly that, including `cex.io`, `pow88.com`, and `bitfinex.com`. Relatively little has been published on the extent or efficacy of such mining exchange services, although they typically charge a premium of up to $\epsilon = 3\%$ over the expected earning capacity of rented mining power.

The downside of this arrangement is it lacks enforcement: a miner can accept payment and then mine independently for its own benefit. Both sides need to trust each other or a third-party exchange to enforce their agreement. Because of the lack of built-in trust, it is also difficult for the attacker to bribe anonymously.

2.2 Negative-fee mining pool

A second approach is to establish a mining pool paying an above-market return. Mining pools exist to allow miners to share risk. Participants try to find blocks paying rewards to the pool manager, who then disburses the profits amongst members. Accounting is done by reporting *shares* or near-blocks. For example, if the current probability of finding a Bitcoin block is 2^{-d} (that is, the block's hash must begin with at least d zero bits), participants will report any blocks found with a hash starting with $s < d$ zero bits, drastically lowering the variance in earnings by the participants as many more shares will be found than blocks.

Popular mining pools now offer a "0% fee" meaning that participants earn as much on expectation as they would by mining solo. That is, for a block reward is B miners in a 0%-fee pool will earn $B \cdot 2^{s-d}$ per share. There is no technical reason why an attacker can't start a pool offering a *negative fee*, that is, $(1+\epsilon)B \cdot 2^{s-d}$ per share reported. Because such a pool would lose money on expectation, no honest pool should be able to match this reward. The larger the negative fee, the greater the interest such a pool should attract.

This setup has the advantage for the attacker of reducing trust-the accounting mechanism ensures they will only pay for legitimate mining work.[2] Alert

[2] An issue remains that pool participants could report shares but withhold valid blocks. This is an issue for all mining pools and has been analyzed in the context of attacks between mining pools [3,2,4], however it is not profitable for individuals.

miners would still have to trust the attacker to pay. However, this trust can be incrementally established as the attacker pays for valid shares, making the setup relatively low-risk for miners. Miners would of course know they were joining an attack pool attempting to double-spend which could harm them via an exchange rate crash, though as we will discuss this would require coordinated action by the miners to ensure no miners are tempted to defect and profit from the attack.

An open question is how "sticky" miner preferences are or how quickly they would move in practice to a pool offering a better return.

2.3 In-band payment via forking

Finally, an attacker could attempt to bribe through Bitcoin itself by creating a fork containing bribe money freely available to any miners adopting the fork. Such an attacker would begin with a large pool of funds in address K_0 as of block B_{i-1}. The attacker would then broadcast a transaction moving all of these funds to address K_1 and wait for it to be included in block B_i. The attacker would then try to introduce a fork[3] by finding an alternate block B'_i (possibly using another bribery method), in which they would include a transaction moving the funds from K_0 into another address $K'_1 \neq K_1$. Note that this transaction would conflict with the transaction in block B_i moving the same funds to K_0.

Once this fork occurs, the attacker broadcasts a transaction sending the funds from K'_1 to a series of m addresses $K_2^1, \ldots, K_2^m$. Each address K_m^j is a script enabling anybody to claim the funds as of block[4] $i + j$, ensuring that miner finding the j^{th} block in the fork can claim the funds in address K_m^j.

The attacker's fork of the blockchain now contains freely available bribe money as desired, incentivizing miners to forgo mining on the current longest branch in exchange for potentially higher rewards. There are several variants of this attack, for example simply broadcasting a stream of time-locked transactions paying a high fee on the attacker's branch, but this version is probably best as it commits the attacker to a fixed sequence of bribes in advance.

Note that if the attacker's fork never overtakes the main branch, this bribe money will not be valid and the miners will be left with nothing. Put another way, the attacker only pays if the attack succeeds. Thus, this method inherently transfers risk from the attacker to the miners accepting bribes.

In practice, most miners today run default node software which would ignore any such attack branch completely. Even if all miners were able to spot the attempted branch and detect the additional available bribe money, they would still be taking a risk by participating in the attack. Unlike the mining pool approach or direct payment, participating miners would not be paid if the attack fails. The attacker could try to accommodate this by making a larger proportion

[3] If the attacker's attempt to introduce a fork fails and another block is found on the main chain, they can move the funds from address K_1 again. By cycling these funds every block they can ensure their fork is arbitrarily close to the longest chain.

[4] This script would be achieved using a single `OP_CHECK_LOCK_TIME_VERIFY` command, which has been standard in Bitcoin since mid-2015.

of the bribery money available in earlier blocks when it is less clear the attack will succeed. Still, it remains unclear how much of a risk premium the attacker would have to pay with this method to attract significant interest.

3 Bribery attacks

Given the above methods for renting mining capacity, we can assume our attacker is able to rent an arbitrary amount of capacity at a cost of $\approx \epsilon \cdot B$ per block mined, where B is the mining reward for one block. Note that ϵ might vary based on the attack method and how deep the attempted fork is.

Given this capability, a bribery attack is straightforward: the attacker publishes a transaction T in block B_i, waits until k follow-up blocks have been published so that some irreversible action is taken as a result of T, introduces a new block B_i' with a conflicting transaction T', and then rents sufficient capacity (at least a majority of the network) to extend the branch containing B_i' until it becomes the longest branch. The attacker has double-spent the funds in transactions T and can potentially earn a profit equal to the entire value of T.

In a very simple model, such an attack would offer profits bound only by the quantity of currency in circulation. Assuming there is no inherent limit on the size of transactions or special security restrictions for large transactions, the size of T is unbounded. The attacker's cost is $k \cdot \epsilon \cdot B$, but with perfectly rational miners ϵ should trend towards zero as accepting any bribe would be more profitable for miners than mining directly. Therefore, in the simplest model the attacker's benefits could be unbounded and costs would a small constant, making the attack infinitely profitable.

3.1 Counter-bribing by miners

In the simple model above, there is no inherent lower limit to the amount the attacker must pay. If miners detect that this attack is occurring, however, miners who have already mined (and tentatively received mining rewards) for the current longest branch would be incentivized to oppose the attacker by *counter-bribing* to encourage miners to continue building on the current longest chain to ensure their mining rewards don't disappear.

If the attacker is attempting to institute a k-block fork, this would mean some miners are poised to lose (at least) $k \cdot B$ if the attack succeeds. They might be willing to spend nearly all of this money to oppose the attacker, as it would disappear if the attack succeeds. In this scenario, the attacker would need to pay at least $k \cdot B$ in bribes (instead of $k \cdot \epsilon \cdot B$ in the case of no counter-bribing). The attack may still be infinitely profitable as long as the amount T which the attacker stands to gain is unbounded while mining rewards are capped.

Limiting the attack requires offering larger mining rewards to ensure a high-incentive for counter-bribing, but this is likely impractical. Preventing the attack would require that the block reward B for each block was at least V, where V

is the total amount transacted in each block (all of which could be funds the attacker is attempting to double spend). This would effectively mean a transaction fee rate of 50% (paid through inflation), making the currency impractical.

4 Analysis of mitigating factors

Despite the apparently lucrative opportunity to perform a bribery attack, there is no evidence that this has ever been seriously attempted. We rule out explanations based on "good will" or lack of motivation given the track record of significant thefts of Bitcoin in practice [5]. We instead consider a number of factors which may hinder this attack in practice, which we will outline in rough order from least to most plausible. None of these explanations is completely satisfactory and all represent stronger assumptions than have previous been made when arguing that Bitcoin-style consensus is incentive-compatible.

4.1 Miners may be too simplistic to recognize or accept bribes

Today, it might not be possible to rent any significant mining capacity through bribes as a potentially large portion of miners are not technically capable of running any algorithm besides the default. They may be unwilling or unable to change pools even at the promise of higher fees, unable to rent their capacity on a mining exchange, or unable to detect in-band bribes. This mitigation goes against the very notion of incentive compatibility, which ensures the system is stable assuming miners behave rationally. Furthermore, as miners become more professional and technically capable this is likely to be less true in practice.

4.2 The attack requires significant capital and risk-tolerance

Profiting from the attack requires creating a very large transaction T. The attacker needs this capital available up front and, while the attacker won't necessarily lose the value of T if the attack fails, the bribes may not be recovered if the attack fails.[5] While this may be a practical limitation for many attackers, it appears to be a poor assumption to build into a mathematical model of Bitcoin.

4.3 Profit from double-spends may not be frictionless or boundless

Our analysis assumed the attacker could turn the opportunity to double-spend into "pure" profit of an unlimited amount. Double-spending in Bitcoin doesn't actually create additional currency, it simply gives an attacker the opportunity to temporarily deceive some other party into believing they have received funds which will later be taken back. Profiting from this capability requires a counterparty the attacker can swindle that will immediately (after k blocks of

[5] As mentioned in Section 2.3, bribers placed in band will not be at risk if the attack fails, though this method may be the most difficult to execute.

confirmation) transfer something of equal value to the attacker. In some scenarios (e.g. exchanges, mixing services), this might be an equal value of Bitcoin. In other cases, it might be physical goods whose shipment may be reversed.

Either way, in practice the attacker might not be able to double-spend without paying transaction fees to the counterparty, or may not be able to double-spend a sufficient amount to make the relative cost of bribes negligible. This seems a poor mitigation as it is relatively fragile and difficult to analyze. In any case, it probably only adds a small constant amount of overhead to the attack.

More practically, infinitely-sized double spends are of course not possible. Bounds exist both due to the limited amount of Bitcoin currency in existence and the amount that victims are willing to exchange. Thus, the profit potential is not infinite, although this is also an inadequate mitigation as in practice it is likely that profits from a double spend will be orders of magnitude higher than mining rewards (and hence the volume of bribes required).

4.4 Extra confirmations for large transactions

Recipients may require more confirmations for larger transactions. This makes the attack more difficult because as the number of blocks in the attempted fork k increases, the attacker's bribery costs increase linearly. Unfortunately, the attack may make many smaller transactions simultaneously and attempt to double-spend all of them. Thus it appears impractical for this approach to have much impact. Furthermore it would require the confirmation time would need to grow linearly with the value of the transaction.

4.5 Counter-bribing by the intended victim

In addition to counter-bribing by miners, the attacker's victim may be willing to counter-bribe to prevent the attack. Note that the attacker's profit is completely derived from the losses incurred by one or more specific parties. Assuming they detect the attack, they may be willing to spend significant money to fight back.

In general, any party receiving funds on the main chain but not on the attacker's branch may counter-bribe, but the attack can easily neutralize all non-targeted recipients by including their transactions on the attack branch as well. Therefore we only need to consider counter-bribing by the intended victim.

In the limit, they should be willing to spend up to the entire value of transaction T in counter-bribes, because if the attack succeeds they will lose this entire value. The attacker would then have to spend this same amount in bribes (plus ϵ), making the attack unprofitable.

This mitigation is undesirable as it significantly changes the security model of Bitcoin, with all parties receiving funds needing to scan for potential bribery attacks and be prepared to fight them off. It also implies recipients must be willing to effectively spend protection money (which miners would ultimately pocket) to protect their transactions' integrity.

4.6 Miners may refuse to help an attack against Bitcoin

The purpose of a bribery attack would be visible to any miners participating in it. It would also invariably damage the reputation of Bitcoin if successful. This is a very similar argument to the general argument that a 51% attacker would be unwise to actually attack the network in practice: miners should be incentivized against accepting short-term bribery if it damages their long-term earning potential.

While this is the most plausible explanation, this suggests a looming tragedy of the commons, particularly in the case of a negative-fee mining pool. The security and reputation of Bitcoin (which maintain the strength of its exchange rate by attracting users) can be viewed as a *common good* shared by miners. All miners might recognize their long-term shared incentive is to resist joining the attacker's negative-fee pool which might damage Bitcoin's reputation. However, any miners who joined would immediately see their profits rise in this scenario, even if the attack failed, providing a direct incentive for miners to defect by accepting bribes to attack. SMiners generally have the capability to mine anonymously (by using new addresses in the coinbase transaction of any block they find), making it impractical to punish miners who defect and accept bribes without radically changing the protocol. This tragedy of the commons suggests it might be hard for small miners without effective political organization to prevent successful bribery attacks, whereas a monolithic majority miner is protecting its own self-interest by not attacking..

5 Concluding remarks

We have outlined the possibility of a bribery attack on Bitcoin and discussed the potential implications. Bribery is possible in Bitcoin and indeed it can be facilitated in several surprising ways by the Bitcoin protocol, namely negative-fee mining pools and anybody-can-spend transactions. Requiring all miners to avoid short-term profits to protect the long-term health of the system appears to introduce a tragedy of the commons.

We do not claim this is currently a practical attack. Our aim was merely to demonstrate that, assuming this attack is not being observed because it is not practical, any model attempting to show that Bitcoin-style consensus is incentive-compatible must be strong enough to rule out such bribery attacks. From our initial analysis of possible new modeling assumptions, none seem highly desirable. This may put the security of Bitcoin's consensus protocol on weaker footing than previously believed.

References

1. Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W.: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In: 2015 IEEE Symposium on Security and Privacy (May 2015)
2. Courtois, N.T., Bahack, L.: On subversive miner strategies and block withholding attack in bitcoin digital currency. arXiv preprint arXiv:1402.1718 (2014)
3. Eyal, I.: The Miner's Dilemma. In: IEEE Symposium on Security and Privacy (2015)
4. Luu, L., Saha, R., Parameshwaran, I., Saxena, P., Hobor, A.: On power splitting games in distributed computation: The case of bitcoin pooled mining. Tech. rep., Cryptology ePrint Archive, Report 2015/155, 2015, http://eprint. iacr. org (2015)
5. Moore, T., Christin, N.: Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk. Financial Cryptography (2013)
6. Nakamoto, S.: Bitcoin: A peer-to-peer electionic cash system (2008)