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

Joseph Bonneau, Edward W. Felten, Steven Goldfeder, Joshua A. Kroll, and Arvind Narayanan

Princeton University

Abstract. The Bitcoin cryptocurrency system relies on a novel distributed consensus mechanism relying on economic incentives. It is often argued that Bitcoin is "incentive-compatible" in simplified models; that is, that the scenario in which all miners follow the Bitcoin protocol is a stable Nash Equilibrium in which no miner has any incentive to defect. We introduce the notion of a bribery attack in which an attacker can purchase mining power (perhaps at a cost premium) for a short duration, using it to profit by double-spending. Explaining the lack of such attacks in practice requires significant additional modeling assumptions, demonstrating the inadequacy of current models of Bitcoin consensus.

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 solve the *double-spending* problem of efficiently detecting if an entity has attempted to spend the same funds twice. Bitcoin addresses this with a public, distributed ledger called the *block chain* that logs all transactions and is designed to ensure that only one transaction redeeming specific funds can ever be logged. Bitcoin uses a computational puzzle system (often somewhat misleadingly called a "proof-of-work" system¹) 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 paper [6] and does not appear to have any agreed-upon definition. We will introduce a working definition of what this means, which has been argued by Nakamoto [6] and others to hold in practice even in the face of an attacker controlling a majority of mining power.

The key assumption behind this argument is that any party with a majority mining capacity at a given time is likely to maintain that capacity for a long time into the future and hence has a large expected future earnings. The risk of compromising this earning potential provides the economic incentive which

¹ Note that Bitcoin's mining puzzle is not a strict *proof-of-work* scheme but a probabilistic proof-of-work scheme. Miners can only analyze which version of the block chain was "most computationally challenging" to produce on expectation.

is believed to discourage possible attacks which may harm Bitcoin's exchange rate. The key contribution of our paper, beyond identifying this assumption, is analyzing the scenario in which it might fail, specifically the case that a miner *temporarily* obtains a majority of mining power by bribery. Such a miner would know this majority would be fleeting and hence would not have a future earning potential 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.

2 Modeling Bitcoin consensus

The goal of the Bitcoin consensus protocol (sometimes called "Nakamoto consensus") is to achieve eventual consensus on a block chain of transactions in a decentralized manner. Perhaps the most novel aspect of Bitcoin is that this is achieved without any designated trusted authorities, or indeed any named parties at all. Participants in the consensus scheme, called *miners*, are completely anonymous (though they may pseudonymously link their blocks by using the same address to receive their mining rewards).

In essence, voting power over the block chain is apportioned by computational power. Finding a valid block is computationally difficult and all participants in the system agree to accept the version of the block chain which was (on expectation) the most computationally challenging to produce. This is called the *longest chain rule* although strictly speaking "length" is defined as the aggregate expected amount of computation and not the total number of blocks.

2.1 Block chain integrity & stability

Integrity requires that all transactions in the block chain are valid, that is, they are well-formed according to the rules of Bitcoin, redeem other valid transactions, and are not double-spends. This is enforced in a straightforward (albeit strict) manner: any entity running a Bitcoin node will permanently reject any version of the block chain containing even a single transaction which it considers invalid, regardless of the length of that block chain. If two distinct groups of miners disagree on the validity of a transaction in some block, this may lead them to permanently accept two different block chains as valid. This may be caused, for example, if the rules of the system are updated and some miners continue to follow the old rules. This situation is called a *hard fork* and significant care must be exercised with miner software to prevent this.²

Block chain stability further requires that consensus is reached somewhat rapidly. Even if all miners agree on the rules for valid transactions, the block chain can suffer a *temporary fork* if different groups of miners temporarily accept

 $^{^2}$ This has occurred exactly once by an accidental software update in March 2013 [1], when a change in acceptable block size caused an unintentional hard fork between clients running version 0.7 and 0.8 of the software. The situation was resolved by an emergency software patch after a 24-block fork.

different (both valid) versions of the block chain. This can occur naturally due to network latency. If two conflicting blocks are discovered at around the same, different populations of miners will (by default) accept the block that they heard first. This situation should resolve quickly as miners working on one branch of the block chain will find a second block first, at which point miners working on the other branch will switch to the longer branch, leaving behind an *orphaned branch* (or an orphaned block if it is only one block long).

Temporary forks are dangerous in that they allow double-spending: if Alice sends funds to Bob in a transaction included in a temporary branch of the block chain which Bob accepts, Alice will be free to send these funds to somebody else if another branch eventually grows longer but doesn't include her transfer to Bob. To defend against this, Bitcoin users typically won't accept a transfer of funds until it is *confirmed* by some number of follow-up blocks (commonly 6 blocks in practice). We define the *length* of the fork as the greatest depth of a temporarily accepted branch which is later orphaned from its closest ancestor block on the eventual consensus branch. A fork of length 6 or more is highly dangerous as this may allow double-spending under the default 6-block confirmation rule.

We are now ready to define a stability property for the Bitcoin block chain:

Definition 2.1. Block chain stability: Block chain stability holds if the probability of a k-block temporary fork is a negligible function of k, that is, $p \in O(2^{-k})$.

Stability holds if all miners follow the default protocol and the communication network remains connected. A temporarily partitioned network clearly negates stability because separate partitions will introduce forks of unbounded length.

Note however that even if all miners follow the default protocol and the network remains connected, we cannot guarantee that temporary forks of length longer than some k_{max} never occur. At any point two blocks may be found near-simultaneously with low probability in which case miners will accept whichever block they first heard on the network. This situation can persist indefinitely if the two groups of miners continue to find blocks near-simultaneously and extent their version of the block chain. However, assuming that latency on the Bitcoin peer-to-peer network is relatively low compared to the expected time to find a block (close to 10 minutes), the probability of such block collisions is low and the probability that they will repeatedly occur indeed shrinks exponentially.

2.2 Basic incentive compatibility

Given this definition, we can now define incentive compatibility:

Definition 2.2. Incentive compatibility (weak): A block chain consensus protocol is weakly incentive-compatible if, assuming the majority of mining power is controlled by agents whose only incentive is to maximize their mining rewards and no single entity controls a majority of mining power, there is a Nash equilibrium in which all miners follow the default protocol.

This is a (rough) paraphrasing of the property claimed by Nakamoto in the original Bitcoin proposal [6]. Nakamoto justified this by analyzing an attacker

with a proportion $q < \frac{1}{2}$ of the total mining power attempting to create a longer chain than the rest of the network, which is a Gambler's Ruin problem. In expectation, the attacker will eventually lose the "mining race" and hence receive no rewards for their work, whereas by mining honestly they will receive an expected proportion q of the rewards.

Therefore, Nakamoto's argument goes, a self-interested attacker will profit more from honest behavior and hence all miners should follow the default "longest chain" protocol, provided that a majority of other miners are following the same strategy. Without defining it as such, he argued that a Nash equilibrium exists in which all miners follow the longest-chain rule. In this equilibrium, block chain stability will hold.

However, his argument does not rule out the possibility that there are other equilibria in which stability does not hold. Kroll, Davey, and Felten posit that there are equilibria in which miners do not always extend the longest chain[4].

2.3 Strong incentive compatibility (security against a 51% attacker)

Nakamoto further claimed [6] that even an attacker with a majority of mining power (often called a 51% attacker) would be incentivized to play honestly. We can define this as a stronger incentive-compatibility property:

Definition 2.3. Incentive compatibility (strong): A consensus protocol is strongly incentive-compatible if, assuming the majority of mining power is controlled by agents whose only incentive is to maximize their mining rewards, there is a Nash equilibrium in which all miners follow the default protocol.

The only difference between this definition and the weaker Definition 2.2 is that we've dropped the requirement that no single entity controls a majority of mining power. Note that we still must assume that mining power is controlled by agents interested in maximizing mining rewards. It is accepted that if an agent acquires the majority of mining power and has incentives beyond simply maximizing their profits in Bitcoin, they may subvert the system. For example, a powerful agent with a majority of mining power that is incentivized to destroy public confidence in Bitcoin can easily do so. This has been dubbed a *Goldfinger attack* [4].

Naively this property should not hold even for economically motivated miners, as they would be incentivized to introduce arbitrarily long forks to doublespend coins and profit. They could also profit without double-spending simply by refusing to build off of any other miner's blocks, ensuring they receive 100% of mining rewards. We can call these two attacks *double-spending* and *exclusive mining*, respectively.

There is a further *selfish mining* attack [3] which may even be profitable with significantly less than a majority of mining power depending on assumptions about the communication network. In this attack the miner withholds a number of blocks temporarily, forcing other miners to work on blocks which will immediately be orphaned when the withheld blocks are announced. The basic argument put forward by Nakamoto against all of these attacks, which has been expanded in subsequent analysis [2,4], is that these attacks would be highly visible and hence likely to cause major damage to the exchange rate of Bitcoin with other currencies. Nakamoto argues that a rational 51% miner would be foolish to attempt either attack, because they would risk damaging their long-term potential to earn mining in exchange for short-term profit.

This argument inherently requires reasoning about the exchange rate value of bitcoins, and not simply the attacker's ability to gain nominal bitcoins within the system. Hence this is no longer a well-defined game theoretic problem as it inherently depends on human judgement and confidence which would influence the exchange rate.

It is not known whether Nakamoto's argument for strong incentive compatibility is correct, nor whether strong incentive compatibility might hold for other reasons. However, neither exclusive mining, selfish mining nor double-spending by forking have been observed in practice, despite the existence of large mining pools including an extended period in July 2014 in which the GHash.IO mining pool exceeded 50% of total mining capacity.

3 Renting mining capacity

The key assumption embedded in the argument that Bitcoin is strongly incentivecompatible (Section 2.3) is that any party with a 51% network stake at a given time is likely to have significant share of the mining power for a long time into the future and hence a large expected future earning potential to protect. We now turn our attention to an attacker with 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. There are multiple ways in which this might be realized in practice which we will discuss in turn. Note that in every scenario, the attacker will have to pay some "premium" ϵ to rent mining capacity.

3.1 Out-of-band payment

The simplest mechanism is to simply pay the owners of mining capacity directly. This payment may be in an outside (state) currency or in bitcoins. 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 charge a premium of up to 3% over the expected earning capacity of the mining hardware if operated independently.

The downside is that this type of arrangement has no enforcement: a miner can accept payment and then mine independently for its own benefit. Both sides need to trust the other in the exchange. Because of the lack of built-in trust, it is also difficult for the attacker to perform while keeping their identity secret.

3.2 Negative-fee mining pool

A second approach is to rent capacity by establishing a mining pool and paying an above-market return. Popular mining pools now offer a "0% fee" meaning that participants earn on expectation the exact same quantity of mining rewards as they would mining as individuals (with participation in the pool having the effect of drastically lowering the variance). There is no fundamental reason why an attacker can't start a pool offering to pay a *negative fee*, that is, to pay a higher expected value to pool participants than they would earn on their own.

For example, if the block reward is B and 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), the best terms an honest pool operator can provide are pay-per-share of $B \cdot 2^{s-d}$ for any block starting with at least s zero bits (called a *share*).

An attacker's pool might offer pay-per-share of $(1 + \epsilon)B \cdot 2^{s-d}$. Such a pool would lose money on expectation. However, an attacker would likely need to pay a premium to attract significant interest in the competitive market of mining pools. The attacker would expect to recoup this cost through double-spending profits.

This setup has the advantage for the attacker of removing trust-they would only be paying for legitimate mining work.³ 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 practical question is how "sticky" miners are to the pools they typically use or how quickly they would flock to a pool offering a better return.

3.3 In-band payment via forking

Finally, an attacker could attempt to rent capacity through Bitcoin itself by creating a fork with readily available money. The attack would proceed as follows: the attacker would begin with a large pool of funds in address K_0 in block B_i . The attacker would then publish a transfer moving all of these funds to address K_1 and have other miners include it in block B_{i+1} . Now, the attacker would need to introduce a fork by finding an alternate block B'_{i+1} . The miner could find this block through their own mining or using another bribery method.

Crucially, the attacker would include a transaction in block B'_{i+1} dividing the funds from K_0 into a series of m addresses K_1, \ldots, K_m . Note that this transaction would conflict (be a double-spend of) the transaction included in the block B_{i+1} found naturally by the miners. Now, the attacker will publish a series

³ An issue remains that pool participants could neglect to report valid blocks if they find them. This is an issue for all mining pools. However, there is little incentive for miners to do so and this can be addressed by paying an extra bonus for valid blocks.

of time-locked transactions transferring the funds in each address K_1, \ldots, K_m to an "anybody-can-spend" address.⁴ Miners should always ensure to claim any available "anybody-can-spend" funds in any block they mine. Thus, the attacker has now essentially put some amount of "bribe money" onto the branch of the block chain they have introduced, with the time-locks ensuring that the next m blocks on this branch will each get a share if the attacker's branch eventually overtakes the longer branch.

In this way, miners can be incentivized to go against the current longest branch in exchange for potentially higher rewards. The advantage of this approach is that no trust is required by either party—miners joining the attacker's branch will receive guaranteed rewards if the branch succeeds, while the attacker will only pay if the attack succeeds.

In practice, most miners today run default node software and hence 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, miners would not be paid if the attack fails. The attacker could try to accommodate this by making a larger proportion 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 in this scenario.

4 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 $\epsilon \cdot B$ per block mined, where B is the mining reward for one block. The cost ϵ would cover the additional fees paid for directly rented capacity, the negative fees paid to quickly grow a mining pool, or the bribe money made available in the attacker's branch.

With this capability, a bribery attack is straightforward: the attacker has a transaction T included in block B_i , waits until k follow-up blocks have been published, introduces a new block B'_i with a conflicting transaction T', and then rents sufficient capacity (at least a majority of the network) to build on the branch with B'_i until it becomes the longest branch. The attacker profits by having double-spent the funds in transactions T and T' and can potentially earn a profit of the entire value of T.

In a completely frictionless model, such an attack would be massively lucrative. 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 infinite and costs would a small constant, making the attack infinitely profitable.

⁴ Technically, Bitcoin addresses are really scripts specifying the conditions for redeeming funds. A script can be written which anybody can claim.

4.1 Counter-bribing by miners

In the frictionless 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 *counterbribing* 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 should rationally 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 counterbribing). Asymptotically speaking, the attack may still be infinitely profitable as long as the amount T which the attacker stands to gain is still unbounded while mining rewards are capped.

There is a more complex case if multiple attackers are attempting to bribe at once, although for simplicity we assume only a single attacker is active.

Limiting the attack requires offering larger mining rewards to ensure a highincentive 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 Vis 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%, making the system impractical for payments.

5 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 in practice. We rule out any explanation based on "good will" or lack of motivation given the track record of significant thefts of Bitcoin in practice [5]. We can consider a number of factors though that may make this attack difficult and risky to profit from in practice, which we will outline in rough order from least to most practical. We argue that some such limiting factor must exist the Bitcoin protocol as practically realized, although none of these explanations is completely satisfactory as they all represent stronger assumptions than have previous been assumed in arguing the Bitcoin consensus is actually incentive-compatible.

5.1 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, as the goal is for the system to be stable assuming miners behave rationally. Furthermore, in practice as miners become more professional and technically capable this is likely to be less true in practice.

5.2 The attack requires significant capital and risk-tolerance

Profiting from the attack clearly requires inserting a very large transaction T into the block chain. The attacker needs this capital available up front and, while the attacker won't lose the value of T if the attack fails, the potentially large value of bribes needed for an attack 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.

5.3 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 confirmation) transfer something of equal value to the attacker. In some scenarios involving exchanges or mixing services, this might actually be an equal value of Bitcoin. In other cases, it might be physical goods.

Either way, in practice the attacker might not be able to double-spend without paying some transaction fee to the counterparty, or may not be able to double-spend an unlimited amount to make the relative cost of bribes negligible. This seems like 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 profit from a double spend which is orders of magnitude higher than mining rewards (and hence the volume of bribes required).

5.4 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,

⁵ As mentioned in Section 3.3, bribers placed in band will not be at risk if the attack fails, though this method may be the most difficult to execute.

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. Even it were possible, it would require the confirmation time would need to grow linearly with the value of the transaction.

5.5 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 wish to counter-bribe, but the attack can easily neutralize all non-targeted recipients by including their transactions on the attack branch as well. Therefore we can only need to consider counter-bribing by the intended victim. This may of course be a group, but for simplicity we can consider a single 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 spend the equivalent of protection money to protect their

5.6 Miners may refuse to help an attack against Bitcoin

The purpose of a potential bribery attack would likely be clear to any miners participating in it. It would also invariably damage the reputation of Bitcoin if successful. This is a very similar argument to that discussed in Section 2.3 that a 51% attacker would be unwise to actually attack the network in practice: miners with a long time-horizon should be incentivized against accepting short-term bribery if it damages their long-term earning potential.

There are several problems with this assumption. First, it requires that the parties being bribed have a stake in Bitcoin. It's possible the attacker could simply be renting general-purpose computational capacity, although at present this is technically infeasible as efficient mining requires dedicated hardware.

An attacker might rent Bitcoin capacity to attack a smaller SHA256-based altcoin (presumably one which does not use merged mining) and the Bitcoin miners would have little incentive to not damage this currency. Indeed, such attacks have been observed multiple times in practice and as a result no SHA256based altcoin has survived without accepting merged-mining from Bitcoin.

Second, even if correct, this argument would mean that even proving *weak* incentive compatibility (stability in the absence of a 51% attacker) requires assuming miners have long-term stake in the value of the currency system, whereas

without considering bribery this assumption was only considered needed to justify *strong* incentive compatibility (stability in the presence of a 51% attacker).

Finally and most importantly, a tragedy of the commons may arise, particularly in the case of a negative-fee mining pool. All miners might recognize their long-term shared incentive is to resist joining the negative-fee pool. However, any miners who joined would immediately see their profits rise in this scenario, even if the attack failed. Security would thus require all miners to resist this profit to protect a common good (the stability and reputation of Bitcoin) even though individual profits can be made by defecting and accepting bribes. Miners generally have the capability to mine anonymously (by using new addresses in the coinbase transaction of any block they find), making it very difficult to punish miners who defect and accept bribes. Therefore, the argument is much more tenuous for small miners without effective political organization than it is for a single monolithic 51% miner.

6 Concluding remarks

In this work, we have outlined the possibility of a bribery attack on Bitcoin and discussed the potential implications. We do not claim this is currently a practical attack. Our aim was merely to demonstrate that, assuming this attack is not actually practical, coming up with a complete model of Bitcoin in which incentive compatibility could be proven requires adding a sufficient number of assumptions to make a bribery attack impossible. From our initial analysis, none of the available options is highly desirable.

This attack possibility may also collapse the definitions of weak and strong incentive compatibility we presented. This means even in a world without a single majority miner, stability of the block chain requires a model which includes miners concerned about the exchange rate who are willing to resist a bribery attack due to its potential for long-term damage to their earning potential. Requiring all miners to avoid short-term profits to protect the long-term health of the system may further require coordination between miners to avoid a tragedy of the commons problem. Collectively, this may put the security of Bitcoin's consensus protocol on weaker footing than is commonly claimed.

References

- 1. Andresen, G.: BIP 50: March 2013 Chain Fork Post-Mortem
- Barber, S., Boyen, X., Shi, E., Uzun, E.: Bitter to Better—How to Make Bitcoin a Better Currency. In: Financial Cryptography (2012)
- 3. Eyal, I., Sirer, E.G.: Majority is not Enough: Bitcoin Mining is Vulnerable. In: Financial Cryptography (March 2014)
- 4. Kroll, J.A., Davey, I.C., Felten, E.W.: The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. In: WEIS (Jun 2013)
- 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)