Too Big to Cheat: Mining Pools' Incentives to Double Spend in Blockchain Based Cryptocurrencies

Ville Savolainen*†

Jorge Soria Ruiz-Ogarrio‡

May 19, 2020

Abstract

In most blockchain based cryptocurrencies majority of verification power is required for facilitating a successful double spending attack, i.e. using the same funds multiple times. Because possibility to double spend sharply deteriorates trust and value, concentration is traditionally considered to be a significant problem. We model agents' incentives to facilitate double spending attacks under opportunity costs. Contrary to a host of previous literature, our main findings indicate that under meager economic profits large pools have higher incentives to act honestly than outsiders, our results hold for 13 major proof-of-work cryptocurrencies. Intuitively, this stems from the fact that mining pools holding more power in a cryptocurrency have stronger vested interest in it.

Keywords: Blockchain, Cryptocurrencies, Bitcoin

JEL Codes: D43, E42, G29

*We are thankful for the comments from Gonul Colak, Campbell Harvey, Mitri Kitti, Timo Korkeamäki, Michael Reher, Hannu Vartiainen, Ellapulli Vasudevan and participants at Graduate School of Finance Winter Workshop in Finance.

†Hanken School of Economics; ville.savolainen@hanken.fi, +358451365687; corresponding author

‡University of Helsinki, Helsinki Graduate School of Economics; jorge.soria@helsinki.fi

1 Introduction

Since Nakamoto (2008) white paper, decentralization of mining power has been considered a corner stone of functionality and transaction immutability. In addition, the genesis of blockchain based cryptocurrencies is deeply rooted in quixotic enthusiasm for trustless systems and avoidance of monopolies – private or public. Contrary to this ideal, a prominent feature in major cryptocurrencies is that most of the mining of new blocks is conducted by very few large pools (Gencer et al. 2018). Additionally, in Bitcoin, there have been at least three occasions since 2011 where a single pool controlled a majority (see below). Economic intuition reveals that this occurs naturally due to miners' willingness to insure against idiosyncratic risk by joining pools.

Should we, just because pools or a coalition of them can double spend, be concerned? If indeed, centralization was such a problem, one would wonder, why have the major cryptocurrencies gained in popularity and market value since their origin, instead of being already collapsed due to dysfunctionality. Besides, why would a large pool or a coalition of them, even if more capable, do anything nefarious while the whole business model depends on the appeal of a particular cryptocurrency? We consider pools' opportunity costs from double spending and examine conditions for which it is costlier for a pool of any size, than for an outsider, to conduct a double spending attack. Our findings implicate that in the 13 largest proof-of-work cryptocurrencies pools have higher costs for conducting a double spending attack than outsiders.

Post the 2016 DAO (Decentralize Autonomous Organization) hack, in which $ 50 million was stolen, most miners opted to fork Ethereum blockchain to cancel the hack. Furthermore, in Bitcoin Cash, two large pools jointly conducted a majority attack in May 2019. However, in lieu of double spending they reversed hacker's transaction(s). This could indicate the following: first, that pools can join forces to attack; second, that it might be in pools' interests to protect the cryptocurrency against misconduct – not to attack against it.

How does double spending harm pools? The value of a cryptocurrency is dependent on

a probability that a transaction will stay as a part of the accepted history after the goods, assets of services have been delivered (see e.g. Pagnotta and Buraschi (2018) and Saleh (2019)). If there is a possibility that a seller would be left without a payment, the sellers should increase prices accordingly, or cease to sell goods in that cryptocurrency. These would cause inflation and a decrease in exchange rate against traditional currencies. Consequently, a pool, of which profits depend on a value of the cryptocurrency, would suffer.

We cast our model in an industrial organization setting, resembling Green and Porter (1984). In our model miners choose to hedge against idiosyncratic risk by conducting mining in pools, this is corroborated by observational evidence. Because of entry costs, incumbent pools gain economic profits in a collusive equilibrium. In our model mining pools' discounted future profits are increasing in pool size and hence the opportunity cost for fraudulent behavior is increasing in pool size. We contribute to the previous literature by modelling expected direct and opportunity costs of double spending in continuous and discrete time. And hence are able to derive a threshold for which for each pool cost of double spending is higher than an outsider's cost.

To illustrate what the threshold values indicate we collect confirmation period data from eight large cryptocurrency exchanges for 13 proof-of-work cryptocurrencies (market capitalization $ 16 - 160,000 million). The model's results indicate that for the observed $\sim$ 2% pool fees (see e.g. Cong, He, and Li (2019)) double spending is costlier for pools than for outsiders. This result holds through the 13 cryptocurrencies in our sample for extremely conservative effects of double spending (- 1 %) on cryptocurrency value. For a conservative decrease of 10 % post attack the threshold pool fees in the cryptocurrencies range from 0.247 % to 0.001 % with a median fee of 0.012 %. Therefore, we argue that even though no model can account for the plethora of possible modifications, the threshold values of our model are robust enough to make general claims on pools' incentives not to double spend.

These results and our model's insights have broader implications for cryptocurrency protocols. Large pools' incentives to maintain honest records are generated by their stake in

the protocol, and hence are dynamic. This resembles monopolists incentives to keep an accurate ledger. Whereas, the proof-of-work miners have static incentives to maintain honesty (Abadi and Brunnermeier 2018). Therefore, requiring large rewards. Because of free entry these rewards, in equilibrium, are pure waste, generating tremendous carbon-dioxide emissions (Stoll, Klaaßen, and Gallersdörfer 2019) and e-waste[1].

Therefore, one could wonder: If the cost to double spend generated by stake (dynamic) is higher than the cost generated by mining (static), is the proof-of-work mining purposeful? In proof-of-stake (PoS) protocols, miners naturally have dynamic incentives to maintain honesty. In addition, the outsiders may double spend only by becoming stakeholders. Furthermore, PoS protocols' CO2 emissions and e-waste generation are insignificant in comparison to those of PoW.

Our results also highlight the importance of dynamic incentives in general. We argue that the prevalent threat for any cryptocurrency comes from agents not having vested interests in it. For example, a coalition of miners, large-scale miner or pool in a major cryptocurrency has only its altruism (and possibly the compatibility of its mining gear and a loss of good will) restraining it from attacking a smaller cryptocurrency where it is not currently operating. Therefore, to build a permissionless blockchain protocol one might consider inducing protocol specific fixed costs and barriers of entry for the miners. This should help to make miners long-term investors; and hence, align their interests with those holding and using the cryptocurrency. In addition, because these means could help barring mining power transfers cross-chain, they might alleviate concerns of bribery attacks discussed in e.g. Bonneau (2016) and Judmayer et al. (2019).

In our analysis we concentrate on proof-of-work protocol because it is the most widely used verification protocol. However, the same economic intuition should be applicable to other verification protocols[2]. We choose to limit our analysis to the double spending attack,

[1]https://digiconomist.net/bitcoin-electronic-waste-monitor/

[2]In proof-of-stake protocol distributed consensus is achieved by randomly selecting a creator of the next block from those holding a stake (current wealth and possibly how long the wealth was held). Stake holders may participate in a pool and hence receive rewards more regularly. The model presented should be

because of its relevance for cryptocurrency users (i.e. buyers and sellers exchanging goods in a cryptocurrency)[3]. We focus on monetary incentives and hence limit our scope to exclude other important aspects of decentralization such as geographic concentration, political implications (Kaiser, Jurado, and Ledger 2018), project competition (Van Wirdum 2018), different attacks against smaller miners (Eyal and Sirer 2014), cryptocurrency holders' taste for diversification, inequality aversion etc.

Furthermore, for our purposes, an agnostic view about why pool sizes are heterogeneous and what causes the empirically observed concentration is sufficient. Cong, He, and Li (2019) provide valuable insights on how pool fees are set and how miners allocate their mining power to different pools and hence explaining pool size differences and variations over time.

To our knowledge, our paper is the first to propose conditions for pools to have higher costs for double spending than outsiders. Hence, our work directly relates to Nakamoto (2008) and the proceeding computer science and blockchain-economics literature. We proposes a novel and contradicting perspective about the importance of decentralization.

2 Literature Review

Our paper is closely related to the literature discussing blockchains' resilience against double spending attacks in proof-of-work protocols. Budish (2018), Kroll, Davey, and Felten (2013), and Rosenfeld (2014) discuss double spending attacks as a threat to trust in a blockchain. Kroll, Davey, and Felten (2013) argue that whenever there exists a possibility to double spend, sellers cannot differentiate between honest and fraudulent buyers, and sellers should cease to accept transactions. Hence, leading to a collapse in cryptocurrency's value. Budish (2018) builds a model to address double spending and draws insights regarding miners' incentives, fixed and variable mining costs, and double spending. Rosenfeld (2014) is one of

applicable to a pool operating in a proof-of-stake cryptocurrency.

[3]Other attacks are mainly conducted by miners against pools or large miners or pools against smaller pools or miners, with the exception of sabotage attacks (also called Goldfinger attacks) which aim at destroying the whole cryptocurrency for exogenous reasons (Budish 2018).

the earliest academic works to discuss double spending and motives behind it. Both Budish and Rosenfeld acknowledge that a double spending attack could harm the double spender, if she has a stake in the cryptocurrency. Our model contributes by analysing pools and their costs from double spending.

Abadi and Brunnermeier (2018) discusses large miners' or pools' profits from a double spending attack and derives conditions for which honest miners fork to cancel a malicious transaction, hence leaving the double spender without profits. Instead of modelling profitability of a double spending attack our analysis contributes by focusing on the direct and opportunity costs of a double spending attack.

Chiu and Koeppl (2019a) discuss the trade-off between fast transactions and finality of payments in a proof-of-work protocols. Eyal and Sirer (2014) discuss a different type of incentive compatible attacks, conducted by large miners or pools, where larger entities profit more per hash than smaller entities. Kiayias et al. (2016) present similar results.

Cong, He, and Li (2019) propose that there are economic forces limiting pool growth. Whereas, Böhme et al. (2015), Arnosti and Weinberg (2018), Alsabah and Capponi (2019) and Ferreira, Li, and Nikolowa (2019), among others, argue that proof-of-work mining is susceptible to severe centralization of mining and governance. The importance of our research is highlighted by the fact that different models provide contradicting outcomes about mining centralization and empirical observations demonstrate high (and occasionally extreme) levels of concentration.

Research more loosely related to our work discussing blockchain from the perspective of market structure (or design) of mining activity is extent: Gans and Halaburda (2015); Huberman, Leshno, and Moallemi (2017); Dimitri (2017); Abadi and Brunnermeier (2018); Ma, Gans, and Tourky (2018); Biais, Bisière, et al. (2018); Auer (2019); and Easley, O'Hara, and Basu (2019) among others. Furthermore, Pagnotta and Buraschi (2018) discuss aggregate mining power as a factor increasing trustworthiness of a cryptocurrency and hence its value. Cong, Li, and Wang (2019) provide a dynamic valuation model for cryptocurrencies based

on the economic activities behind tokens. Biais, Bisiere, et al. (2018) present an overlapping generations model to price cryptocurrencies, where agents can move from central-bank issued currencies to cryptocurrencies. Gandal and Halaburda (2016) analyse competition between cryptocurrencies. Cong and He (2019) discuss blockchain and smart contracts in an industrial organization setting. Yermack (2017) studies blockchain from the perspective of corporate governance and provides a detailed introduction to the topic. Among others Lee (2016) and Chiu and Koeppl (2019b) discus securities trading in a blockchain. Saleh (2019) models consensus and double spending attacks in proof-of-stake. Hinzen, John, and Saleh (2019) present a model of network effects and externalities in the adoption of a cryptocurrency. Iyidogan (2019) models delays, block-size limitations and transaction fees to describe the long-run equilibrium in which block-rewards are solely based on transaction fees.

3 Blockchain Institutions and Double Spending

The first implementations of electronic cash[4] required a trusted third party to verify transactions and prevent double spending. In a double spending attack, the very same virtual money is used multiple times and sellers are left without a payment. Because no seller would be willing to deliver goods if the payments were not immutable, these attacks are considered to be a major threat to cryptocurrencies' viability.

Nakamoto (2008) introduced a novel decentralized double-spending preventing transaction verification solution – blockchain technology[5][6]. To cancel a payment after receiving goods, services or assets a fraudulent agent has to tamper with the blockchain by redoing

[4]The idea of cryptographic currencies was -most probably- first proposed by Chaum (1982). During the 90s and through out the expansion of the Internet, the idea of digital money was latent in various fields. Some early commercial attempts to develop cryptographic protocols for e-money were e.g. Beenz, DigiCash, Flooz and Peppercoin (Bonneau et al. 2015; Huang 2003). The milestones in the process include launching e-gold Ltd. in 1996 and PayPal in 1999. E-gold was the initially successful and widely accepted digital currency, however due to multiple legal issues, it was ultimately shut down. Although PayPal was introduced when e-gold was already relatively established in the market, it managed to gain the attention of successful companies, such as Elon Musk's X.com and, latter, eBay. This could explain its fast and durable success.

[5]An idea similar to blockchain had been proposed by Haber and Stornetta (1990).

[6]A glossary of blockchain terminology may be found from Appendix B.

a costly proof-of-work (PoW). To succeed, an attacker has to outpace all of the network's honest agents, which requires controlling a majority of network's computational power. Consequently, concentration of computational power is considered to open a door for double spending attacks[7].

3.1 Blockchain

In a blockchain based currency, ownership is based on a ledger[8] that is updated by collecting transactions into blocks chained in chronological order. This way, a blockchain keeps track of ownership over time. Nodes (i.e. miners) propose blocks including transactions that are not yet registered in the main chain. Each block is linked directly to the previous one, thus creating a chronological chain.

In a centralized consensus mechanism such as banking a valid history of events is kept by a third party (a single node). In a permissionless blockchain valid state of the protocol (e.g. account balances) is agreed upon through a consensus protocol. All nodes store the same history of events in a peer-to-peer network of multiple nodes. The consensus about the valid history is achieved through longest chain rule, this occurs mainly because the rule functions as a focal point (Kroll, Davey, and Felten 2013). To incentivize miners, adding blocks on the main chain is rewarded by the protocol with newly created units of the cryptocurrency and possible transaction fees.

Multiple miners can propose alternative blocks causing forks. In this case the valid block is the one to which miners chain subsequent blocks. To coordinate, the *longest chain rule* is used: when there exists alternative branches of the blockchain, miners will normally work on the fork with most blocks – however, alternative equilibria exists (Biais, Bisière, et al. 2018).

[7]Double spending attacks are also called majority attacks and 51 % attacks.

[8]A ledger is a registry log. The term comes from *legger* (denoting a large bible or breviary) (*Oxford English Dictionary* 2019). Typically a ledger will be a list of credit and debit transactions that keeps record of ownership.

3.2 Mining

The process of hashing the content of blocks is known as *mining*. Miners compete to be the first in forging the next block. Mining, as a process, functions as follows:

1. A node sets a reference to the block to which she is linking the block, typically the latest block in the main chain. This reference is the hash number of the previous block.

2. The node will collect transactions and events into the block, forming the ledger.

3. Finally, the miner will try random numbers i.e. *nonces* to make the digest of the hash function to begin with the desired number of zeroes (i.e. the number defined by a difficulty criterion). As the number of consecutive zeros, required by the difficulty criterion increases, so does the difficulty of finding a nonce that gives an adequate block number. This makes the process time and energy consuming, and thus, expensive.

Generating new blocks (*mining*) is made artificially expensive by requiring a block number to meet the difficulty criteria. This enforces miners to solve a computationally complex puzzle that requires significant amounts of processing power. Hence, the name of the consensus mechanism *proof-of-work*.

Because every block must refer to the previous block's unique block number which is a function of the blocks content; changing one block or altering its content detaches it from all the following blocks. This is the feature that makes direct tampering with a blockchain impossible. If an agent would attempt to change the transaction history, she would need to mine a new chain which should become the longest. Therefore, with less than majority of hashing power, it is exponentially difficult to cancel a transaction as the number of subsequent blocks validating it increases. Hence there exists a trade-off between transaction finality and immediacy (Chiu and Koeppl 2019a).

3.3 Pools

Mining rewards are economically significant, e.g. in Bitcoin around $80,000 every 10 minutes in March 2020. With the increased number of miners in the network, the probability of an individual miner to find a valid block has decreased significantly[9]. Simultaneously values of many cryptocurrencies, such as Bitcoin, saw an unprecedented increase. Due to this development, mining rapidly became a risky activity involving large rewards arriving at sporadic rates.

Having significant risky rewards, a natural development for the cryptocurrency environments was the apparition of mining pools that provide risk sharing. There exist various types of mining pools, depending on their payment and fee schemes and Appendix A presents the main categories of mining pools by fee types. Pools have acquired such popularity that since 2015 between 95 and 100 percent of the hashing (processing) power in Bitcoin has been controlled by pools. The situation is similar in all major cryptocurrencies.

Concentration of verification process to few large pools, especially in the major cryptocurrencies Bitcoin and Ethereum, has been a serious concern especially amongst computer scientists and practitioners[10]. Gencer et al. (2018) present empirical results about the distribution of mining power in Bitcoin and Ethereum networks: During 2015-2017 top four mining pools controlled 51 percent of the mining power in Bitcoin, and in Ethereum the three largest pools alone controlled 63 percent of the mining power on average. A better Byzantine fault tolerance could be acquired by employing 20 desktop computers (ibid.)[11].

Furthermore, in Bitcoin some mining pools have controlled over or nearly 50 percent of

[9]For a single $ 500 mining device a probability of mining a block in Bitcoin is 0.000028 %. Hence, given current mining hardware's average lifespan of 18 months, there is approximately a 2 % probability of mining a block with a single device.

[10]However, here the contrary is argued by a practitioner: https://finance.yahoo.com/news/no-concentration-among-miners-isn-050000938.html

[11]A central topic in decentralized protocols is how to achieve consensus among peers in a network, especially when nodes cannot differentiate trustworthy nodes from fraudulent ones. Blockchain protocols are designed to have a high degree of Byzantine fault tolerance (BFT), meaning that the protocol can remain trustworthy even if multiple nodes fail or send fraudulent information. The academic discussion about the topic began with Pease, Shostak, and Lamport (1980).

hashing power: DeepBit (June 2011), BTC Guild (April 2013) and Ghash.io (July 2014). At the time of writing, the largest mining gear producer Bitmain Technologies Ltd controls AntPool and BTC.com pools which combined have about 30 to 40 percent of Bitcoin's total hashing power. In addition, Bitmain has large proprietary mining operations. These observations are concerning in the light of Nakamoto's original idea (Nakamoto 2008), that implies that the security of a cryptocurrency rests on its decentralization.

Could mining pools collude to attack? Anecdotally, in May 2019 two largest pools in Bitcoin Cash jointly conducted a majority attack to reverse hacker's transactions. This indicates that, some pools do have incentives for not only acting honestly but also helping to maintain trust in the network. In addition, this demonstrates that pools may collude to conduct such attacks.

Probably a more well-known case is the 2017 split of Ethereum to Ethereum Classic and Ethereum. In this case miners coordinated to fork. Forking was motivated by hacking of the DAO (Decentralized Autonomous Organization) in 2016. In DAO hack an unknown attacker found a vulnerability in DAO smart contracts and stole around $ 50 million. Those opposing the fork continued with the chain that contains the hack and named it Ethereum Classic, while those whom wanted to cancel the hack forked to another chain that kept the name Ethereum. Currently the market capitalization of Ethereum is about 40 times higher than that of Ethereum Classic.

Contrary, in 2017, miners decided not to hard fork after $ 300 million worth of ether was frozen due to a buck in digital multi-signature wallets developed by a third party (for more details on these two cases see e.g. Neitz (2019)). In both cases, there was nothing wrong with the Ethereum protocol per se. However, in the DAO hack funds for further development of Ethereum were at stake. This could further illustrate that large miners and pools can have strong incentives to prevent malevolent activity in the network, if it hurts the viability of the cryptocurrency.

From an economist's perspective centralization is not necessarily a problem whenever

agents' increased capability to conduct double spending attacks is countered by increased incentives to maintain honest conduct. Given the concerning levels of mining concentration, we build our model to better understand large pools' incentives for maintaining honest conduct. The contribution of this paper is to demonstrate that concentration is harmless. To our knowledge, this is the first paper to formally present such result.

Our results' external validity relies on the assumption that mining pools make economic profits. Unfortunately, sufficient data about profitability of pools is practically nonexistent. As an exception, Bitmain –the company owning and operating two largest bitcoin mining pools with 30 to 40 percent of the hashing power– reported 36.3 million dollar gross profit from mining pool service for the first six months of 2018 with a gross profit margin of 84 percent in its Proof of Application filing to Hong Kong Security Exchange in 2018. Since 2015 the margins have ranged from 80 to 89 percent (BitMain 2018). In addition, two empirical findings yield support for the existence of economic profits: First, larger pools win mining competitions with larger probability than their relative computational power would suggest (Gencer et al. 2018). Hence, even under constant (or decreasing) scale costs, in Bertrand type competition, the largest pool(s) would generate economic profits due to higher quality. Second, observed pool fees are increasing in pool size (Cong, He, and Li 2019). Cong, He, and Li (ibid.) propose that this is due to the large pools' market power generated by frictions in miners' willingness to reallocate hashing power between pools.

3.4 Double spending attacks

Most blockchain based cryptocurrencies draw their value from the fact that it is very difficult to tamper with transactions in their main chains.

Trying to eliminate a block that contains a certain transaction becomes exponentially difficult as the number of blocks added subsequently increases. However, if a group of miners control over 50% of the hashing power of the network, they would be able to outpace other miners and generate a new longest chain. This generates double spending possibilities.

Moreover, even if the fraudulent party was only a single miner, it is technically possible for her to present herself as multiple small-size independent miners. Other miners are not able to recognize that behind an alternative chain there is a unique user or coalition trying to double spend. Only after some time, it will be evident that the tampered transaction(s) is not to be included in the chain again, thus revealing that some party has successfully committed fraud.

Figure 1 illustrates how a typical double spending attack would be facilitated. The attack could proceed in the following order:

1. The attacker uses her coins to purchase some goods, services or assets ($Block_t$ in Figure 1).

2. The seller observes that the transaction is included in the blockchain. The seller waits for a number of confirmation blocks to be added after the block containing the transaction. In Figure 1, K is the number of confirmations. Goods are released once observing $Block_{t+5}$.

3. Meanwhile, the attacker is privately working on an alternative chain that does not contain the original transaction. In Figure 1, this corresponds to blocks: $Block_t^* - Block_{t+5}^*$.

4. Once the transaction has been confirmed with enough blocks, the seller delivers the purchased goods ($Block_{t+5}$).

5. After receiving the goods, the attacker continues to mine her alternative chain until it is longer than the current main chain. Then immediately broadcasts her alternative chain.

6. Following the longest chain rule, miners start working on the alternative chain ($Block_{t+6}$).

7. The fraudulent chain is accepted as the main ledger. On this new chain the attacker never used the funds, thus she can double spend them.

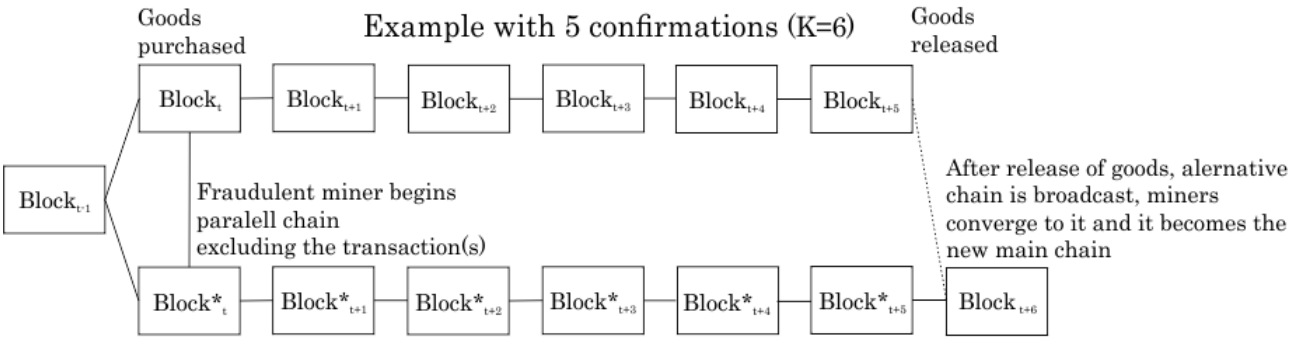

Figure 1: Illustration of a double spending attack.
The number of confirmations (K-1) has been set to 5 in this example.

The attacker knows that at some point other users will notice the fraud. However, as users can have multiple accounts and they are anonymous, it is virtually impossible to know who is fraudulent and who is not. Therefore, post attack both sellers and buyers know that an anonymous user has enough power to tamper with the blockchain and undo transactions. No user can distinguish trustworthy transactions from spurious ones. Hence, one can expect trust in the currency to collapse. Anticipating this, the attacker would spend again her coins until the attack is noticed.

4 Simple Model

To illustrate the intuition of our model let us consider a simplified framework. In the model section we shall build the micro foundations for most of these postulations and attempt to motivate whatever is left as an assumption.

In the model there exists M pools. All the miners participate in pools and the pools compete for a constant reward R. The reward is awarded for the pool winning the mining competition. Each pool m has an exogenously given probability $\frac{H_m}{H}$ of winning the competition, where H_m is the pool m's hashing power and H is the aggregate hashing power. There exists C which is a hashing power unit cost. All pools collect the same fractional fee f before distributing the reward R to its members (i.e. miners). The pools are risk neutral and have

a time discount factor β, corresponding to a subjective discount rate δ (i.e. $\beta = e^{-\delta}$). At each period t a new block is mined and the corresponding reward is awarded for the winner. Therefore the present value of pool m's expected future returns is given by

$$\sum_{t=0}^{\infty} \beta^t \frac{H_m}{H} fR = \frac{fR}{1-\beta} \frac{H_m}{H} \tag{1}$$

We postulate that a following condition holds

$$CH = (1-f)R \tag{2}$$

This condition states that in every period t the aggregate mining cost equals the reward after fees. Intuitively this follows from miners' free entry and their ability to perfectly insure against idiosyncratic risk by joining pools. Later in the model this is shown to be an equilibrium condition in Corollary 6.

To conduct a successful double spending attack an agent would have to outpace all honest miners in the protocol and after receiving the goods broadcast an alternate history with more blocks (i.e. a longer chain) where the original transaction(s) did not occur. Hence, sellers would be left without the payment and the attacker would be able to spend the very same coins multiple times. A seller usually waits for K blocks to be added to the chain starting from the transaction block before delivering the goods. This is to minimize possible attackers' success probability P.

Let us assume that there exists some arbitrary fixed time $T^* < \infty$ during which the K blocks are mined to the main chain. In addition, postulate that the average time between the blocks $(t_{B-1} - t_B)$ approaches zero. From this follows

Lemma 1

$$for \quad \frac{H_a}{H} > \frac{1}{2} \qquad \lim_{t_{B-1} - t_B \to 0} P(T^*) = 1$$

and

$$for \quad \frac{H_a}{H} < \frac{1}{2} \qquad \lim_{t_{B-1}-t_B \to 0} P(T^*) = 0$$

where H_a is the attackers hashing power.

Therefore the direct cost for conducting a double spending attack for a pool m is

$$\int_0^{T^*} max \left\{ \frac{1}{2} - \frac{H_m}{H}, 0 \right\} CH \quad dt \tag{3}$$

here we have abstracted from discount rate as the attacks should usually conclude within few hours.

Let us further assume that after a successful double spending attack the value of the cryptocurrency decreases to zero. In the model section we will discuss the assumption that a value of the cryptocurrency collapses to zero post double spending attack; and in the main results section we will demonstrate that our results hold for value decreases of 1 % and larger throughout all 13 major proof-of-work cryptocurrencies. Therefore the pool m's present value in Equation 1 equals the pool's opportunity cost for conducting an attack, which can be expressed in continuous time limit as

$$\int_0^{\infty} e^{-\delta t} \frac{H_m}{H} fR \quad dt \tag{4}$$

If we combine the direct cost of conducting a double spending and the opportunity cost and use the aggregate mining cost condition (Equation 2) we get the pool m's total cost for conducting a double spending attack

$$\max \left\{ \left(\frac{1}{2} - \frac{H_m}{H} \right), 0 \right\} (1-f) RT^* + \frac{fR}{\delta} \frac{H_m}{H} \tag{5}$$

From Equation 5 we may immediately observe that the cost for conducting a double spending attack is increasing in reward and confirmation time (i.e. the time sellers wait before releasing the goods). Intuitively, the larger the reward the more hashing power is used

in mining on aggregate level for each block which increases the associated costs. Longer waiting time implies greater number of confirmation blocks and hence costlier double spending attacks. In addition, the more patient the pools are (i.e. lower the δ) the higher is their opportunity cost for double spending.

From Equation 5 the Theorem 1 follows

Theorem 1 *for* $\frac{H_m}{H} < \frac{1}{2}$ *while*

$$(1-f)T^* - \frac{f}{\delta} < 0$$

cost of double spending is strictly increasing in $\frac{H_m}{H}$*. For* $\frac{H_m}{H} > \frac{1}{2}$ *the cost is always increasing in* $\frac{H_m}{H}$*.*

Theorem 1 implies that for example a waiting time of 1, 10 and 24 hours (e.g. in Ethereum these correspond to 240, 2400 and 57600 confirmations) and an annual $\delta = 0.1$ the fees for which the cost of double spending is increasing in pool size are 0.001 %, 0.011 %, and 0.027 % respectively. Furthermore, if pool fees are 2 % and confirmation time is 1 hour for pools controlling 1 %, 20 %, 40 % of hashing power attacking is 35, 679, and 1358, times more expensive than for an outsider, respectively.

However, if we assume that only a fraction $\alpha \in (0, 1]$ of the value is lost we modify Theorem 1 which yields

Corollary 2 *for* $\frac{H_m}{H} < \frac{1}{2}$ *while*

$$(1-f)T^* - \alpha\frac{f}{\delta} < 0$$

cost of double spending is strictly increasing in $\frac{H_m}{H}$*. For* $\frac{H_m}{H} > \frac{1}{2}$ *the cost is always increasing in* $\frac{H_m}{H}$*.*

Again, for confirmation periods of 1, 10 and 24 hours and an annual $\delta = 0.1$ and $\alpha = 0.1$ the fees for which the cost of double spending is increasing in pool size are 0.011 %, 0.114

%, and 0.274 % respectively. We could also reverse the question and ask for how small decreases in value post attack the results hold? For a fee of 2 % the corresponding αs are 0.0006, 0.0056, 0.0134. This implies that even for almost insignificant decreases in value pools have higher cost for double spending attacks than outsiders.

This simple modelling exercise summarizes the main result of our model: the ubiquitous concern about centralization might merely be tilting at windmills. In the model section we attempt to build a robust grounding for what has been explicitly or implicitly assumed here. In addition, because many cryptocurrencies have relatively long block intervals (e.g. 10 minutes in Bitcoin) and only few confirmations required we will derive a discrete time equivalent which is more accurate for numerical illustrations.

5 Model

To better understand how concentration in a blockchain affects double spending attacks we consider pools and miners in an industrial organization framework. We find that concentration in mining power is harmless for the networks resilience against double spending attacks. The findings stem from the fact that, the larger a pool is, the more it loses if the network value collapses. Hence, even if a large pool is more able to conduct mischief, it should be less willing to do so. Our model is stylized, yet its intuition carries over to other settings where large miners, pools or coalitions receive economic profits.

5.1 Model Setup

Consider a world in which time is infinite and discrete and is indexed by t, $t = 0, 1, 2, \ldots$ There are two types of agents – miners and pools – having a discount factor $\beta \in (0, 1)$. Miners are homogeneous, risk averse and atomistic, whereas pools are risk neutral. In every period $t \geq 0$, miners choose their hashing power at a unit cost C, and hashing power allocation h_m for each pool $m \in \{1, 2, ..., M\}$ and h_0 for solo mining.

5.2 Mining Pools

Mining pools offer different fee and reward contracts; the simplest mechanisms being proportional payment and pay-per-share[12]. In a proportional reward system, whenever a pool wins a mining competition a miner receives

$$(1 - f^m)R\frac{h_i}{H_m} \tag{6}$$

where h_i is the miner's hash rate contributed to the pool m, R is block reward, H_m is the total hashing power in that pool and f^m is a fee collected by pool m.

The simplicity of proportional reward makes it vulnerable to block withholding and pool-hopping attacks, where a miner receives more than her proportional share of the rewards. Hence, more complicated reward mechanisms have been developed. The purpose of these structures is to ensure that miners receive rewards proportional to the hashing power they have contributed and avert other malicious behavior (Rosenfeld 2011). Assuming that these means are effective we can, without a loss of generality, model all pools' reward mechanisms as proportional.

Other typical reward contract is a pay-per-share reward mechanism, in which a pool effectively rents miner's hashing power and pays a rent regardless of whether the pool wins block rewards or not, fully insuring participating miners. However, pay-per-share is uncommon and usually associated with significantly higher fees. In addition, diversification of miner's hashing power to different pools would effectively insure miners against idiosyncratic risk. Hence, in our model we choose to concentrate on proportional reward mechanisms.

5.2.1 Collusive Equilibria

We restrict each pool's strategy to the standard supergame *grim trigger strategy*. Specifically, consider the following strategy for M incumbent pools to collude:

[12]For more a detailed description of the most common fee and reward mechanisms used by pools, see Appendix A.

1. *Collusion*: In every period, pools agree upon a fee f^c. Miners allocate their hashing power to pools.

2. *Punishment phase*: Once one of the incumbent pools does not have any participants, punishment phase is triggered and the pools enter into a Bertrand competition. In absence of marginal costs, and because the pools are homogeneous, the pools will receive zero profits.

In a collusive phase the pools discounted future profits are

$$\sum_{t=0}^{\infty} \beta^t \frac{H_m}{H} f^c R = \frac{f^c R}{1-\beta} \frac{H_m}{H} \tag{7}$$

where H is network's total hashing power and β is the time discount factor.

Theorem 3 *A collusive strategy is an equilibrium if*

$$\sum_{t=0}^{\infty} \beta^t \frac{H_m}{H} f^c R = \frac{f^c R}{1-\beta} \frac{H_m}{H} > f^c R \qquad \forall \quad \frac{H_m}{H} \tag{8}$$

Theorem 3 states that the profit from lowering the fee, and hence capturing the whole market, should be less than the value of discounted future profits in collusion phase. From this naturally follows:

Corollary 4 *If*

$$\exists \frac{H_m}{H} \qquad for \quad which \qquad \frac{H_m}{H} < \ 1-\beta \tag{9}$$

no collusion equilibrium exists.

Corollary 4 states that – given the discount factor – there should not exist extremely small pools for collusion equilibrium to exist. E.g. for annual β of 0.9, there should exist pools vesting less than 0.0002 percent of hashing power for collusion strategy not to be a Nash Equilibrium. For the remaining part of the analysis we will assume that such pools don't exist in the market, thus a collusion equilibrium can be sustained.

Above, we have assumed that R is constant. In reality, because rewards are paid in a cryptocurrency, they are highly volatile. In our model this would yield the same result, because pools are assumed to be risk neutral. In addition, (some) cryptocurrencies have expected declines in rewards (e.g. BTC reward is halved every 210,000 blocks, which occurs approximately every four years). It is a standard result that in these cases the benefit from deviating would be highest just prior to the expected decrease in reward (Rotemberg and Saloner 1986). For parsimony we have restricted our analysis from considering such cases.

5.2.2 Entry and Collusive Fee Setting

Every period $t \geq 0$ there exists a possible entrant pool without miners. Therefore, an entrant would set a fee $f^e < f^c$ to obtain miners. Prior to an entry the entrant pays a positive entry fee ζ. An entry will trigger the price competition phase and, hence, each pool makes zero profits post entry.

Therefore, a condition for a feasible entry is given by

$$f^e R - \zeta > 0 \quad \text{where} \quad f^e < f^c \tag{10}$$

Lemma 2 *It follows from feasible entry condition (Equation 10) that in order to deter entry colluding pools set a fee* f^c

$$f^c \leq \frac{\zeta}{R} \tag{11}$$

To keep the model parsimonious we have chosen a very simple barrier of entry as is manifested by Lemma 2. However, one could equivalently assume that, once an entry occurs only an active fraction of miners observes it. Hence the active miners would face a trade-off between lower fees and smaller diversification benefits. In this case, to deter entry incumbent pools' fee setting strategy should make active miners indifferent between choosing an entrant pool or staying in incumbent pools. In addition, incumbent pools have likely established

credibility for not siphoning rewards, having a reliable infrastructure etc. all attributes that an entrant might easily lack.

5.3 Miners

In every period t, a reward R is randomly assigned to a solo miner or a pool[13]. The probability of winning the reward in every period t is $\frac{h_i}{H}$ for a solo miner and $\frac{H_m}{H}$ for a pool, where H is network's total hashing power and H_m is pool m's hashing power. Whenever a pool wins the mining competition it collects a fee $f^m \in (0,1)$ and distributes the remaining reward to participants according to their contribution to the pool's total hashing power $\frac{h_i}{H_m}$. The miner j's expected utility at t for $t+1$ is hence given by the von-Neumann-Morgenstern Utility Function

$$U(H_j) = \frac{h_0}{H} u\left(R - C\sum_{i=0}^{M} h_i\right) + \sum_{i=1}^{M} \frac{H_m}{H} u\left((1-f^m)R\frac{h_i}{H_m} - C\sum_{i=0}^{M} h_i\right) \tag{12}$$

where, $U(\ )$ is a continuous, monotonic and concave utility function and h_0 is the allocation to solo mining and h_m $m \in [1,2,3,\ldots,M]$ are the allocations to M different pools. Each pool sets a fee f^m to maximize its profit.

5.3.1 Equilibrium Hashing Power and Allocation

Lemma 3 *Given fees and total hashing power, all miners' symmetric allocations among pools offering the lowest fee are Subgame Perfect Equilibria.*

Lemma 3 was initially discussed by Cong, He, and Li (2019). Following intuition of Modigliani and Miller (1958) the initial pool size does not matter whenever miner's are able to diversify by allocating their hashing power to multiple pools. Hence, any allocation where all pools get a share and is symmetric amongst the miners is a Nash Equilibrium. By a symmetric allocation we refer to an allocation in which each miner j allocates the same

[13]see Appendix C for more detailed discussion.

proportion of hashing power as all the other miners to each pool i.e. $\frac{h_{m,j}}{H_j} = \frac{h_{m,-j}}{H_{-j}}$ for each $m \in [1, 2, 3, \ldots, M]$ and j.

Lemma 4 *Miners allocate their hashing power amongst pools.*

To acquire miners, pools set fees for which miners prefer pools over solo mining. If miners are atomistic, once a miner prefers mining in pool(s) over solo mining all miners will prefer pools over solo mining. Because pools, in our model, do not have costs and miners are risk averse, there exists a fee $f^m > 0$ for which miners prefer pools and which pools are willing to offer.

Theorem 5 *Miners' utility function simplifies to the Bernoulli utility function*

$$U(H_i) = u\left((1 - f^c)R\frac{\sum_1^M h_i}{H} - C\sum_1^M h_i\right) \tag{13}$$

Proof. It follows from the assumptions that miners are atomistic and mining is competitive, that miners gain zero utility in equilibrium. Therefore, by employing Lemma 3 and Lemma 4 we get Theorem 5. ■

By allocating according to Lemma 3 miners are able to perfectly diversify mining risk. Total costs are equivalent to a net reward payed to miners. Hence, profits for miners are zero. This simplifies our analysis and corresponds to what is observed in most cryptocurrencies, namely that small scale mining is not profitable.

As proposed above, all miners symmetric allocations are Nash Equilibria. Miners, however, would need to coordinate to reach this allocation. Hence, to simplify our analysis we make the following assumption:

Assumption 1 *Miners coordinate their allocation amongst pools offering lowest fees at t by employing aggregate allocation at $t-1$ as a focal point in every period $t > 0$. Miners' allocation at $t = 0$ is exogenously given.*

In the absence of a definite coordination device, a focal point may function as such (Schelling 1960; Mehta, Starmer, and Sugden 1994; Bacharach and Bernasconi 1997). We argue that if a set of pools is homogeneous and provides the same service for the same price, previous aggregate allocation is a natural focal point for miners to allocate hashing power. This is accentuated, when there exists a large number of miners causing coordination to be unfeasible. An allocation determined by a focal point is an allocation in the set of possible Nash Equilibria allocations given by Lemma 3. The assumption implies that, ceteris paribus, pool sizes are stable[14].

Corollary 6 *In equilibrium total hashing power H is a function of f, R and C*

$$H = \frac{(1 - f^c)R}{C} \tag{14}$$

Corollary 6 follows from Theorem 5 by summing over all miners and it states that in equilibrium, because miners are fully insured against idiosyncratic shocks and make zero profits, total cost of hashing power equals the net reward.

5.4 Blockchain Security and Concentration

The purpose of proof-of-work (PoW)[15] is to make block generation expensive. This increases the cost of altering the history and spamming. Most cryptocurrency protocols follow the longest chain rule, in which blocks are linked to the chain that has the most blocks and hence the most proof-of-work invested in it. Proof-of-work and mining rewards[16] ensure that

[14]In reality pool sizes vary, however, as long as the current pool size is the best predictor of the next period pool size this does not affect our results.

[15]Early work on proof-of-work is difficult to track down in history. However, the term and its first formalization can be found in Jakobsson and Juels (1999), where authors propose the name Proof-of-Work for an idea already present in various works. Jakosson and Juels attribute the first conceptualization of PoW to Dwork and Naor (1992), who present an application to prevent spam by forcing the use of processing power to generate a cost to mail sending and thus to deter junk mail. Without knowing the contribution by Dwork and Naor, a similar idea was suggested 1997 by Back `www.hashcash.org/papers/announce.txt,1997`. In his 2002 paper he acknowledged the similarities between both ideas (Back et al. 2002). More detailed information about proof-of-work can be found in Becker et al. (2012) and Laurie (2004).

[16]e.g. in 2019 in Bitcoin the miner of a block receives a reward of 12.5 bitcoins, which are created with the new block. The amount halves every 210,000 blocks, so that the supply of bitcoins per block decreases over time until reaching total supply of 21 million Bitcoins

block-mining requires large amounts of processing power[17]. Because new blocks are added on top of an existing block[18], the probability of successfully mining a longer alternative chain starting from that block decreases exponentially with the number of blocks mined, unless an attacker holds more than 50% of the hashing power. Consequently, discussion on the security of open blockchain protocols is often focused on majority attacks.

Controlling over a half of the hashing power makes it technically feasible to fraudulently alter the chain. However, as will be demonstrated below, a party controlling enough hashing power seldom has incentives to conduct such attacks. A double spending attack may be facilitated without any direct cost by a pool controlling more than a half of the network's hashing power. This has been considered one of the main problems of pool concentration. The attack may also be facilitated by an entity controlling less than a half, if it acquires the required hashing power.

5.4.1 Cryptocurrency's Value Post Successful Double Spending Attack

Once sellers observe a double spending attack they can infer that someone has a capacity to conduct such attacks and these attacks are in those agents best interest. If the sellers could differentiate between those capable and incapable of conducting double spending attacks they could choose their customers accordingly. However, because blockchain protocols are anonymous by nature this is not achievable. Hence, the sellers would need to set a premium for payments made in that cryptocurrency. This however, would lead to devaluation of the currency and to a further need to increase prices. Without major changes in the protocol

[17]For example, in the case of Bitcoin from a global hash rates going under 0,15 TH/s in the beginning of 2011 to values fluctuating around 50 000 000 TH/s during the first quarter of 2019. Energy wise, if Bitcoin were a country, it would rank around 40th in energy consumption per year, slightly behind Chile.

[18]However, it is not necessarily incentive compatible to follow the rule: Kroll, Davey, and Felten (2013) argue that mining the longest chain is one possible Nash equilibrium, perpetuated mainly because it acts as a focal point. However, there exist infinitely many alternative equilibria, which could result if a large enough actor or coalition impels them. Biais, Bisière, et al. (2018) demonstrate that although miners usually find it optimal to cooperate, there are situations where coordination is sub-optimal causing a blockchain to fork into two or more alternative main chains (e.g. Ethereum Ethereum Classic fork discussed above). Furthermore, also Yermack (2017) expresses concerns regarding the use of both public and private blockchains. He elaborates that even decentralized public chains are subject to changes *ex post*, if sufficient proportion of chain's members join to undo some outcome.

design, this would eventually lead to a collapse in the value. The situation is reminiscent of Akerlof (1970) lemons problem, although what in our case destroys the market is the information asymmetry on the value of the payment rather than uncertainty about the value of the purchased good. Hence, we make the following assumption

Assumption 2 *Once a double spending attack is successfully conducted, trust in the network vanishes and value of the future rewards decreases to zero.*

This assumption is similar to Kroll, Davey, and Felten (2013), where after a double spending attack sellers cease accepting the cryptocurrency due to the impossibility of distinguishing fraudulent transactions from valid ones.

The outcome would be the same if we assumed that only a fraction of value is lost due to an attack. Because it would be an optimal strategy to attack subsequently, rational agents would infer from one attack that the there will be a series of attacks, eventually driving the value of the currency to zero. Therefore, through agents' anticipation, the value of rewards would decrease to zero once an attack has taken place. Later we will demonstrate that the premise of our model holds even if only a residual of value is destroyed because of a double spending attack. This might occur if e.g. there were substantial changes in the protocol increasing difficulty to conduct double spending attacks.

5.4.2 Cost of Conducting Double Spending Attack

An agent controlling any positive amount of hashing power may attempt to conduct a double spending attack, and will succeed with some strictly positive probability. Hence, if the pools hashing power does not decrease due to an attack attempt (and attacking is free), a pool could conduct an infinite number of attempts and would therefore with certainty conduct a successful double spending attack for free for any positive amount of controlled hashing power. In reality, after a while, pool members would detect that the pool is trying to conduct a double spending attack (or siphoning rewards), because pools mining forked chains do not

receive rewards in the main chain and hence cannot reward pool members. To avoid such an obscure result we postulate:

Assumption 3 *Once commenced a double spending attack cannot be canceled and it has to conclude at some arbitrary time T.*

To conduct a double spending attack an agent has to control a sufficient amount of hashing power. If a miner lacks it, she may increase her capacity buy acquiring required facilities, mining gear or by renting hashing power from other miners. These two options are analogous to buying computer storage or renting it from a cloud service provider. For the rest of the paper we will assume that the market for hashing power is frictionless.

Assumption 4 *Market for hashing power is frictionless.*

An attacker has to wait for a certain number of blocks K to be mined in the main chain before the seller releases the goods. After this, the attacker broadcasts a chain which is longer than the main chain and hence becomes the new valid chain. The attacker chooses hashing power $\frac{H_a+H_m}{H}$ to minimize the expected cost of conducting a successful attack, where H_a is the hashing power that an attacker acquires in addition to H_m which is the hashing power already controlled by the attacker.

There exists two conditions for a successful attack:

1. The main chain has reached the block where goods are released.

2. The attacker's chain must be one block longer than the main chain.

Implying that $B_{attack} > B_{main} \geq K$ where B_{attack} is the number of blocks mined by the attacker, B_{main} the number of blocks added to the main chain starting from the block including the fraudulent transaction and K is the number of confirmation blocks starting from the transaction block.

Lemma 5 *In a frictionless market, once the main chain has reached a length where goods are delivered, an attacker will acquire all available hashing power and then broadcast a chain that is one block longer with a cost of* $(K+1-B_{attack})\left(1-\frac{H_m}{H}\right)$.

See Appendix D for a proof.

Lemma 6 *While* $B_{main} < K$ *an attacker's hashing power is* $\frac{H_m}{H}$.

An attack may conclude in two manners: First, an attacker has mined $K+1$ blocks before $B_{main} = K$ and waits before broadcasting the new blocks. Second, $B_{main} = K$ and an attacker mines $B_{main} + 1 - B_{attack}$ blocks. While $B_{main} < K$ an attacker has a strictly positive probability of mining enough blocks for a successful attack without any cost. Hence, by increasing hashing power over $\frac{H_m}{H}$ an attackers increases the probability of "wasting" free hashing power.

See Appendix D for a proof.

By combining Lemmas 5 and 6, the expected direct cost for conducting a double spending attack can be expressed as

$$\sum_{a=0}^{K} \underbrace{\left(1-\frac{H_m}{H}\right)^K \left(\frac{H_m}{H}\right)^a \binom{K+a}{a}}_{\text{Probability mass function Pr(X=K)}} \underbrace{\left(1-\frac{H_m}{H}\right)(K+1-a)CH}_{\text{Cost for each a}} \tag{15}$$

If we substitute CH with the equilibrium condition $CH = (1-f^c)R$ given in Corollary 6, and consider the opportunity cost $\frac{H_m}{H}\frac{f^cR}{1-\beta}$ (Equation 7) from losing the future profits, we get the following Theorem:

Theorem 7 *In a frictionless markets, the expected cost for conducting a double spending attack for* $\frac{H_m}{H} \leq \frac{1}{2}$ *is*

$$\sum_{a=0}^{K} \underbrace{\left(1-\frac{H_m}{H}\right)^K \left(\frac{H_m}{H}\right)^a \binom{K+a}{a}}_{\textit{Probability mass function Pr(X=K)}} \underbrace{\left(1-\frac{H_m}{H}\right)(K+1-a)(1-f^c)R}_{\textit{Cost for each a}} + \underbrace{\frac{f^cR}{1-\beta}\frac{H_m}{H}}_{\textit{Opportunity cost}} \tag{16}$$

From Theorem 7 we may observe that cost of conducting a double spending attack is increasing in R and K. This is intuitive because, the longer the confirmation period (K) and the higher the reward (R), the costlier it is produce sufficient amount of proof-of-work.

5.4.3 Main Results

For $\frac{1}{2} < \frac{H_m}{H} \leq 1$ a double spending attack may be conducted without any direct costs and opportunity costs are strictly increasing in $\frac{H_m}{H}$ we therefore limit our focus to the nontrivial case where $0 < \frac{H_m}{H} \leq \frac{1}{2}$.

Theorem 8 *While*

$$(1-\beta)\left(2(K+1) - \frac{(K+2)\binom{2K+2}{K+2}}{2^{2K+1}}\right) \leq \frac{f^c}{1-f^c} \tag{17}$$

for all $0 < \frac{H_m}{H} \leq \frac{1}{2}$ *the cost of double spending is higher than for* $0 = \frac{H_m}{H}$.

See Appendix D for a proof.

From Theorem 8 it follows that for reasonably small positive fees and high annual discount rates, network concentration is not harmful for maintaining honest conduct. Quite contrary, larger pools are less likely to conduct a double spending attack. From Equation 17 we may observe that the threshold for f^c is increasing in K and decreasing in β. For an annual $\beta = 0.9$, average block intervals of 10 minutes and different $K = 6$, 60, 600 and 6000 the corresponding thresholds for fees are 0.003, 0.024, 0.24, and 2.3 percent. Virtually the same thresholds apply to all major cryptocurrencies. The current pool average fee in e.g. Bitcoin is approximately 2 percent (see e.g. Cong, He, and Li (2019)). Obviously these fees are not all economic profit. However, for example, the largest pool operator – Bitmain – had from January 2015 to June 2018 gross profit margins ranging between 80 and 89 percent (BitMain 2018). This demonstrates that even with rather conservative discount rates, limited economic profits are enough to sustain desired equilibrium, where pool concentration is not harmful.

Table 1: This table contains the corresponding thresholds for 13 PoW based cryptocurrencies. Median confirmations indicate the median number of confirmations required before a cryptocurrency deposit is credited to the holder's account and is available for withdrawal. The information is retrieved from eight major cryptocurrency exchanges from their APIs, by opening an account or from their web pages. An annual $\beta = 0.9$. All the data is collected in March 2020.

Cryptocurrency	Market Cap in Millions of USD	Median Confirmations	Median Conf. Time (h)	Threshold fee	Threshold fee $\alpha = 0.1$
Bitcoin	159,399	2.5	0.42	0.00082 %	0.00823 %
Ether	24,679	18.5	0.07	0.00014 %	0.00143 %
Bitcoin Cash	5,906	12	2	0.00398 %	0.03984 %
Bitcoin SV	4,353	72	12	0.02473 %	0.24730 %
Litecoin	3,842	6	0.25	0.00050 %	0.00502 %
Monero	1,180	13.5	0.45	0.00089 %	0.00889 %
Ethereum Clas.	854	450	1.75	0.00372 %	0.03715 %
Zcash	467	15	0.63	0.00125 %	0.01248 %
Dogecoin	285	9	0.15	0.00030 %	0.00299 %
Bitcoin Gold	172	36	6	0.01218 %	0.12178 %
DigiByte	72	15	0.06	0.00012 %	0.00125 %
Verge	60	120	1	0.00208 %	0.02082 %
Vertcoin	16	600	25	0.01060 %	0.10605 %

To further examine the thresholds for realistic K we collect deposit confirmation data from eight different major cryptocurrency exchanges[19]. The data is collected directly from the exchanges' official web pages, from the corresponding APIs or by registering to the exchange whenever that is possible without excessive compromise of personal privacy. Some exchanges list two different confirmation numbers. The first is for the deposit to be credited to the account and the second is for a possible withdrawal, whenever this is the case we choose the larger number of confirmations (i.e. the latter). Then we calculate the median for each of the PoW based cryptocurrencies. We choose to conduct this analysis with exchanges deposit confirmations, because of their prominent role in the cryptocurrency ecosystems and the large

[19]The cryptocurrency exchanges are chosen according to a Bloomberg news article in 2018 https://www.bloomberg.com/news/articles/2018-03-05/crypto-exchanges-raking-in-billions-emerge-as-kings-of-coins. However, one of the exchanges had suspended deposits and withdrawals and another had no (easily) available data.

amount of cryptocurrencies deposited to them deeming them vulnerable to double spending attacks. From the numerical illustration in Table 1 we may observe that the thresholds are 100-10,000 times lower than the 2 % fee. This further corroborates robustness of our claim that indeed pools have higher costs for double spending than outsiders.

The longer the confirmation period K is, the larger the proof-of-work burden becomes for an attacker. This affects directly the cost of double spending. One might argue that for a major attack, an attacker would need to conduct a large transaction and hence large transactions should have long confirmation periods. However, to conduct a double spending attack one could conduct multiple small transactions without being observed; and hence, we argue that K should correspond to the average time in a particular cryptocurrency protocol. For a detailed discussion about confirmation times see Chiu and Koeppl (2019a). In addition, it would be hard to argue that pools would benefit form the hashing power they control if the confirmation periods where longer, because during an attack the pools would not be able to distribute rewards for the participants e.g. $K = 6000$ and 10 minute block intervals corresponds to an attack that would require on average 42 days. It is reasonable to assume that during those 42 days miners would exit the mining pool.

5.4.4 Robustness of Results Under Relaxed Assumption 2

In Assumption 2 we postulate that the value of a cryptocurrency decreases to zero after a major double spending attack. As discussed above, this can be motivated in various ways. However, one might still argue that after a double spending attack there might be protocol changes or substantial increases in K by the merchants etc. Effectively causing agents to maintain some trust in the protocol. A simple modification of Theorem 8 lets us examine such case. For illustrative purposes let us assume that there exists $\alpha \in (0, 1]$, which indicates an exogenously given fraction lost in the value of a cryptocurrency post a successful double spending attack.

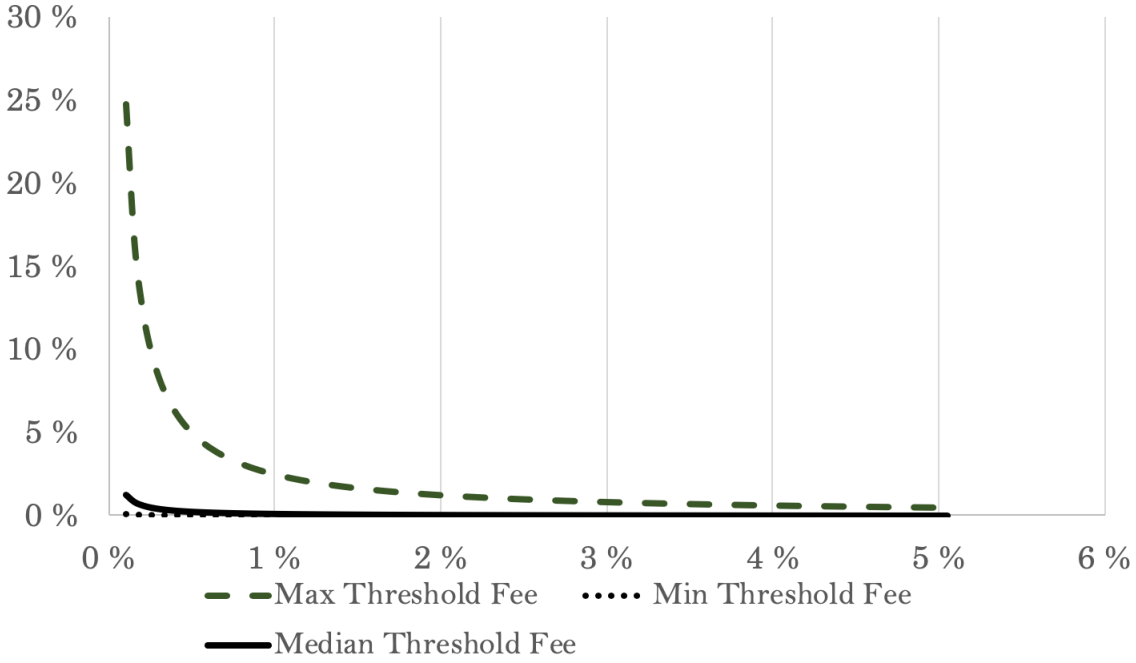

Figure 2: Demonstrates the relationship for percentage decrease in the value of a cryptocurrency after a successful double spending attack (horizontal-axis) and the corresponding threshold fee for double spending to be costlier for a pool than for an outsider (vertical-axis). Max, Min, and Median Threshold Fees are from those reported in Table 1.

Corollary 9 *While*

$$\frac{(1-\beta)}{\alpha}\left(2(K+1)-\frac{(K+2)\binom{2K+2}{K+2}}{2^{2K+1}}\right)\leq\frac{f^c}{1-f^c} \tag{18}$$

for all $0<\frac{H_m}{H}\leq\frac{1}{2}$ *the cost of double spending is larger than for* $0=\frac{H_m}{H}$.

The first column from the right in Table 1 reports the corresponding thresholds for a rather conservative case of 10 % decrease in the value of a cryptocurrency post successful double spending attack. The fee thresholds are highest for Bitcoin SV, Bitcoin Gold and Vertcoin. However, for none of them is the threshold in the neighborhood of 2 %.

To further illustrate the robustness of our results Figure 2 reports the median, maximum, and minimum for the threshold fees of the 13 cryptocurrencies in Table 1. Even for extremely conservative decreases in value post a double spending attack (horizontal-axis) the threshold fees (vertical-axis) are in the neighborhood of 2 %. Only for decreases less than 1 % the

cryptocurrency with the highest threshold Bitcoin SV has considerably higher threshold fees than 2 %. Whereas for median threshold fees even the smallest effect reported 0.1 % does not increase the threshold close to 2 %.

6 Conclusion

Concentration of mining to large pools in different blockchains has been considered a vital threat to trust and viability. This stems from the fact that in a concentrated network, large pools can conduct double spending attacks more easily. If a pool controlled more than half of the network's hashing power, it would be able to use the same funds multiple times. However, mining pools' present value is dependent on the value of the network: Pool's future profits consist of collected fees which are proportional to the rewards paid in the cryptocurrency. These profits are hence dependent on the value of the cryptocurrency. By attacking against the network a pool would lose it's future profits through a collapse in the network's value. We show that, even if large pools are more able to conduct double spending attacks, they are less willing to do so.

Our analysis focuses on simple and intuitive economic incentives for large pools to maintain honest conduct. There are other aspects such as political, ideological or environmental aspects to be considered in order to gain a more holistic understanding of the role of concentration. We derive thresholds of fees, discount rates, post attack decreases in value, and confirmation periods for which large pools would be worse off by double spending, even though they are capable of conducting it. We conclude that the historically observed pool concentration does not indicate a higher risk of double spending attacks. This holds with conservative discount rates, pool fees and confirmation periods, which are aligned with those observed in main cryptocurrencies. Hence, our result directly contradicts the common belief that concentration is harmful. This result demonstrates the well-known economic insight that feasibility does not imply desirability.

References

Abadi, Joseph and Markus Brunnermeier (2018). *Blockchain economics*. Tech. rep. National Bureau of Economic Research.

Akerlof, George A. (1970). "The Market for "Lemons": Quality Uncertainty and the Market Mechanism". In: *The Quarterly Journal of Economics* 84.3, pp. 488–500. ISSN: 00335533, 15314650. URL: http://www.jstor.org/stable/1879431.

Alsabah, Humoud and Agostino Capponi (2019). "Pitfalls of Bitcoin's Proof-of-Work: R&D arms race and mining centralization". In: *Available at SSRN 3273982*.

Arnosti, Nick and S Matthew Weinberg (2018). "Bitcoin: A natural oligopoly". In: *arXiv preprint arXiv:1811.08572*.

Auer, Raphael (2019). "Beyond the doomsday economics of "proof-of-work"". In: *Cryptocurrencies (February 2019). CEPR Discussion Paper No. DP13506*.

Bacharach, Michael and Michele Bernasconi (1997). "The variable frame theory of focal points: An experimental study". In: *Games and Economic Behavior* 19.1, pp. 1–45.

Back, Adam et al. (2002). "Hashcash-a denial of service counter-measure". In: URL: http://www.hashcash.org/papers/hashcash.pdf.

Becker, Jörg, Dominic Breuker, Tobias Heide, Justus Holler, Hans Peter Rauer, and Rainer Böhme (Jan. 2012). "Can We Afford Integrity by Proof-of-Work? Scenarios Inspired by the Bitcoin Currency". In: DOI: 10.1007/978-3-642-39498-0_7.

Biais, Bruno, Christophe Bisière, Matthieu Bouvard, and Catherine Casamatta (Jan. 2018). *The blockchain folk theorem*. Swiss Finance Institute Working Paper 17-75. Institut d'Économie Industrielle (IDEI), Toulouse. URL: https://dx.doi.org/10.2139/ssrn.3108601.

Biais, Bruno, Christophe Bisiere, Matthieu Bouvard, Catherine Casamatta, and Albert J Menkveld (2018). "Equilibrium bitcoin pricing". In: *Available at SSRN 3261063*.

BitMain, Technologies Holding Company (2018). *Application Proof of Bitmain*. Tech. rep. URL: http://www.hkexnews.hk/APP/SEHK/2018/2018092406/Documents/SEHK201809260017.pdf.

Böhme, Rainer, Nicolas Christin, Benjamin Edelman, and Tyler Moore (2015). "Bitcoin: Economics, technology, and governance". In: *Journal of Economic Perspectives* 29.2, pp. 213–38.

Bonneau, Joseph (2016). "Why buy when you can rent?" In: *International Conference on Financial Cryptography and Data Security*. Springer, pp. 19–26.

Bonneau, Joseph, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten (2015). "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies". In: *2015 IEEE Symposium on Security and Privacy*. IEEE, pp. 104–121.

Budish, Eric (2018). *The economic limits of bitcoin and the blockchain*. Tech. rep. National Bureau of Economic Research.

Chaum, David (1982). "Blind signatures for untraceable payments (1983)". In: *Advances in Cryptology*.

Chiu, Jonathan and Thorsten V Koeppl (2019a). "Blockchain-Based Settlement for Asset Trading". In: *Review of Financial Studies* 32.5, pp. 1716–1753. URL: https://EconPapers.repec.org/RePEc:oup:rfinst:v:32:y:2019:i:5:p:1716-1753..

Chiu, Jonathan and Thorsten V Koeppl (2019b). "Blockchain-based settlement for asset trading". In: *The Review of Financial Studies* 32.5, pp. 1716–1753.

Cong, Lin William and Zhiguo He (Apr. 2019). "Blockchain Disruption and Smart Contracts". In: *The Review of Financial Studies* 32.5, pp. 1754–1797. ISSN: 0893-9454. DOI: 10.1093/rfs/hhz007. eprint: https://academic.oup.com/rfs/article-pdf/32/5/1754/28275179/hhz007.pdf. URL: https://doi.org/10.1093/rfs/hhz007.

Cong, Lin William, Zhiguo He, and Jiasun Li (2019). *Decentralized mining in centralized pools*. Tech. rep. National Bureau of Economic Research.

Cong, Lin William, Ye Li, and Neng Wang (2019). "Tokenomics: Dynamic adoption and valuation". In: *Becker Friedman Institute for Research in Economics Working Paper* 2018-49, pp. 2018–15.

Dimitri, Nicola (2017). "Bitcoin mining as a contest". In: *Ledger* 2, pp. 31–37.

Dwork, Cynthia and Moni Naor (1992). "Pricing via processing or combatting junk mail". In: *Annual International Cryptology Conference*. Springer, pp. 139–147.

Easley, David, Maureen O'Hara, and Soumya Basu (2019). "From mining to markets: The evolution of bitcoin transaction fees". In: *Journal of Financial Economics*.

Eyal, Ittay and Emin Gün Sirer (2014). "Majority is not enough: Bitcoin mining is vulnerable". In: *International conference on financial cryptography and data security*. Springer, pp. 436–454.

Ferreira, Daniel, Jin Li, and Radoslawa Nikolowa (2019). "Corporate capture of blockchain governance". In: *Available at SSRN 3320437*.

Gandal, Neil and Hanna Halaburda (2016). "Can we predict the winner in a market with network effects? Competition in cryptocurrency market". In: *Games* 7.3, p. 16.

Gans, Joshua S and Hanna Halaburda (2015). "Some economics of private digital currency". In: *Economic Analysis of the Digital Economy*. University of Chicago Press, pp. 257–276.

Gencer, Adem Efe, Soumya Basu, Ittay Eyal, Robbert Van Renesse, and Emin Gün Sirer (2018). "Decentralization in bitcoin and ethereum networks". In: *arXiv preprint arXiv:1801.03998*.

Green, Edward J and Robert H Porter (1984). "Noncooperative collusion under imperfect price information". In: *Econometrica: Journal of the Econometric Society*, pp. 87–100.

Haber, Stuart and W Scott Stornetta (1990). "How to time-stamp a digital document". In: *Conference on the Theory and Application of Cryptography*. Springer, pp. 437–455.

Hinzen, Franz J, Kose John, and Fahad Saleh (2019). "Proof-of-work's limited adoption problem". In: *NYU Stern School of Business*.

Huang, Gregory T (2003). "The Web's new currency". In: *Technology Review* 106.10, pp. 28–28. URL: https://www.technologyreview.com/s/402317/the-webs-new-currency/.

Huberman, Gur, Jacob Leshno, and Ciamac C Moallemi (2017). "Monopoly without a monopolist: An economic analysis of the bitcoin payment system". In: *Bank of Finland Research Discussion Papers*. URL: https://helda.helsinki.fi/bof/bitstream/handle/123456789/14912/BoF_DP_1727.pdf.

Iyidogan, Engin (2019). "An equilibrium model of blockchain-based cryptocurrencies". In: *Available at SSRN 3152803*.

Jakobsson, Markus and Ari Juels (1999). "Proofs of work and bread pudding protocols". In: *Secure Information Networks*. Springer, pp. 258–272.

Judmayer, Aljosha, Nicholas Stifter, Alexei Zamyatin, Itay Tsabary, Ittay Eyal, Peter Gazi, Sarah Meiklejohn, and Edgar Weippl (2019). *Pay-to-win: Incentive attacks on proof-of-work cryptocurrencies*. Tech. rep. Cryptology ePrint Archive, Report 2019/775.

Kaiser, Ben, Mireya Jurado, and Alex Ledger (2018). "The looming threat of China: An analysis of Chinese influence on Bitcoin". In: *arXiv preprint arXiv:1810.02466*.

Kiayias, Aggelos, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis (2016). "Blockchain Mining Games". In: *Proceedings of the 2016 ACM Conference on Economics and Computation*. EC '16. Maastricht, The Netherlands: ACM, pp. 365–382. ISBN: 978-1-4503-3936-0. DOI: 10.1145/2940716.2940773. URL: http://doi.acm.org/10.1145/2940716.2940773.

Kroll, Joshua A, Ian C Davey, and Edward W Felten (2013). "The economics of Bitcoin mining, or Bitcoin in the presence of adversaries". In: *Proceedings of WEIS*. Vol. 2013, p. 11.

Laurie, Ben (July 2004). "Proof-of-Work" Proves Not to Work". In:

Lee, Larissa (2016). "New Kids on the Blockchain: How Bitcoin's Technology Could Reinvent the Stock Market". In: *Hastings Business Law Journal* 12 (2). URL: http://dx.doi.org/10.2139/ssrn.2656501.

Ma, June, Joshua S Gans, and Rabee Tourky (2018). *Market structure in bitcoin mining*. Tech. rep. National Bureau of Economic Research.

Mehta, Judith, Chris Starmer, and Robert Sugden (1994). "The nature of salience: An experimental investigation of pure coordination games". In: *The American Economic Review* 84.3, pp. 658–673.

Modigliani, Franco and Merton H Miller (1958). "The cost of capital, corporation finance and the theory of investment". In: *The American Economic Review* 1, p. 3.

Nakamoto, Satoshi (2008). "Bitcoin: A peer-to-peer electronic cash system". In:

Neitz, Michele Benedetto (2019). "The Influencers: Facebook's Libra, Public Blockchains, and the Ethical Considerations of Centralization". In: *North Carolina Journal of Law and Technology, Forthcoming*.

Oxford English Dictionary (2019). Ledger. Oxford University Press. URL: https://en.oxforddictionaries.com/definition/ledger.

Pagnotta, Emiliano and Andrea Buraschi (2018). "An equilibrium valuation of bitcoin and decentralized network assets". In: *Available at SSRN 3142022*.

Pease, M., R. Shostak, and L. Lamport (Apr. 1980). "Reaching Agreement in the Presence of Faults". In: *J. ACM* 27.2, pp. 228–234. ISSN: 0004-5411. DOI: 10.1145/322186.322188. URL: http://doi.acm.org/10.1145/322186.322188.

Rosenfeld, Meni (2011). "Analysis of bitcoin pooled mining reward systems". In: *arXiv preprint arXiv:1112.4980*.

Rosenfeld, Meni (2014). "Analysis of hashrate-based double spending". In: *arXiv preprint arXiv:1402.2009*.

Rotemberg, Julio and Garth Saloner (1986). "A Supergame-Theoretic Model of Price Wars during Booms". In: *American Economic Review* 76.3, pp. 390–407. URL: https://EconPapers.repec.org/RePEc:aea:aecrev:v:76:y:1986:i:3:p:390-407.

Saleh, Fahad (2019). "Blockchain without waste: Proof-of-stake". In: *Available at SSRN 3183935*.

Schelling, Thomas C (1960). *The strategy of conflict*. Harvard university press.

Stoll, Christian, Lena Klaaßen, and Ulrich Gallersdörfer (2019). "The carbon footprint of bitcoin". In: *Joule* 3.7, pp. 1647–1661.

Van Wirdum, Aaron (2018). *How the Bitcoin Cash "Hash War" Came and Went and Not Much Happened*. URL: https://bitcoinmagazine.com/articles/week-2-how-bitcoin-cash-hash-war-came-and-went-and-not-much-happened (visited on 06/30/2019).

Yermack, David (Jan. 2017). "Corporate Governance and Blockchains*". In: *Review of Finance* 21.1, pp. 7–31. ISSN: 1572-3097. DOI: 10.1093/rof/rfw074. eprint: http://oup.prod.sis.lan/rof/article-pdf/21/1/7/26322010/rfw074.pdf. URL: https://doi.org/10.1093/rof/rfw074.

Appendix A. Main categories of mining pools by fee types

Mining pools use various fee and payment schemes. This section presents the most typical categories. A deeper presentation of the multiple payment schemes can be found in Rosenfeld (2011).

Pay-per-Share (PPS): the most basic insurance mechanism in pool mining. The pool rents hashing power from the miners at fixed price, regardless of whether the pool is able to mine blocks or not. Therefore it transfers the risk to the pool managers. Typically, PPS pools have the highest fees.

Shared Maximum Pay Per Share (SMPPS): especial kind of PPS that limits the payment to the miners to the earnings of the pool.

Proportional: Similar to PPS, but rewards are shared only when the pool creates a block.

Pay Per Last N Shares (PPLNS): as the number of hashes used to get a block varies from one block to another, PPLNS uses an especial kind of proportional fee setting contract. The rewards are distributed when the pool finds a block, but rather than paying by share of hashes divided by the total number of hashes needed for that block, it divides the share of hashes provided by the miner by a fixed number.

Slush's Bitcoin Pooled Mining (BMP) or Score: Introduced by Slush pool, uses a proportional scheme that weights shares to time during one mining round, so that later shares are rewarded more than early ones. The system was introduced to desincentivize pool switching during one round.

Geometric method (GM): the pool first takes a fixed fee from the block reward and distributes the rest among all miners in proportion to their score keeping the expected payoff per submitted share constant regardless of time during one round (mining one block) (ibid.).

Double Geometric Method (DGM): a midway between GM and PPLNS. At every

new block part of the score of the miners is transferred to the pool, so that if multiple blocks are found in a row, the pool keeps more rewards. On the other hand, as the time until forging a valid block increases, the miners keep a higher expected revenue. This method moves most of the risk to the pool, and thus is beneficial to it when the pool successfully finds multiple blocks, and also for the risk-averse miners as their revenues get insurance against the pool not finding a block in a longer time (Rosenfeld 2011).

Peer-to-Peer Mining Pool (P2Pool): similar to proportional pools but works in a decentralized peer-to-peer architecture. Miners in the pool work their own independent blockchain and, when finding a block that meets the difficulty criterion of the main chain, they merge it into the main chain and share the rewards proportionally to their share in their own independent blockchain. The goal of a P2Pool is to decentralize pool mining by substituting its central server with a P2P network owned by the members of the pool.

Appendix B. Glossary of Blockchain Terminology

51% Attack: A miner or group of miners that own more than half of the hashing power in the network and use it to generate an alternative chain that contains fraudulent blocks.

Altcoin: Alternative cryptocurrencies.

Application Specific Integrated Circuit (ASIC): A chip designed to complete very efficiently an specific task. The apparition of ASICs designed for Bitcoin mining in 2013 raised exponentially the total hash of the network and made mining with normal computers completely obsolete.

Bitcoin: The decentralized peer-to-peer cryptographic currency that introduced proof-of-work based blockchain technology. We use the term Bitcoin for the protocol, the network and the currency. The term bitcoin without capital letters refers to a unit of cryptocurrency. Bitcoin was introduced together with blockchain technology and thus was its first application.

Bitcoin Whitepaper: The paper "Bitcoin: A Peer-to-Peer Electronic Cash System"

by Satoshi Nakamoto (2008) that introduced the concepts of Blockchain and of Bitcoin, together with its protocol. Available online here: bitcoin.org/bitcoin.pdf.

Block: A compilation of transactions, information, and a header. Blockheaders contain a reference to the previous block, which in turn refers to the previous one and so forth until the first block. This main structure of connected blocks gives name to the blockchain.

Block reward: Most blockchain protocols assign a reward when a miner adds a new block to the blockchain. Normally the reward is based on transaction fees and the creation of new units of the currency at every block . Rewards create incentives to mine and to keep the blockchain consistent. Bitcoin's protocol sets that the reward per block began at 50 BTC and is halved every 280 000 blocks. Once the total amount of bitcoins reaches 21 million, the block reward will be only the transaction fees decided by the users and no new bitcoins will be created.

Blockchain: The technology used to concatenate information in blocks and form a ledger or list of events. Blockchains can be private or public (centralized or decentralized). Blockchain technology was first proposed in Nakamoto's Whitepaper.

Byzantine Generals' Problem: The problem of gaining consensus in a network of parties where some might be faulty or fraudulent and there is no information about how to know which ones.

Cryptocurrency: A digital unit of value used as currency and based in the use of cryptography.

Cryptography: The science of securing a message and/or its parts through ciphers and codes.

Difficulty Level: The number of consecutive zeroes required in the output of the encryption function used in a blockchain. Typically this number is adjusted dynamically as the amount of hash in the network changes to keep the expected time between blocks constant.

Double spending: Using the same units of cryptocurrency two times by substituting the block in the main chain where those cryptocurrencies were used. Double spending becomes

virtually impossible after few blocks, unless the party that undertakes the fraud owns more than half of the hashing power of the network.

Elliptic Curve Cryptography: The cryptographic algorithm used to generate Public Keys from Private Keys.

Ethereum: A public blockchain cryptocurrency focused in smart contracts that was introduced in 2015. Ethereum uses a proof-of-work algorithm called Ethash that aims to reduce the prominence of ASIC mining. In July, 2016 Ethereum had a hard fork, resulting in Ethereum and Ethereum Classic.

Fork: The division of a blockchain into two or more alternative chains. The forks create different tracks of transactions and thus distort ownership. Typically a fork will end when one of the chains becomes longer. However, both chains might continue to exist as independent blockchains, in which case is called a hard-fork. Prominent examples of hard forks are the split of Bitcoin into Bitcoin and Bitcoin Cash in August, 2017, and the split between Ethereum and Ethereum Classic in July, 2016.

Genesis Block: The first block of a blockchain. The Genesis Block is especial because it is the only block that doesn't refer to previous blocks.

Goldfinger attack: An attack that aims to disrupt a blockchain protocol or its consensus. The aim of the attack is not based in incentives related to gaining power or utility from the cryptocurrency but from damaging it.

Hash rate/hashing power: The amount of nonces (numbers) that a processor can generate per second, i.e. the processing power of a machine, when used to calculate hashing functions.

Initial Coin Offering: To offer initial units of a cryptocurrency in order to raise funds to establish it.

Miner: A node in a blockchain network that generates new blocks.

Mining: The action of forging new blocks by finding a nonce that, together with the rest of the blockheader, when used as input in the hashing function gives as output a number

small enough to meet the difficulty criteria.

Mining Pool: A risk-sharing coalition of miners where each miner participates with its own hash and rewards are distributed following some payoff mechanism. Typically, a mining pool will charge a fee to participate and will either divide won rewards among its participants or will pay per hash, independently of whether a pool member mines the next block or not.

Node: A computer in a blockchain network.

Nonce: A random number that forms part of the content of a blockheader. The nonce is the only variable content in the block. It is changed so that the blockheader's content, when used as input in a hashing function, produces as output a number that meets the difficulty criteria.

Orphan Block: A properly mined block that ends in a discontinued fork.

Peer-to-Peer (P2P): A system of interconnected nodes that doesn't rely in a central coordinator such as a server or similar. Is widely used for file-sharing networks, cryptocurrencies and other applications. Typically P2P systems are open and based in decentralized cooperation.

Private Key: A secret code that is used for message ciphering. It is used together with the public key, such that public keys are widely known and used to encript a message that only the holder of the private key can read. In blockchains private keys are used to sign transactions.

Proof-of-stake (PoS): An alternative consensus protocol to proof-of-work. In PoS protocols the right to include the next block is based on the stake of the node. PoS was developed mainly to reduce the energy externalities of proof-of-work protocols, however its effect on incentives is unclear and has risen criticism.

Proof-of-work: Used in the blockchain to ensure that creation of blocks is costly and thus, deter spam generation of blocks. Abbreviated as PoW, it is normally based in using processing power to find the input to a hash function such that its output meets a difficulty criteria.

Public Key: A cryptographic key that refers to a person and allows to encrypt messages so that the owner of the key can decipher them using his private key.

Quantum computing attack: The use of quantum computers to hack a private key associated with a public key. This kind of attacks are still speculative, but might become more relevant as quantum computing technology develops. The use of quantum computers reduces drastically the number of trials the computer needs to discover a private key.

Satoshi: One hundred millionth of a bitcoin. Named after the author of the Bitcoin White Paper. One satoshi is the smallest fraction of a bitcoin accepted by the Bitcoin protocol.

SHA-256: A version of the Secure Hash Algorithm 2, whose output is a 256 bit string. It is the function used as PoW when mining bitcoins, so that the block-header is the input to the SHA-256 function and the output is the hash number of the block.

Transaction fee: An amount of currency linked to a transaction and that is assigned to the miner that includes the transaction into a validated block. Some cryptocurrencies require mandatory fees, while others -such as Bitcoin- rely on voluntary fees.

Appendix C. Difficulty Adjustment

To generate a new block a miner needs to guess a *nonce* which, with the other information in the block header, generates a hash number meeting a difficulty restriction. In proof-of-work protocols valid nonces are found by trying random numbers until one of them, together with the rest of the content of the block, generates via hash function a hash number that meets this criteria. This process requires computational power and, therefore, is costly. As an example, Bitcoin employs as PoW a difficulty criterion that dynamically sets a required count of consecutive zeros at the beginning of the hash number. The larger the number of zeros required by the difficulty criterion, the harder it is to find a proper nonce.

The difficulty adjustment (at least in the major protocols) serves two purposes: First,

it keeps the expected addition rate of new blocks constant (e.g. 10 minutes in Bitcoin, and 17 to 19 seconds in Ethereum). Hence, the time between blocks follows an exponential distribution, meaning that blocks are mined following a Poisson process at a constant average rate regardless of the network's total hashing power. Second, if blocks would always require a fixed amount of hashing power there could easily emerge disparities between cost of mining a block and a fixed reward granted for a block. There could emerge cases in which block reward is higher than the average cost of finding a nonce. Therefore, block creation would be accelerated, because new units of computational power would increase number of blocks mined but not diminish the reward per block. The fixed difficulty could also cause mining to be nonprofitable and hence to cease until, for exogenous reasons, price of computational power decreases or rewards appreciate. Hence, difficulty adjustment acts as a *market clearing mechanism*.

The difficulty level can be modeled in the following way. Let $i \in I = \{1, 2, ..., n\}$ be a node in the network, such that each node has an amount of hashing power [20] h_i. Following the characteristics of exponential distributions, node i's instantaneous probability of creating a block meeting the difficulty criterion is $x_i = \frac{h_i}{D}$, where D is the difficulty level. The protocol modifies regularly the difficulty level to keep the expected time T between blocks constant. We can define

$$T = \frac{1}{\sum_{i \in I} x_i} \tag{19}$$

Therefore,

$$T = \frac{1}{\sum_{i \in I} \frac{h_i}{D}} \tag{20}$$

[20] Hashing power is defined as the number of *nonces* per second a machine can calculate. It is related both to the machine's processing power and to the structure of its processor. Bitcoin and similar blockchain environments changed drastically with the introduction of Application-Specific Integrated Circuits (ASICs) designed specifically to maximize their hashing power's efficiency. ASICs are designed to perform only this task, and thus have substantially larger hashing power, typically ranging between 4 and 16 Terahash per second (TH/s). For comparison, an Intel Core i7 has a hashing power between 10 and 20 MH/s, i.e. around 1 million times less than an ASIC. The total hashrate of the Bitcoin network by the beginning of June 2018 was around 31 million TH/s.

The difficulty criterion is then defined by

$$D = T \sum_{i \in I} h_i \tag{21}$$

Hence, the difficulty increases as the total amount of hash $\sum_{i \in I} h_i$ increases; and miner i finds a *nonce* that meets the criterion following $\tilde{B} \sim Poisson(\frac{t}{D}\frac{h_i}{H})$. Therefore, without a loss of generality, in our model we can make a simplifying assumption that mining rewards are randomly assigned to a node (a miner or a pool) at every period. This allows us to avoid unnecessary complexity generated by explicitly modelling reward arrivals as Poisson process and difficulty adjustments.

Appendix D. Proofs

Proof of Lemma 5

Proof. From Assumption 3 it follows that an attacker must conclude the attack at some T. For a proof we need to demonstrate that while $B_{main} \geq K$ and $B_{attack} \leq B_{main}$ an attacker would be better off by paying $CH(B_{main,T-1}+1-B_{attack,T-1})(1-\frac{H_m}{H})$ with certainty than by paying $0 \leq \frac{H_a}{H}CH < (1-\frac{H_m}{H})CH$ with certainty and $CH(B_{main,T-1} - B_{attack,T-1})(1-\frac{H_m}{H})$ with probability $\frac{H_a+H_m}{H} < 1$ and $CH(B_{main,T-1} - B_{attack,T-1}+2)(1-\frac{H_m}{H})$ with probability $1-\frac{H_a+H_m}{H}$. If this holds, then by backward induction the attacker would be better off by acquiring all available hashing power immediately. These conditions yield the following inequality

$$\begin{aligned} &(B_{main,T-1}+1-B_{attack,T-1})\left(1-\frac{H_m}{H}\right)CH \leq \\ &\left\{\frac{H_a+H_m}{H}(B_{main,T-1}-B_{attack,T-1})\left(1-\frac{H_m}{H}\right)\right. \\ &+\frac{H_a}{H}+\left(1-\frac{H_a+H_m}{H}\right)(B_{main,T-1}+2-B_{attack,T-1})\left(1-\frac{H_m}{H}\right)\Big\}CH \end{aligned} \tag{22}$$

for $0 < \frac{H_a}{H} \leq (1 - \frac{H_m}{H})$,

A simple manipulation yields

$$0 < \left(1 - \frac{H_m}{H}\right)\left(1 - 2\frac{H_m + H_a}{H}\right) + \frac{H_a}{H} \tag{23}$$

Substituting $\frac{H_a}{H}$ with $1 - \frac{H_m}{H} - \eta$ where $0 < \eta \leq 1 - \frac{H_m}{H}$ yields

$$0 \leq \left(1 - 2\frac{H_m}{H}\right)\eta \tag{24}$$

which holds for $\frac{H_m}{H} < \frac{1}{2}$ as an inequality and for the special case $\frac{H_m}{H} = \frac{1}{2}$ as an equality. Implying that for $\frac{H_m}{H} < \frac{1}{2}$ an attacker strictly prefers ending the attack immediately and weakly for the special case.

■

Proof of Lemma 6

Proof. For the trivial case $B_{attack} > K$ it is self evident that $\frac{H_a}{H} > 0$ increases costs without any benefit.

To prove Lemma 6 it is necessary and sufficient to demonstrate that for all $B_{attack} \leq K$ and $B_{main} < K$ it holds that

$$\frac{H_a + H_m}{H}C_W + \left(1 - \frac{H_a + H_m}{H}\right)C_L + \frac{H_a}{H}CH > \frac{H_m}{H}C_W + \left(1 - \frac{H_m}{H}\right)C_L \tag{25}$$

where, C_W and C_L are expected cost of conducting a double spending attack after winning and losing one mining competition, respectively.

$$C_W = \sum_{a=0}^{K} \underbrace{\left(1 - \frac{H_m}{H}\right)^K \left(\frac{H_m}{H}\right)^a \binom{K+a}{a}}_{\text{Probability mass function}} \underbrace{\left(1 - \frac{H_m}{H}\right)(K + 1 - a - 1)CH}_{\text{Cost for each a}} \tag{26}$$

$$C_L = \sum_{a=0}^{K} \underbrace{\left(1 - \frac{H_m}{H}\right)^{K-1} \left(\frac{H_m}{H}\right)^{a} \binom{K+a-1}{a}}_{\text{Probability mass function}} \underbrace{\left(1 - \frac{H_m}{H}\right)(K+1-a)CH}_{\text{Cost for each a}} \tag{27}$$

Both of these are sums of probability mass functions for each positive cost $(1 - \frac{H_m}{H})(K - a)CH$ and $(1 - \frac{H_m}{H})(K - a + 1)CH$, respectively, where a is a number of blocks mined by the attacker before $B_{main} = K$.

Substituting C_W and C_L to Equation 25, rearranging and manipulating yields

$$CH > C_L - C_W = 1 > \sum_{a=0}^{K} \left(1 - \frac{H_m}{H}\right)^{K-1} \left(\frac{H_m}{H}\right)^{a} \binom{K-1+a}{a} \left(1 - \left(\frac{H_m}{H}\right)^{K+1-a}\right) \tag{28}$$

which holds because

$$Pr(a \leq K) = \sum_{a=0}^{K} \left(1 - \frac{H_m}{H}\right)^{K-1} \left(\frac{H_m}{H}\right)^{a} \binom{K-1+a}{a} \leq 1 \tag{29}$$

and, for any value $K + 1 - a < \infty$. [21]

$$\left(1 - \left(\frac{H_m}{H}\right)^{K+1-a}\right) < 1 \tag{30}$$

holds.

■

Proof of Theorem 8

Proof.

For cost of double spending to be larger for $0 < \frac{H_m}{H} \leq \frac{1}{2}$ than for $0 = \frac{H_m}{H}$ it must be the

[21] $K+1-a = \infty$ would imply infinite cost for conducting a double spending attack and an infinite waiting time for goods to be delivered.

case that

$$(1-f^c)R\left\{(K+1)-\frac{\left(1-2\frac{H_m}{H}\right)(K+1)+\binom{2K+2}{K+2}((1-\frac{H_m}{H})\frac{H_m}{H})^{K+2}\,{}_2F_1(2,2K+3;K+3,\frac{H_m}{H})}{1-\frac{H_m}{H}}\right\}$$
$$-\frac{f^cR}{1-\beta}\frac{H_m}{H}<0 \tag{31}$$

For every $0<\frac{H_m}{H}\leq\frac{1}{2}$ and $0<K$. Where $(1-f^c)R(K+1)$ is the cost of double spending for an attacker with $\frac{H_m}{H}=0$.

Simple rearrangement of Equation 31 yields

$$(1-f^c)\left\{(K+1)-\frac{(1-2\frac{H_m}{H})(K+1)+\binom{2K+2}{K+2}((1-\frac{H_m}{H})\frac{H_m}{H})^{K+2}\,{}_2F_1(2,2K+3;K+3,\frac{H_m}{H})}{1-\frac{H_m}{H}}\right\}\frac{H}{H_m}$$
$$<\frac{f^c}{1-\beta} \tag{32}$$

Taking a partial derivative w.r.t. $\frac{H_m}{H}$ of the left hand side of the Equation 32 and setting it to zero yields $\frac{H_m}{H}=\frac{1}{2}$. This and $f''(\frac{1}{2})<0$ imply that it is sufficient to analyze inequality at $\frac{H_m}{H}=\frac{1}{2}$; i.e. where the direct benefit from H_m is largest in comparison to opportunity costs. After analysing the Equality 32 in a case where $\frac{H_m}{H}=\frac{1}{2}$ and applying Gauss' Summation Theorem to the hypergeometric function and Legendre's relation and some manipulation and expressing hypergeometric function as a sum Equation 31 simplifies to the equation in Theorem 8. ■