Corruption is the Path to Security

Tymofiy Mylovanov [*]& Rakesh Vohra[†]

February 2018

Abstract

We propose a simple model of the blockchain to argue that high mining rewards are not needed to attract 'honest' miners and high entry costs are unable to deter 'dishonest' miners. While it is true that lowering the costs of being a miner encourages the entrance of corrupt miners it also gives incentives to honest agents to do the same. This competition prevents any one miner from securing a majority share of computing power. Therefore, rather than discouraging agents from being miners via costly money burning activities one should do the reverse.

1 Introduction

Not since sliced bread has a technology so captured the imagination as blockchain. It is a database which anyone can read. Its purpose is to ensure agreement among users about an underlying state that changes over time such as each agent's endowments of goods. Property records are an example. The resources needed to support a property database are provided by the local authority via taxes. The novelty of the blockchain is that it eschews dependence upon a trusted entity to manage the database. Why? An entity that charges for access to the database (as a way to cover the costs), will be tempted restrict access to the database in return for a higher price. If the entity is the Government, one must trust that it will not corrupt the database to further political ends. In both cases, the decision of an entity to abuse its position is the outcome of a trade-off between the present rewards of restricting access and the long run benefits of maintaining the database for all.

Given instances of ostensibly trusted intermediaries abusing their role, as well as their absence in some countries, it is natural to seek alternatives. The blockchain is an alternative that relies

[*] University of Pittsburgh, Kyiv School of Economics, and National Bank of Ukraine, mylovanov@gmail.com

[†] University of Pennsylvania, rvohra@seas.upenn.edu

on competition between intermediaries. Users of the blockchain platform coordinate on common software, that allows anyone access to the database as well as copy it. The software enables users to participate using pseudonyms if they wish. While identities are private, transactions are not.

Multiple copies of the database ensure access and reliability. However, maintaining consistency between independent copies when records are added to the database is a problem. Conflicts (called **forks**) can arise because of latency in the network or on purpose as a way to exploit counterparties via what is called a **double spend** attack (see section 2).

The blockchain limits conflicts by restricting the right to update the database. The most popular protocol employed, called proof of work (PoW), assigns the right to augment the database via a costly 'electricity burning' contest. The amounts consumed are staggering (Vries (2018)). Contest participants are called **miners** and the probability of winning is proportional to the amount of computational power held. As long as no miner holds more than 51% of all computational power, it is argued the probability of the double spending problem arising is vanishingly small (Lamport, Shostak and Pease (1982)). The winner of the contest is rewarded with 'tokens' whose value is linked to the integrity of the database. The blockchain also bars deletions from the database so as to allow anyone to verify the consistency of a new entry with previous entries in the database. This feature encourages competition by allowing anyone to branch off the existing database while retaining all the information contained in the original database. There is a protocol for resolving conflicts should they arise called the longest chain rule. Readers unfamiliar with the details should consult section 2.

The central question is whether the blockchain protocol provides sufficient incentives to prevent conflicts arising in the database. Biais et. al (2018) model the blockchain protocol as a stochastic game with multiple Markov perfect equilibria, some with no conflicts among the databases. Abadi and Brunnermeier (2018) argue that the fees paid to miners as well as the money burnt to secure the right to update the database are essential to maintaining validity of the database, and limits the value of transactions that will be executed on the blockchain. Budish (2018) argues that the flow payments to miners for running the blockchain must be large relative to the one-off, benefit from attacking it.

In this paper we propose a simple model that generates conclusions that run counter to those just summarized. In our model the optimal reward for mining is equal to zero, contradicting the intuition that high mining rewards are needed to attract 'honest' miners. In our model social welfare is *independent* of the level of mining and entry costs. This shows that high entry costs don't decrease social welfare because of wastefulness of the proof of work. Similarly, it establishes that high entry costs are unable to increase social welfare by deterring fraudulent miners.

This difference in conclusions arises from the fact that prior work has focused on the competition needed to make the database secure rather than the competition to 'capture the platform'. We illustrate this in section 4 where we introduce a model of imperfect competition between miners that is a counterpart to the perfect competition model of Budish (2018). In this model, unlike Budish (2018), there are two equilibria. In one, a single miner becomes dominant (secures a strict majority of all computational resources). In the other equilibrium, there is no dominant miner and the blockchain is secure. This suggest that a multiplicity of blockchain platforms and competition among them will help ensure that the blockchain in which participants coordinate on the good equilibrium, with no miner pre-eminent, will come to dominate the market.

1.1 Prior Literature

The economic literature on blockchain has focused on three kinds of questions. The first is on the economic implications of such a distributed database on traditional intermediaries. Examples are Catalini and Gans (2017) and Cong and He (2018), but are not relevant to this paper.

The second is whether the protocols furnish the incentives to prevent conflicts arising in the database. Easley, O'Hara, and Basu (2017) use a game-theoretic framework to analyze the emergence of transaction fees in Bitcoin and the implications of these fees for mining costs. The R&D race between miners is described in Ma, Gans and Tourky (2018), who argue that regulation of mining would reduce the overall costs of the system and improve welfare. Huberman, Moallemi, and Leshno (2017) study transaction fees and conclude that the blockchain market structure completely eliminates the rents that a monopolist would extract despite the fact that only one miner processes transactions at a time.

The third seeks to determine the optimal fee setting mechanisms for miners who secure the blockchain (see, for example, Basu *et al* (2019), Buterin (2018) and Lavi *et al* (2017)). In this paper, we argue that it is optimal to set the fees to zero.

2 How Blockchain Works

Users of the blockchain platform coordinate on a common software that allows anyone access to the database as well as copy it. Users may participate using pseudonyms if they wish but all transactions are public. The blockchain bars deletions from the database so as to allow anyone to verify the consistency of a new entry with previous entries in the database. When a conflict arises, a protocol for resolving it is needed. The blockchain must also generate incentives for users to furnish the resources needed to support the database. We outline how the blockchain does this with an example.

Suppose a Mr. A who wishes to sell a property to a Mr. B. For the sale to be consummated, Mr. A must prove he owns the property and Mr. B must prove he has the cash. Assuming a single database, it suffices for Mr. A and Mr. B to identify the entry in the database that records these facts. Entries in the database are listed in the order in which they were added to the database. Each new entry should point back to the previous entry in the database. Hence, the term **chain**. Assuming the pointer is set correctly, one can, starting with the most recent entry, backtrack through the database and recreate the history of all transactions executed prior to the most recent entry. Ensuring the pointer is set correctly is crucial.

If each party believes in the integrity of the database, the sale is executed and a new entry to the database is added which records the transfer of the property to Mr. B and a transfer of money to Mr. A. Multiple copies of the database are an obstacle to execution if in some copy there is no record showing that Mr. A is the owner of the property. We outline how this can happen and the protocol employed to resolve the conflict.

Suppose all extant copies of the database are initially identical. Mr. A and B can verify to their satisfaction that each has the endowment they claim. They write a tentative contract of sale that is added to a pool containing other tentative contracts written by others. Cryptographic schemes prevent either party unilaterally adding a tentative contract to the pool. The tentative contract is finalized only when it is recorded in the database.

At intervals, a user is selected to augment the database. Whoever secures this right may select any tentative contract from the pool and add it to the database.[1] By convention each tentative contract added should point back to the last item in the database.

Why might an agent selected to add a tentative contract to the database not point back to the last item in the database? Suppose the database initially consists of three items. Item #1 records the fact that Mr. A owns the property. Item #2, which points back to item #1, records the cash holdings of Mr. B. Item #3, which points back to item #2, records the cash holdings of a Mr. C:

$$1 \leftarrow 2 \leftarrow 3.$$

Now, there are two provisional contracts in the pool. One in which Mr. A sells his property to Mr. B and another in which the same property is sold to Mr. C. Given sufficient time, Mr. B and Mr C might discover this by searching the list of provisional contracts in the pool. This requires agents to continuously monitor the pool which is impractical.

Suppose in the first time instant, Mr. A wins the right to augment the database. Mr A selects the provisional contract involving himself and Mr. C from the pool and adds it to the database as item #4 and it points back to item #3. The updated database is not communicated instantaneously to all. The asynchronous updates of all copies of the database means that it is possible that Mr. B will be unaware of it for a period. During this period, suppose Mr. A is again allocated the right to update the database. Mr. A now has an incentive to add the contract with Mr. B as item #5. However, Mr. A could set the pointer to item #3 instead. This causes a fork in the database generating two paths. One is

$$1 \leftarrow 2 \leftarrow 3 \leftarrow 4,$$

and the other is

$$1 \leftarrow 2 \leftarrow 3 \leftarrow 5.$$

In one path Mr. C is the owner of the property and in another it is Mr. B. Mr. A has succeeded in selling the property twice. This is called the **double spending problem**.

[1] In practice they select a prescribed number of contracts, called a **block**. For purposes of exposition we assume that only a single tentative contract is selected.

The blockchain deals with forks in three ways. First, to avoid causing them, the rule for allocating the right to append a provisional contract to the database is designed so as to reduce the chance that an interested party secures it twice in a row. One way to do this is to select a user at random. With a large number of users, the chance that one selects a user with a vested interest is small. The anonymity feature of the Blockchain means that we don't know how many distinct users on the platform there are. One can track accounts, but a single agent can set up multiple distinct accounts.

The generally accepted fix, called **proof of work**, allocates the right to augment the database via a contest which requires the solution of a mathematical puzzle that is designed so that it can only be solved by guessing and checking (see Nakamoto (2008)).[2] The greater one's computing power, the greater the number of guesses that can be processed per unit of time. In fact, the probability of being first to solve the puzzle is proportional to a measure of one's computational power. Computational power is expensive as it requires electricity and increasingly specialized chips. This encourages agents to consolidate their computational resources. Agents who make these investments are called **miners**. The difficulty of the puzzle scales with *total* computational resources so as to ensure that the time it takes to solve the puzzle remains constant (currently ten minutes) and that the chance that two or more miners solve it simultaneously is close to zero. Hence, at most one miner at any time can augment the database.

To encourage users to become miners, the platform rewards the first miner to solve the problem with tokens whose value is tied to the integrity of the database. Assuming computational power is sufficiently dispersed, the chance that the same agent is selected twice in a row to augment the database is negligible. Sufficiently dispersed is generally taken to mean that no agent controls more than 51% of total computational resources (see Nakamoto (2008)).

The second way to deal with forks is to induce delay so as to allow time for additions to the database to be propagated. Thus, provisional contracts are not executed upon addition to the database. Instead, it is customary to delay until 4 or 5 additional items (or blocks) have been added to the database without forks.

The third way is a protocol for resolving forks. In the presence of a fork, a miner selected

[2] An alternative is called proof of stake, see Saleh (2018).

to augment the database, is supposed to point to the last item in the longest of the two paths. In our example above, it will mean that eventually all transactions that point back to 3. The transactions on the shorter path are deemed never to have occurred, i.e., orphaned. To encourage miners to comply, the tokens they receive are linked to the path which was updated by the miner. Their value relies on the willingness of others to accept tokens in exchange for goods and services. If the path that was updated subsequently becomes orphaned, then, others may not accept the associated tokens.

Abadie and Brunnermeier (2018) point out that forks combined with the no deletion property enables any subset of users to replicate information on the existing database and 'fork off' into an alternative with a different rules should they wish. This serves to discipline miners.

In the next section we introduce a stylized model to investigate whether the rules of the blockchain have the desired effect.

3 A Model

As before, suppose Mr. A wishes to sell a property to Mr. B. For the sale to be consummated, Mr. A must prove that he owns the property and Mr. B must prove that he has the cash. Suppose the property in question had originally been owned by a Mr. C and this is agreed upon by all parties. Suppose also that Mr. A did not acquire the property from Mr. C. Instead, Mr. A simply appended to his copy of the database a record that he acquired it. Recall, anyone can hold a copy of the database, so Mr. A possesses one. Clearly, Mr. A has an incentive to modify his copy of the database to his advantage and one can forestall that by ignoring Mr. A's copy of the database. By the same token, one should ignore Mr. B's copy of the database. More generally, one should ignore copies of the database owned by interested parties. But, what is to prevent an individual maintaining multiple copies of the database under different names? One could rely on a trusted party to certify identities, but, the entire premise of the blockchain is to minimize dependence on such trusted parties! It does so by requiring individuals to compete for the right to verify the transaction. We introduce a simple model to examine this claim.

Suppose $2n$ honest agents and $2k$ dishonest agents. Dishonest agents seek to engage in double spending. Randomly pair up all the agents to trade. There will be $n + k$ such pairs. If a matched pair choose not to trade, each earns a payoff of zero. Should they choose to trade, the payoffs earned are as follows:

1. If both parties are dishonest, each earns zero.
2. If both parties are honest, each earns w.
3. If one is honest and the other not, the honest agent earns $-v$ while the dishonest agent earns u with $v > u$.

The matched pair can choose to trade without verifying each others endowments or they can trade contingent upon verification of their respective endowments. Verification is performed by agents called miners. Any agent can choose to be a miner. A miner is expected to check the endowment claims of the counter-parties and, if the trade is executed, update the database. To avoid conflicts, only one miner is permitted to update the database. This makes selecting multiple miners superfluous as only one of them can eventually update the database and it is the incentives of this miner that the counter-parties should care about.

The costs of maintaining a single database of prior transactions as well as augmenting it is denoted C. Call each such database an account. Any agent, honest or otherwise, can establish multiple accounts as long as they pay C for each account. In reality these costs correspond to the costs of acquiring servers and specialized chips and need not be linear with the number of accounts. Our point is this is unnecessary.

Given C, verification is offered for a fee $2p^*$ that will be divided equally between the counter-parties. An agent not a counter-party to the trade that is selected as miner, will correctly determine if the counter-parties are honest or not. In the event that at least one counter-party is discovered to be dishonest, the trade is not executed.

When a pair is matched, they can choose to solicit a miner. If p^* is sufficiently small, honest agents will choose to trade contingent upon verification by an agent not a counter-party to the transaction. Dishonest agents, of course, would rather not. If they decline this would signal they are dishonest. Hence, we assume that each counter-party will agree to verification by an agent not a counter-party to the transaction. In the absence of trusted parties to verify identities it is impossible to distinguish between a miner unrelated to any of the agents from one that is not.

Therefore, each dishonest agent has an incentive to offer verification services that will certify itself as having the relevant endowment.

It is immediate that conditioning transfers to miners based on their reported independence is not incentive compatible. Therefore, transfers must ether be constant or contingent upon the outcome of trade. We ignore contingent transfers as this requires verifying trade outcomes ex-post. Assuming a constant transfer, $2p^*$, there is no way to distinguish between miners, hence one of the miner accounts should be chosen at random to verify the transaction.

Each agent, honest or otherwise, may act as a miner. In fact, they have an incentive to establish multiple accounts so as to increase the chance that they will be selected. Let μ_D be the number of miner accounts held by a dishonest agent and μ_H the number of miner accounts held by an honest agent. Note, we are assuming two agents of the same type will choose the same number of miner accounts. If $\alpha = \frac{2n}{2(n+k)}$, the proportion of honest agents, the expected payoff to an honest agent, denoted V_H is:

$$\alpha w - (1-\alpha)\frac{\mu_D v}{2n\mu_H + 2k\mu_D} - p^* + \frac{(n+k)\mu_H}{2n\mu_H + 2k\mu_D}2p^* - \mu_H C. \tag{1}$$

The first three terms in (1) represent the expected gains from trade. The remaining terms represent the net expected profits from acting as a miner.

The expected payoff to a dishonest agent, denoted V_D, is

$$\frac{\alpha u \mu_D}{2n\mu_H + 2k\mu_D} - p^* + \frac{(n+k)\mu_D}{2n\mu_H + 2k\mu_D}2p^* - \mu_D C. \tag{2}$$

Our first is result shows that in equilibrium no miner holds more than half of all accounts. In fact, the largest coalition of dishonest miners holds at most half of all verification accounts.

Theorem 3.1 *The equilibrium number of verification accounts held by honest agents exceeds that of dishonest agents. Furthermore, as p^* increases, the share of verification accounts held by dishonest agents increases.*

Proof: We derive the value of μ_H that maximizes V_H, holding μ_D fixed. The relevant first order condition is

$$(1-\alpha)\frac{2n\mu_D v}{(2n\mu_H+2k\mu_D)^2}+\frac{p^*(n+k)}{n\mu_H+k\mu_D}-\frac{n(n+k)\mu_H p^*}{(n\mu_H+k\mu_D)^2}-C=0.$$

It is straightforward to check that the second order condition holds.

$$\Rightarrow\ \frac{(1-\alpha)n\mu_D v}{2}+(n+k)p^*(n\mu_H+k\mu_D)-n(n+k)\mu_H p^*-C(n\mu_H+k\mu_D)^2=0$$

$$\frac{(1-\alpha)n\mu_D v}{2}+(n+k)p^*k\mu_D=C(n\mu_H+k\mu_D)^2$$

$$\frac{kn\mu_D v}{2(n+k)}+(n+k)p^*k\mu_D=C(n\mu_H+k\mu_D)^2$$

After some manipulations we obtain

$$k\mu_D[\frac{nv}{2(n+k)}+p^*(n+k)]=C(n\mu_H+k\mu_D)^2. \tag{3}$$

The relevant first order condition for dishonest agents is:

$$\frac{\alpha u}{2n\mu_H+2k\mu_D}-\frac{2k\alpha u\mu_D}{(2n\mu_H+2k\mu_D)^2}+\frac{p^*(n+k)}{n\mu_H+k\mu_D}-\frac{kp^*(n+k)\mu_D}{(n\mu_H+k\mu_D)^2}-C=0.$$

Simplification yields

$$\frac{\alpha u(n\mu_H+k\mu_D)}{2}-\frac{k\alpha u\mu_D}{2}+p^*(n+k)(n\mu_H+k\mu_D)-kp^*(n+k)\mu_D=C(n\mu_H+k\mu_D)^2$$

$$\frac{\alpha un\mu_H}{2}+p^*(n+k)n\mu_H=C(n\mu_H+k\mu_D)^2$$

$$n\mu_H[\frac{nu}{2(n+k)}+p^*(n+k)]=C(n\mu_H+k\mu_D)^2. \tag{4}$$

Dividing (4) by (3) produces:

$$\frac{n\mu_H}{k\mu_D}=\frac{2p^*(n+k)^2+nv}{2p^*(n+k)^2+nu}>1, \tag{5}$$

as $v>u$. Hence, the number of verification accounts held by honest agents exceeds that of dishonest agents. Notice also that as p^* increases, the ratio $\frac{n\mu_H}{k\mu_D}$ decreases, i.e. the share of verification accounts held by dishonest agents increases. ■

The conclusion of Theorem 3.1 relies on the marginal cost of a verification account being constant. It is not true otherwise. Were verification costs to grow quadratically with number of accounts, say, the conclusion of Theorem 3.1 does not hold. This suggests that raising the cost of holding multiple verification accounts is counterproductive. Instead, one should encourage many agents to hold multiple verification accounts, i.e., become miners.

It is interesting to contrast the conclusion of Theorem 3.1 with Budish (2018). In that paper it is argued that one must increase p^* to discourage a dishonest agent from setting up multiple verification accounts. The difference arises because we allow multiple agents to compete to secure the right to be selected as miners. An increase in p^* encourages dishonest agents to hold relatively more accounts. Hence one should lower the fee paid for verification.

The next result shows that the number of miner accounts decreases as C, the cost of setting up such an account increases. This is what one would expect.

Theorem 3.2 *In equilibrium, the total number of miner accounts is*

$$\frac{1}{C(n+k)[\frac{1}{nu+2(n+k)^2p^*}+\frac{1}{nv+2(n+k)^2p^*}]}.$$

Proof: Equation (3) can be rewritten as

$$Ck\mu_D(\frac{n\mu_H}{k\mu_D}+1)^2=\frac{2p^*(n+k)^2+nv}{2(n+k)}$$

$$\Rightarrow\ Ck\mu_D(\frac{2p^*(n+k)^2+nv}{2p^*(n+k)^2+nu}+1)^2=\frac{2p^*(n+k)^2+nv}{2(n+k)}$$

$$\Rightarrow\ k\mu_D=C^{-1}(\frac{2p^*(n+k)^2+nv}{2p^*(n+k)^2+nu}+1)^{-2}(\frac{2p^*(n+k)^2+nv}{2(n+k)}).$$

Similarly, (4) can be rewritten as,

$$n\mu_H=C^{-1}(\frac{2p^*(n+k)^2+nu}{2p^*(n+k)^2+nv}+1)^{-2}(\frac{2p^*(n+k)^2+nu}{2(n+k)}).$$

Therefore,

$$n\mu_H+k\mu_D = C^{-1}[(\frac{2p^*(n+k)^2+nv}{2p^*(n+k)^2+nu}+1)^{-2}(\frac{2p^*(n+k)^2+nv}{2(n+k)})+(\frac{2p^*(n+k)^2+nu}{2p^*(n+k)^2+nv}+1)^{-2}(\frac{2p^*(n+k)^2+nu}{2(n+k)})]$$

$$= \frac{C^{-1}(2p^*(n+k)^2+nu)(2p^*(n+k)^2+nv)}{2(n+k)(4p^*(n+k)^2+nv+nu)}$$

$$= \frac{1}{C(n+k)[\frac{1}{nu+2(n+k)^2p^*}+\frac{1}{nv+2(n+k)^2p^*}]}.$$

■

Next, we show that total welfare is *independent* of C.

Theorem 3.3 *In equilibrium, total welfare is*

$$\frac{n^2w}{n+k} - \frac{1}{(n+k)[\frac{1}{nu+2(n+k)^2p^*}+\frac{1}{nv+2(n+k)^2p^*}]}(1+\frac{n(v-u)}{nv+2(n+k)p^*}).$$

Proof: Total welfare is $nV_H + kV_D$ which is

$$-C(n\mu_H+k\mu_D)+n\alpha w - \frac{n(1-\alpha)(v-u)\mu_D}{n\mu_H+k\mu_D}$$

$$= \frac{n^2w}{n+k} - \frac{1}{(n+k)[\frac{1}{nu+2(n+k)^2p^*}+\frac{1}{nv+2(n+k)^2p^*}]}(1+\frac{n(v-u)}{nv+2(n+k)p^*}).$$

■

Theorem 3.3 runs counter to the intuition that a high cost for setting up a verification account is needed to discourage dishonest agents from corrupting the database. This is a consequence of the assumption that verification costs grow linearly with the number of accounts but that strengthens the point made.

Bertrand competition between miners will cause p^* to decline to zero.[3] Theorem 3.3 shows that this results in an increase in welfare. When $p^*=0$, total welfare is

$$\frac{n^2w}{n+k} - \frac{n^2uv}{(n+k)(nv+nu)}(1+\frac{n(v-u)}{nv}) = \frac{n}{n+k}[nw - \frac{u(2v-u)}{v+u}].$$

[3] Reflection will show that agents cannot use the choice of verification fee to signal their type.

It is straightforward to check that total welfare is increasing in u and decreasing in v as one might expect.

When $p^* = 0$ the total number of verification accounts is

$$\frac{1}{C(n+k)[\frac{1}{nu+2(n+k)^2p^*} + \frac{1}{nv+2(n+k)^2p^*}]} = \frac{1}{C(n+k)[\frac{1}{nu} + \frac{1}{nv}]} = \frac{nuv}{C(n+k)(u+v)}.$$

Theorem 3.4 *When $p^* = 0$ the equilibrium expected profit of an honest agent is*

$$(n+k)^{-1}[nw - \frac{vu}{2(v+u)}(1 + \frac{v}{v+u})].$$

The equilibrium expected profit of a dishonest agent is

$$\frac{nu^3}{2k(n+k)(v+u)^2}.$$

Proof: Substituting $p^* = 0$ into (5) yields:

$$\frac{n\mu_H}{k\mu_D} = \frac{v}{u} \;\Rightarrow\; n\mu_H = (\frac{v}{u})k\mu_D \;\Rightarrow n\mu_H + k\mu_D = (\frac{v}{u} + 1)k\mu_D. \tag{6}$$

Now, focus on honest agents. Recall (3):

$$k\mu_D[\frac{nv}{2(n+k)} + p^*(n+k)] = C(n\mu_H + k\mu_D)^2.$$

Substituting $p^* = 0$ and (6) into (3) yields:

$$k[\frac{nv}{2(n+k)}] = C((\frac{v}{u} + 1)k)^2\mu_D \;\Rightarrow\; \mu_D = C^{-1}\frac{nvu^2}{2k(n+k)(v+u)^2}.$$

Turning to dishonest agents, recall (4):

$$n\mu_H[\frac{nu}{2(n+k)} + p^*(n+k)] = C(n\mu_H + k\mu_D)^2.$$

Substituting $p^* = 0$ and (6) into (4):

$$n\mu_H[\frac{nu}{2(n+k)}] = C(n\mu_H + k\mu_D)^2 = C(\frac{v}{u}+1)^2k^2\mu_D^2 = C^{-1}(v+u)^2(\frac{k}{u})^2\frac{(nvu^2)^2}{4k^2(n+k)^2(v+u)^4}$$

$$\Rightarrow\ n\mu_H[\frac{nu}{2(n+k)}] = C^{-1}(v+u)^2\frac{(nvu)^2}{4(n+k)^2(v+u)^4}$$

$$\Rightarrow n^2u\mu_H = C^{-1}\frac{(nvu)^2}{2(n+k)(v+u)^2}$$

$$\Rightarrow \mu_H = C^{-1}\frac{uv^2}{2(n+k)(v+u)^2}$$

The expected profit for an honest agent is

$$\alpha w - (1-\alpha)\frac{\mu_D v}{2n\mu_H + 2k\mu_D} - \mu_H C.$$

Substituting into the profit expression:

$$\alpha w - (1-\alpha)\frac{v}{2(\frac{v}{u}+1)k} - \mu_H C = \alpha w - (1-\alpha)\frac{v}{2(\frac{v}{u}+1)k} - \frac{uv^2}{2(n+k)(v+u)^2}$$

$$= \frac{nw}{n+k} - \frac{vu}{2(v+u)(n+k)} - \frac{uv^2}{2(n+k)(v+u)^2}$$

$$= (n+k)^{-1}[nw - \frac{vu}{2(v+u)} - \frac{uv^2}{2(v+u)^2}]$$

$$= (n+k)^{-1}[nw - \frac{vu}{2(v+u)}(1+\frac{v}{v+u})].$$

The payoff to a dishonest agent is:

$$\frac{\alpha u\mu_D}{2n\mu_H + 2k\mu_D} - \mu_D C = \frac{\alpha u}{2(\frac{v}{u}+1)k} - \mu_D C$$

$$= \frac{\alpha u^2}{2(v+u)k} - \frac{nvu^2}{2k(n+k)(v+u)^2} = \frac{nu^2}{2(v+u)k(n+k)} - \frac{nvu^2}{2k(n+k)(v+u)^2}$$

$$= \frac{nu^2}{2k(n+k)(v+u)}[1 - \frac{v}{v+u}] = \frac{nu^3}{2k(n+k)(v+u)^2}.$$

■

If there was free entry of agents until the point of zero profit, the number of dishonest agents has to grow to infinity. To understand the impact on welfare we also need to specify how number of honest agents grows.

Suppose first that k and n grow such that $\frac{n^2}{k}$ goes to zero, i.e., the number of dishonest agents grows much faster than the number of honest agents. From Theorem 3.3, we see that the expression for welfare tends to $-\infty$ no matter the value of p^*. This means that at some point agents are better off opting out the platform, i.e., there is no trade. If k and n grow such that $\frac{n}{k}$ goes to zero. Thus, the number of dishonest agents still grows faster than the number of honest agents. The expression for welfare in Theorem 3.3 converges to zero.

Now suppose that k and n grow such that $\frac{k}{n}$ goes to zero. Thus, the number of honest agents grows faster than the number of dishonest agents. In this case, when $p^* > 0$, total welfare grows with n provided w is sufficiently large. However, if $p^* = 0$, total welfare grows with n as long as $w > 0$. This emphasizes the point that a large verification fee, i.e. $p^* > 0$, has negative effects. If k and n grow at the same rate, total welfare grows with n. This highlights the fact that for the platform to be economically viable it must attract honest agents at least as high a rate than dishonest ones.

4 Comparison with Budish (2018)

Given that the conclusions we arrive at differ from Budish (2018), say, in this section we pinpoint the source of the difference. Budish (2018) argues that there is an incentive for a miner to capture a strict majority of computational power. They could deploy their dominant position to exploit counterparties. They could also choose to maintain the validity of the database but charge a premium for doing so. In our view this analysis does not account for the competition between miners to secure more than half of all computational power. To illustrate we introduce a model inspired by Budish (2018).

Suppose two miners each simultaneously choosing a quantity of computing power. Denote the amount of computing power chosen by miner $i = 1, 2$ to be q_i. If $q_1 > 0.5(q_1 + q_2)$, then, agent 1's payoff is $Vq_1/(q_1+q_2) - Cq_1$, where V is the value earned by miner 1 from double spending and C is the marginal cost of computing power. If not, miner 1's payoff is $q_1 F/(q_1+q_2) - Cq_1$, where F is the fee from adding a block. Assume $V > F$. The same will be true of miner 2's payoff.

If there is a symmetric equilibria, then, no miner has a strict majority of computational power. In this case no miner can engage in double spending. However, there is an asymmetric equilibrium. Fix miner 2 at q_2. Let us look at miner 1's best response. We will be interested in conditions under which miner 1's best response is no larger than q_2. Then, miner 1's expected payoff is $Fq_1/(q_1+q_2) - Cq_1$. This is the profit miner 1 earns assuming that he does not have strictly more than half of all computational power. The first order condition for optimality is:

$$\frac{F}{q_1+q_2} - \frac{Fq_1}{(q_1+q_2)^2} - C = 0 \;\Rightarrow\; \frac{Fq_2}{(q_1+q_2)^2} - C = 0.$$

Therefore,

$$(q_1+q_2)^2 = \frac{Fq_2}{C} \tag{7}$$

A similar argument with F replaced by V for miner 2, tells us that

$$(q_1+q_2)^2 = \frac{Vq_1}{C} \tag{8}$$

Dividing one by the other yields $q_1 = \frac{Fq_2}{V}$. If $V > F$ it follows that $q_1 < q_2$, i.e. an asymmetric equilibrium exists. Thus, if the rewards from mining are smaller than the rewards from double spending, there is an incentive for a miner to secure a strict majority of computational power. This is precisely the claim in Budish (2018), but, as our analysis makes clear, it relies on the selection of a particular equilibrium.

It is generally believed that the marginal costs of computation are increasing. A simple way to account for this is to make the costs in the model quadratic in computational power. In this case there is *no* pure strategy asymmetric equilibrium.

5 Conclusion

Trade presumes no uncertainty about the ownership of endowments being exchanged. The blockchain avoids dependence on a centralized authority to maintain consensus about the distribution of endowments. Instead, it uses a software embodied protocol. This paper argues that the incentives needed to maintain consensus can be reduced if we encourage those who directly benefit from trade to shoulder the burden.

References

Abadi, Joseph, and Markus K. Brunnermeier. ‘Blockchain Economics,’ Working Paper, 2018.

Basu, Soumya, David Easley, Maureen O’Hara, and Emin Sirer. 2019. ‘Towards a Functional Fee Market for Cryptocurrencies,’ retrieved from https://ssrn.com/abstract=3318327.

Biais, Bruno, Christophe Bisire, Matthieu Bouvard, and Catherine Casamatta. 2017.‘The Blockchain Folk Theorem,’ TSE Working Paper No. 17-817. Revised January 4, 2018.

Buterin, Vitalik. 2018.‘Blockchain Resource Pricing,’ retrieved from https://ethresear.ch/uploads/default/original

Catalini, Christian and Gans, Joshua S. 2017.‘Some Simple Economics of the Blockchain,’ Rotman School of Management Working Paper.

Cong, Lin and He, Zhiguo. 2018.‘Blockchain Disruption and Smart Contracts,’ Working Paper.

Easley, David, Maureen O’Hara, and Soumya Basu. 2017. ‘From Mining to Markets: The Evolution of Bitcoin Transaction Fees,’ Cornell University Working Paper.

Huberman, Gur, Jacob D. Leshno, and Ciamac C. Moallemi. 2017. ‘Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System, ’ Columbia Business School Research Paper No. 17-92.

Lavi, Ron, Or Sattath, and Aviv Zohar. 2017.‘Redesigning Bitcoin’s Fee Market,’ retrieved from https://arxiv.org/abs/1709.08881.

Nakamoto, Satoshi. 2008.‘Bitcoin: A Peer-to-Peer Electronic Cash System,’ retrieved from https://bitcoin.org/bitcoin.pdf.

Lamport L., R. Shostak, and M. Pease.1982.‘The Byzantine Generals Problem,’ **ACM Transactions on Programming Languages and Systems**, 4(3), 382401.

Ma, J. Joshua S. Gans and Rabee Tourky. 2018. ‘Market Structure in Bitcoin Mining,’ NBER Working Paper 24242.

Saleh, Fahad. 2018.‘Blockchain Without Waste: Proof-of-Stake,’ Working Paper.

Vries, Alex De. 2018.‘Bitcoin’s Growing Energy Problem,’ **Joule**, 2(5): 801805.