NBER WORKING PAPER SERIES

"ZERO COST" MAJORITY ATTACKS ON PERMISSIONLESS BLOCKCHAINS

Joshua S. Gans
Hanna Halaburda

Working Paper 31473
http://www.nber.org/papers/w31473

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
July 2023

All correspondence to joshua.gans@utoronto.ca. The latest version of this paper is available at joshuagans.com. Thanks to seminar participants at the a16z Crypto Lab and Eric Budish for useful discussions. All errors remain our own. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

At least one co-author has disclosed additional relationships of potential relevance for this research. Further information is available online at http://www.nber.org/papers/w31473

"Zero Cost" Majority Attacks on Permissionless Blockchains
Joshua S. Gans and Hanna Halaburda
NBER Working Paper No. 31473
July 2023
JEL No. D42,D82,E42

ABSTRACT

Permissionless blockchains were constructed with a view to being sustainably secure. At the heart of blockchain consensus mechanisms was an explicit cost (whether it be work or stake) for participation in the network and the opportunity to propose blocks that would be added to the blockchain. A key rationale for that cost was to make attacks on the network, which could be theoretically carried out if a majority of nodes were controlled by a single entity, too expensive to be worthwhile. Here we demonstrate that a majority attacker can successfully attack with a negative cost, which shows that explicit participation requirements do not necessarily result in a sustainably secure network. This suggests that any benefits of an attack that drive sustainable security are regulated from outside the network itself.

Joshua S. Gans
Rotman School of Management
University of Toronto
105 St. George Street
Toronto ON M5S 3E6
and NBER
joshua.gans@rotman.utoronto.ca

Hanna Halaburda
Stern School of Business
New York University
44 W 4th St.
New York, NY 10012
hhalaburda@gmail.com

1 Introduction

Permissionless blockchains were designed to be self-sustaining networks operating without the support of legal institutions and with the goal of allowing for open participation and censorship resistance. To achieve this, the consensus achieved across many nodes must be resistant to attacks by malicious agents. Consequently, to participate as a validating node that can propose new information on the blockchain, agents are required to incur participation costs. The general idea is that any attacker would have to incur similar costs to co-opt the network to any malicious end. Thus, the magnitude of the costs faced by attackers is a measure of the security or resilience of the network being a key component of any "no attack incentive constraint."

Budish (2022) has recently highlighted what might potentially be a fundamental conflict between participation constraints faced by honest nodes and incentive constraints intended to deter attackers. In order for nodes to participate in a network, they cannot expect to earn losses; that is, any rewards compensating nodes for their participation must (weakly) exceed expected participation costs. At the same time, because a permissionless blockchain protocol does not make distinctions based on the (unobserved) motivations of node operators, while an attacker may face costs in taking control of a network, they are able to access the very same rewards that offset those costs for honest nodes. Thus, Budish (2022) argues that, in a frictionless setting, the net costs incurred by a majority attacker are zero.[1] Suffice it to say in such situations, blockchains would be open to attacks and have effectively zero resilience.

Budish focuses on permissionless proof of work blockchains and identifies a number of frictions that may create positive net attack costs. Bakos and Halaburda (2021) identify a distinct set of frictions and also consider proof of stake blockchains. Our purpose here is to focus squarely on the within-protocol effects to understand the nature of the costs imposed on an attacker by the protocol and how these might change with other factors of the environment. Thus, we abstract away from external forms of regulation such as anticipated changes in exchange rates following an attack or the enforcement of contractual compliance including payment.

Our paper finds a stronger concern than that raised by Budish (2022). We find that the net costs for an attacker may be *less than zero*. While the block rewards in the form of newly minted tokens do not change during the attack, the total transaction fees may change if the blocks themselves are capacity constrained. Thus, because transactions are presumed to arrive at the same rate regardless of whether an attack is underway or not,

[1]While Budish focuses on an attacker who has access to majority of hashing power, Chiu and Koeppl (2022) develop a similar "no double spending incentive condition" for the case of minority attacker.

there is reduced total block capacity on attacking chains and hence, the fees of transactions that are confirmed are higher than the attacker would have received by processing blocks in the absence of an attack. Thus, the net costs of an attack can be positive and hence, there is an incentive for an agent who controls more than half of the hash power to attack even if there are no external benefits to an attacker. This suggests that costs incurred by a potential attacker are not securing the blockchain at all.

The paper proceeds as follows. The next section sets up the model by looking at the outcome in the absence of a majority attack on a permissionless Proof of Work blockchain. The model is more general than previous treatments as it relies on general cost functions that may differ between miners. Section 3 describes the attack leading to the main contribution of the paper in Section 4 where the costs of a majority attack are examined in close detail. Section 5 then highlights the role of transaction fees that have been, thusfar, neglected in the blockchain security literature.[2] Section 6 then reviews some dynamic considerations while a final section concludes.

2 Outcome without an Attack

We examine a Proof of Work protocol.[3] Following Ma et al. (2018) and Biais et al. (2019), let h_i denote the resource allocation by miner i to a blockchain, where $i \in \mathcal{M}$. The cost of providing h_i is $c_i(h_i)$, a non-decreasing, (weakly) quasi-convex function. Let H denote the total hash power in any given period, i.e., $H = \sum_{i \in \mathcal{M}} h_i$, where we drop time subscripts for convenience. In this environment, if the time it takes miner i to solve the required computational puzzle is y_i, then this time is a random variable with an exponential distribution with parameter $\frac{h_I}{D}$ where D parameterizes the difficulty of the computational puzzle. A block is mined by the first miner to solve the puzzle which happens at time $Y = \min\{y_1, y_2, ...\}$. By the properties of exponential distributions, Y also follows an exponential distribution — with parameter $\frac{1}{D}H$. Difficulty (D) is adjusted periodically (in Bitcoin every two weeks) so that, on average, a block is mined every τ periods. That is, $\tau = \frac{1}{H/D}$ or $D = \tau H$. Initially, we will assume that all agents take D as fixed.[4]

[2]Huberman et al. (2021) do examine the determination of fees but only under the assumption that there are no attacks in progress.

[3]For an overview of Proof of Work and Proof of Stake protocols, see e.g., Halaburda et al. (2022a) and Halaburda et al. (2022b).

[4]Another way of expressing this outcome is that, if a node contributes resources, then there is a probability, p_i, they will be selected as a leader to propose, and perhaps confirm, a block of transactions at a given point in time. Leshno and Strack (2020) show that when the selection probability, p_i, is equal to $\frac{h_i}{H}$, then this satisfies properties such as anonymity, Sybil resistance and zero returns to merging. That same selection probability is what is explicitly used in proof of stake protocols and the implied probability in proof of work protocols.

If a miner wins the contest and proposes a valid block, they receive a reward or newly minted tokens with the expected value in fiat currency terms of R. In addition, the proposer receives transaction fees paid by users to confirm transactions in the block with an expected fiat currency value of Φ.

All of this implies that miner i expects to mine $\frac{h_i}{\tau H}$ blocks per unit of clock time. Thus, in each period, i's expected profits are:

$$\frac{h_i}{\tau H}(R+\Phi) - c_i(h_i)$$

Let h_i^* be the hash power that maximizes these expected profits. Then the *participation constraint* for miner i is:

$$\frac{h_i^*}{\tau H}(R+\Phi) \geq c_i(h_i^*)$$

Thus, if $\mathcal{M}' \equiv \{i \in \mathcal{M} | \frac{h_i^*}{\tau H}(R+\Phi) \geq c_i(h_i^*)\}$, then $\mathcal{M}'$ is the set of active miners in a period for given τ, R and Φ. Note that if D remains fixed, if, ceteris paribus, there is an increase in H, then τ will fall and the cardinality of $\mathcal{M}'$ will be weakly smaller.

3 Description of an Attack

It has been long understood that Proof of Work protocols are vulnerable to a majority attack that can be implemented by a miner who controls a majority of the hash power. As noted earlier, Nakamoto (2008), and the literature that follows had presumed that the within-protocol costs associated with acquiring and applying hash power to carry out that attack would create a barrier to such attacks. Here we describe such attacks as their nature is critical to the determination of attack costs in the following section.

There are two extreme versions of a majority attack: an *outside* or an *inside* attack. Budish (2022) focuses on outside attacks. In an outside attack, holding D as constant for the duration of the attack and recalling that H is aggregate 'incumbent' aggregate hash power, an attacker adds hash power of h_A to the network where $h_A > H$. In an inside attack, the attacker is someone who is able to coordinate or acquire control of h_A incumbent nodes where $h_A > \frac{1}{2}H$. More generally, an attack comprises a mixture of incumbent and additional hash power. That is, h_A must exceed $\underline{h}_A$, the minimum hash power for a successful attack where $\underline{h}_A \equiv \max\{\frac{1}{2}H, H - h_A^*\}$. For some cases below, it will be convenient to denote α as the share of hash power procured internally (e.g., $\alpha = \frac{h_A^*}{h_A}$). Then a majority attack requires $h_A > H - \alpha h_A$ or $h_A > \frac{1}{1+\alpha}H$.

A majority attack is an attempt to control and re-write transactions on the blockchains.

We follow Budish (2022) in assuming that the (net presented discounted expected) private benefits to an attacker, A, are V_{attack}. In specific applications, V_{attack} can be derived from further analysis to calculate the returns to double-spending or the censorship of other users.

The attack involves the majority miner creating a fork of the blockchain that they keep private. Difficulty does not adjust.

- on the attacking branch, all the blocks are mined by the attacker; the expected number of blocks per unit of time is $\frac{h_A}{D} = \frac{h_A}{\tau H}$.
- on the honest branch, the expected number of blocks per unit of time is $\frac{H-\alpha h_A}{\tau H} < \frac{h_A}{\tau H}$; the attacking branch is mining blocks faster on average than the honest branch.
- the attack will succeed with probability 1, and the expected (or intended) duration of the attack is $\mathcal{L}$.[5]

If the costs of mounting an attack of expected length $\mathcal{L}$ are $C_{attack}(\mathcal{L})$, then a majority attack is not worthwhile if:

$$\Big(\frac{h_i^*}{\tau H}(R+\Phi) - c_i(h_i^*)\Big)\mathcal{L} \geq V_{attack} - C_{attack}(\mathcal{L})$$

Budish (2022) refers to this as an *incentive compatibility constraint* on Proof of Work blockchains for an equilibrium free of attacks to exist.

4 Costs of a Majority Attack

We are now in a position to characterise the expected costs of an attack, $C_{attack}(\mathcal{L})$. We will note here that these costs are comprised of the direct costs of an attack and indirect costs. We will characterise each in turn.

If A supplies hash power of h_A to the attack chain, the costs of doing so are $c_A(h_A)$. As these are applied for the duration of the attack, the direct cost component of $C_{attack}(\mathcal{L})$ are $c_A(h_A)\mathcal{L}$.

The indirect cost component of C_{attack} is actually a mitigation of costs because, once A establishes the longest chain following the attack, they will receive the block rewards and transaction fees earned during the attack. Let $\tilde{R}$ and $\tilde{\Phi}$ be those benefits for each block mined during an attack. As A is the only miner on the attack, chain their expected number of blocks per unit of time is $\frac{h_A}{D}$. Thus, the second component of C_{attack} is $\frac{h_A}{D}(\tilde{R}+\tilde{\Phi})\mathcal{L}$.

[5]Budish (2022), examining an outside attack, provides a calculation for the expected number of blocks to be mined before the attacker has the longest chain. Here we want $\mathcal{L}$ to be the clock time, not the number of blocks but also, as will be shown, the precise level of $\mathcal{L}$ does not matter for the results below.

Putting these together,

$$C_{attack}(\mathcal{L}) = c_A(h_A)\mathcal{L} - \frac{h_A}{D}(\tilde{R} + \tilde{\Phi})\mathcal{L}$$

However, we can expect that during the attack, A will choose hash power $\tilde{h}_A$ to minimize $C_{attack}(\mathcal{L})$ subject to $h_A \geq \underline{h}_A$ which implies that

$$C_{attack}(\mathcal{L}) = \Big(c_A(\max\{\tilde{h}_A, \underline{h}_A\}) - \frac{\max\{\tilde{h}_A, \underline{h}_A\}}{D}(\tilde{R} + \tilde{\Phi})\Big)\mathcal{L}$$

It is assumed, for the moment, that $\tilde{R} + \tilde{\Phi} \geq R + \Phi$ (which will be shown to be true below) and, therefore, that $\tilde{h}_A \geq h_A^*$. Note that it is entirely possible that C_{attack} could be zero or negative. Regardless, the incentive compatibility constraint becomes:

$$\Big(\frac{h_i^*}{\tau\, H}(R + \Phi) - c_i(h_i^*) - \frac{\max\{\tilde{h}_A, \underline{h}_A\}}{\tau\, H}(\tilde{R} + \tilde{\Phi}) + c_A(\max\{\tilde{h}_A, \underline{h}_A\})\Big)\mathcal{L} \geq V_{attack}$$

The left-hand side of this inequality, presenting the opportunity costs (that is, direct costs C_{attack} plus non-attack earnings) could be positive or negative. We turn now to explore the conditions that allow us to sign those opportunity costs.

4.1 Linear mining costs

To build intuition, consider the case that Budish (2022) focuses, on where $c_i(h_i) = c\, h_i$ for $i \in \mathcal{M}'$ and for external hash power $c_i(h_i) = \kappa c\, h_i$ where $c > 0$ and $\kappa \geq 1$. Also, assume that $\underline{h}_A$ is the minimal required hash power for a successful attack of expected duration $\mathcal{L}$. Thus, the total direct costs of an attack are $c(\alpha + \kappa(1-\alpha))\underline{h}_A\mathcal{L}$. The indirect cost component is $\frac{h_A}{D}(\tilde{R} + \tilde{\Phi})\mathcal{L}$.

As Budish (2022) notes, with linear and symmetric costs, the participation constraint holds for each miner with equality giving rise to a free entry condition for all $i \in \mathcal{M}'$:

$$\frac{h_i}{\tau\, H}(R + \Phi) = c\, h_i$$

Aggregating over $i \in \mathcal{M}'$, this simplifies to:

$$R + \Phi = c\,\tau\, H$$

Thus, the incentive compatibility constraint becomes:

$$\underline{h}_A\Big(0 - \frac{1}{\tau H}(\tilde{R} + \tilde{\Phi}) + c(\alpha + \kappa(1-\alpha))\Big)\mathcal{L} \geq V_{attack}$$

Using the free entry constraint, $c\tau H = R + \Phi$, this can be re-written as:

$$\frac{\underline{h}_A}{\tau H}\Big(- (\tilde{R} + \tilde{\Phi}) + (R + \Phi)(\alpha + \kappa(1-\alpha))\Big)\mathcal{L} \geq V_{attack}$$

Note that the right hand side is decreasing in α, κ and $\tilde{R} + \tilde{\Phi} - (R + \Phi)$.

This analysis provides some generalization over Budish (2022) who implicitly assumes that $\tilde{R} + \tilde{\Phi} = R + \Phi$. Note that, in this case, the incentive compatibility constraint becomes:

$$\frac{h_A}{\tau H}(\kappa - 1)(1 - \alpha)(R + \Phi)\mathcal{L} \geq V_{attack}$$

The left-hand side will be zero if either $\kappa = 1$ (implying that $C_{attack} = 0$ or if $\alpha = 1$). In these cases, so long as $V_{attack} > 0$, the blockchain is not secure. A majority attacker either faces "zero" net attack costs or "zero" opportunity costs from an attack. Thus, a majority attacker need not lose anything in order to earn V_{attack}.

Our analysis puts focus on what happens to block rewards and transaction fees during an attack as these drive the right-hand side of the incentive compatibility constraint when $(\kappa - 1)(1 - \alpha) > 0$.

4.2 General mining costs

When there are general mining costs, $c_A(h_A)$ represents the total cost's A has to expend per period in order to apply h_A in hash power to the network (whether attacking or not). Thus if $\underline{h}_A \leq h_A^*$, i.e., the required hash power for a majority attack is less than the hash power A would devote internally to mining when not attacking. In this case, some outside hash power is required and given the quasi-convexity of $c_A(h_A)$, then that additional hash power will involve higher marginal costs than the purely internal average costs of $\frac{c_A(h_A^*)}{h_A^*}$.

Given this, it is instructive to provide some more precise definitions of an inside versus an outside attack.

- If $\underline{h}_A \leq h_A^*$, a *purely internal attack* is feasible. However, even in this case, if $\tilde{h}_A > h_A^*$, outside hash power of $\tilde{h}_A - h_A^*$ is applied to the attack.

- If $\underline{h}_A > h_A^*$, a purely internal attack is not feasible and outside hash power is required.

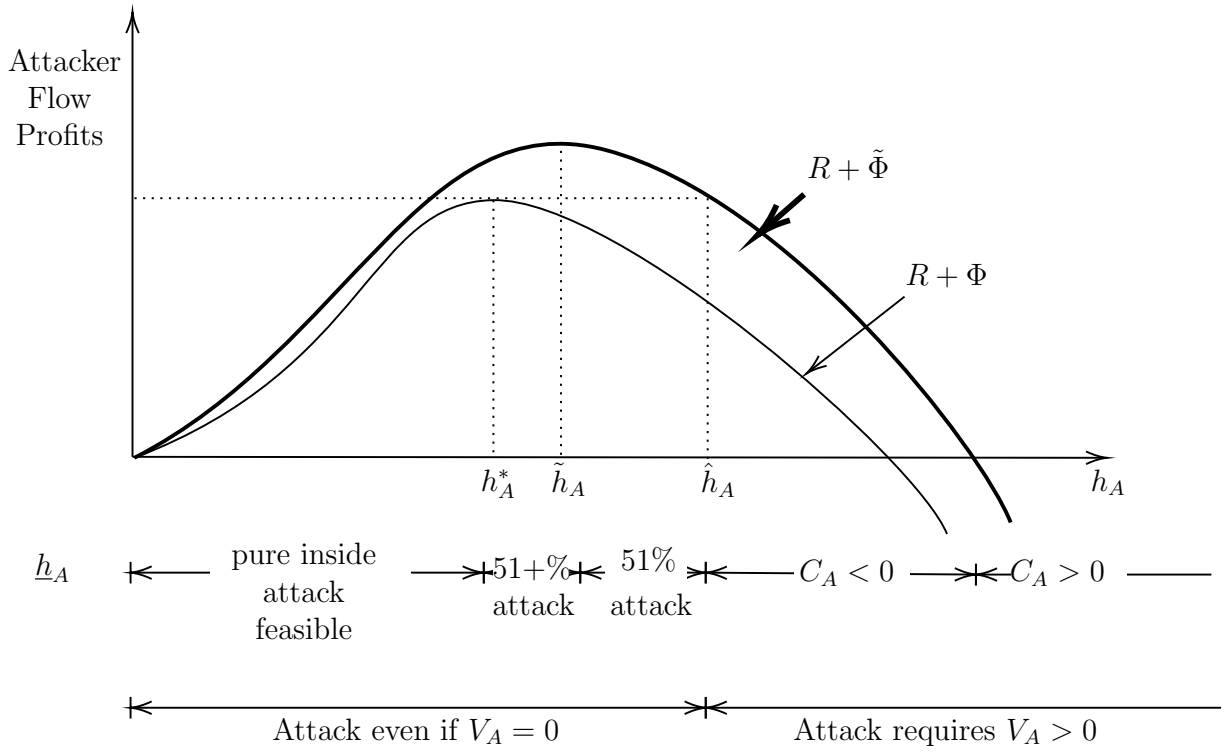

Figure 1: **Attacker Profits and Costs**

- If $\underline{h}_A > \tilde{h}_A$, then the total amount of hash power applied to the attack is $\underline{h}_A$. Let $\hat{h}_A \equiv \{h_A | \frac{h_A^*}{\tau H}(R+\Phi) - c_A(h_A^*) = \frac{h_A}{\tau H}(\tilde{R}+\tilde{\Phi}) - c_A(h_A)\}$. If $\underline{h}_A > \hat{h}_A$, then A's earnings during the attack will be lower than their earnings had they not undertaken the attack. In this case, an attack involves a positive *opportunity* cost for A.

This implies that for direct attack costs to be positive, i.e., $C_{attack}(\mathcal{L}) > 0$, it must be the case that $\frac{h_A}{\tau H}(\tilde{R}+\tilde{\Phi}) - c_i(\underline{h}_A) < 0$. A necessary condition for this is that $\tilde{h}_A < \underline{h}_A$. Thus, if a purely internal attack is feasible, then $C_{attack}(\mathcal{L}) < 0$; direct attack costs are negative. Figure 1 depicts these different attack characteristics.

One case where the sign of the left-hand side of the incentive compatibility constraint can be identified is when mining costs are symmetric amongst all miners (i.e., $c_i(h_i) = c(h_i)$ for all i) and the majority attack is purely outside (i.e., $h_A^* = 0$). In this case, $= \frac{h_A^*}{\tau H}(R+\Phi) = c_A(h_A^*)$ and so $\frac{h_A}{\tau H}(\tilde{R}+\tilde{\Phi}) \leq c_A(\underline{h}_A)$. Thus, $C_{attack} > 0$. This, however, is just another variant of the result found by Budish (2022) for the linear cost case.

4.3 Renting internal hash power

As argued by Bonneau (2016), it may be possible for the attacker to rent hash power from miners currently participating in the blockchain. To rent h_I^R of hashing power currently in

use, the attacker needs to compensate the owner, i.e., pay

$$r(h_I^R) = \frac{h_I^R}{D}(R+\Phi) - c_I^R(h_I^R) + \varepsilon_I \gtrsim \frac{h_I^R}{D}(R+\Phi) - c_I^R(h_I^R)\,.$$

for some $\varepsilon_I \gtrsim 0$.

Then the cost of the attack becomes

$$c_A(h_A) - \frac{h_A}{D}(\tilde{R}+\tilde{\Phi}) + c_I^R(h_I^R) - \frac{h_I^R}{D}(R+\Phi) + r(h_I^R) = c_A(h_A) - \frac{h_A}{D}(\tilde{R}+\tilde{\Phi}) + \frac{h_I^R}{D}(R+\Phi-(\tilde{R}+\tilde{\Phi})) + \varepsilon_I$$

where h_A is the power the attacker directly provides himself, and h_I^R is the power he rents from the current miners.

Notice that renting hashing power that is currently in use affects $\underline{h}_A$. The minimal hashing power needed for majority attack when renting internally is $\underline{h}_I^A(h_I^R) \gtrsim H - h_A^* - h_I^R$. When i rents $h_I^R > \frac{1}{2}H - h_A^*$, then $h_A^* + h_I^R > \underline{h}_A(h_I^R)$. This means that the attacker can conduct a successful majority attack using no more than their original h_A^*.

5 Transaction fees collected during an attack

The above analysis shows that a key driver of the costs of a majority attack are differences between block rewards and transaction fees during an attack (i.e., $\tilde{R}$ and $\tilde{\Phi}$). Note that, with Bitcoin, the block reward rarely changes and is fixed by the protocol. Thus, it is difficult to think of reasons why $\tilde{R}$ would differ from R. For this reason, we focus here instead on what happens to transaction fees.

Every transaction t is characterized by amount of space it takes, $\sigma_t > 0$ and transaction fee *per unit of space* offered by a user, $\varphi_t \geq 0$. Transactions arrive randomly and independently, and on average, the sum of σ_t for all transactions arriving in a unit of time is σ.

Blocks have a fixed capacity of b that is set by the protocol. For Bitcoin, the block size is limited to 1MB, which can house over 2,000 transactions. If blocks, on average, are not at capacity, i.e., $\tau\,\sigma \leq b$, then users gain no advantage in terms of the speed of transaction processing and so will not find it optimal to offer a fee above zero. In this case, $\Phi = 0$.

On the other hand, if the demand for blockchain transactions is higher than the supply, i.e., $\tau\,\sigma > b$, then only transactions paying the highest fee per unit of space get included in the block. In this case, to determine Φ, order all transactions available at a point in time (e.g., transactions that arrived during the last τ units of time since the last block) by φ in such a way that $\varphi_1 \geq \varphi_2 \geq ...$ with some of these inequalities strict. To avoid the knapsack problem, it is assumed that all transactions have the same size and therefore, σ is simply the

average number of transactions. Then the fees collected with this block are $\Phi = \sigma \sum_{t=1}^{B} \varphi_t$, where T is defined by $\sum_{t=1}^{B} \sigma_t = b$; which is equivalent to picking the b highest φ's from the distribution where φ_t is repeated σ times

Given this, we can prove the following.

Lemma 1 *Suppose that users do not alter their transaction fee submissions during an attack and that* $h_A \in (\frac{1}{2}H, H]$. *If* $\tau\,\sigma > b$, $\tilde{\Phi} \geq \Phi$.

Proof. Note that unless $h_A \geq H$, the total number of blocks confirmed per period on the attacking chain is strictly less than the number of blocks confirmed per period when there is no attack. As users do not alter their transaction fee submissions, the attacking chain has a strictly lower total block capacity per period. Thus, there exist transactions that would have been confirmed in equilibrium but for the attack that is not confirmed on the attacking chain. As the attacker chooses transactions for each block to maximize total fees, this implies that total fees on the attacking chain per block exceed those that would otherwise arise per block in equilibrium. ■

The simple intuition behind this result is that under the attack, fewer blocks are added to the blockchain per unit of time, and therefore the capacity of the blockchain (i.e., supply of processed transactions) is lower, while the demand for processing transactions remains the same. When mining a block under attack, A has not only the same transactions available as they would have without an attack but also some new arrivals, which may offer higher fees. (These "new arrivals" would have been picked up by other miners without attack.)

It is useful to note that, for the same reason, fees on the honest chain during the attack also increase. However, these fees are never paid out to miners as the attacking chain eventually becomes the longest chain.

Given this, we have demonstrated the following:

Proposition 1 *Suppose that there exists a miner* A *such that* $\underline{h}_A \leq \hat{h}_A$. *An equilibrium without a majority attack does not exist even if* $V_{attack} = 0$.

Proof. The condition that $\hat{h}_A \geq \underline{h}_A$ says that A earns positive profits during the attack and that these profits equal or exceed the profits A would earn without an attack. Thus, the opportunity cost of an attack is negative.

Note that, be definition, $\underline{h}_A \leq H$, so that the attacking blockchain at most matches the speed of the original blockchain without an attack. Thus, $\tilde{\Phi} \geq \Phi$ as per Lemma 1. ■

If $\hat{h}_A < \underline{h}_A$, then the opportunity cost of an attack is positive (see Figure 1). In this case, an attack would only proceed if $V_{attack} > 0$.

5.1 Possibility of renting internal hashing power

With $\tilde{\Phi} > \Phi$, the possibility of renting internal hashing power makes the attacks less costly.

Proposition 2 *With possibility of renting internal hashing power, any miner can successfully attack at negative cost of attack, i.e., the attack is beneficial even if* $V_{attack} = 0$.

Proof. Acknowledging that $\tilde{R} = R$, when $\tilde{\Phi} > \Phi$, the cost of attack with renting h_I^R of hashing power already used in the blockchain is

$$c_A(h_A) - \frac{h_A}{D}(R + \tilde{\Phi}) + \underbrace{\frac{h_I^R}{D}(\Phi - \tilde{\Phi}) + \varepsilon_I}_{<0}$$

Since A can rent $h_I^R > \frac{1}{2}H - h_A^*$, then successfully attacking with h_A^* of own hash power is an option. Optimizing attacking has power to minimize the cost,

$$C_{attack} \leq \left(c_A(h_A^*) - \frac{h_A^*}{D}(R + \tilde{\Phi}) + \frac{h_I^R}{D}(\Phi - \tilde{\Phi}) + \varepsilon_I\right)\mathcal{L} < \left(c_A(h_A^*) - \frac{h_A^*}{D}(R + \tilde{\Phi})\right)\mathcal{L} < \left(c_A(h_A^*) - \frac{h_A^*}{D}(R + \Phi)\right)\mathcal{L}$$

which means that the incentive compatibility constraint is violated, i.e., it is beneficial to attack even if $V_{attack} = 0$. ■

5.2 Reflections on an outside attack

The above analysis demonstrates that the scenario of a purely outside attack is somewhat of a special case in understanding the security of the blockchain. For that reason, it is useful to consider this case more carefully. In so doing, we focus on the linear cost case.

Recall that in a purely outside attack, the expected direct cost of mining per unit of time is $ah_A = a\alpha H$, whereby free entry condition $a = \frac{R+\Phi}{\tau H}$. So the expected cost of mining during the attack is $ah_A = \frac{\alpha}{\tau}[R + \Phi]$. The mitigating mining rewards during the attack bring in expectation $\frac{\alpha}{\tau}(R + \tilde{\Phi})$. So the net cost of the attack per unit of time is $\frac{\alpha}{\tau}([E\Phi - \tilde{\Phi})$. The total expected cost of the attack is:

$$\frac{\mathcal{L}(\alpha)}{\tau}\alpha(\Phi - \tilde{\Phi}) > 0$$

It would seem that with an attack from outside, the attack's cost increases with the attack's length. However, the attacker can always limit the number of blocks mined in the attack

to one more than the number of blocks mined on the honest blockchain. That is, the moment that A mines one more block than the honest blockchain, the attack ends.[6] Thus, A produces only one more block than the honest blockchain, no matter how long the attack lasts. Therefore, more properly, the expected number of blocks A mines in the attack is $\frac{\mathcal{L}}{\tau}+1$. The expected mining cost (with shutting down if necessary) is $\left(\frac{\mathcal{L}}{\tau}+1\right)(R+\Phi)$. The mitigating mining rewards are $\left(\frac{\mathcal{L}}{\tau}+1\right)(R+\tilde{\Phi})$.

Note that with $\left(\frac{\mathcal{L}}{\tau}+1\right)\tilde{\Phi}$, A processes all the transactions that would go into the honest blocks during $\frac{\mathcal{L}}{\tau}$, and they also creates one additional block of "second tier" transactions paying strictly less. That is, the total fees collected by the attacker throughout the attack are $\frac{\mathcal{L}}{\tau}\Phi+\tilde{\Phi}(\mathcal{L})$. Note also that $\tilde{\Phi}(\mathcal{L})$ is increasing in the length of the attack $\mathcal{L}$. This is because there are more second-tier transactions, and the highest paying b transactions of second-tier transactions pay more. Therefore, the total expected cost of the majority attack from the outside is

$$\Phi-\tilde{\Phi}(\mathcal{L})>0$$

The net cost of the attack is still positive, but it decreases as the attack takes longer.

5.3 Comparison with selfish mining

It is useful to compare this result with the work by Eyal and Sirer (2018). That paper demonstrates that a non-majority miner controlling between one-third and one-half of total hash power can find it optimal to deviate for the honest outcome and mine a private chain when all other nodes are honest. The difference here is that the mechanism for a majority attack is a straightforward majority attack where the attacker controls more hash power than all other miners during the attack. Thus, it is an attack available only when the attacker can achieve a majority whereas selfish mining is only profitable for a minority miner. In this regard, the two results are complementary. They expand the potential range of outcomes whereby an honest equilibrium outcome may not be sustainable.

[6]Recall there is no escrow period as that is a convention outside of Proof of Work protocols. If there were an escrow rule, this would not change the logic. Suppose an attacker is forced to wait until the honest blockchain produces at least w blocks before revealing their longer chain. If A mined continuously during that time, they would mine in expectation αw blocks, possibly more than $w+1$. However, the attacker does not need to mine more than $w+1$. Thus, they can shut down mining after reaching $w+1$, and wait until the honest block reaches w to reveal their longer chain. This will allow them to save on the mining cost that is not necessary for the success of the attack.

6 Accounting for Dynamic Adjustments

The above model is static in nature in that it is assumed that the behavior of users, difficulty levels and honest miner behavior do not change during the attack. Here we account for these potential dynamic adjustments to examine the robustness of our results above.

6.1 Users adjusting their transaction fee bids

Until now, we have assumed that the transactions (including fees offered) arrive in the same way during the attack as they would without an attack. But, in fact, the users may adjust the fees they offer based on the congestion they observe.

We assume that the attacker keeps their branch of the fork secret until ready to reveal the longest chain. Only the honest blockchain (first the benchmark blockchain and then the honest non-attacking branch) is visible to the users before the attack is executed, and users respond only to this.

In the case of a fully outside attack, users see no change and do not adjust their bids. In the case of an internal attack, the time between the blocks on the honest branch increases, decreasing the supply of transactions recorded per unit of time. Therefore, users who adjust their bids will adjust upward, increasing the difference, $\tilde{\Phi} > \Phi$.

6.2 Algorithm adjusting mining difficulty

The previous analysis held the difficulty, D, of the blockchain fixed at $D = \tau H$ where H is the previous aggregate hash power and τ is the targeted average time between block confirmations. However, as already noted, during an attack, when $\underline{h}_A < H$, the time between block confirmations (on both the honest and attacking chain) increases. If difficulty were to adjust during the attack, it would restore the timing of blocks back to τ. Because the aggregate hash power on the honest and attack chain are different, then, by the time, the attack chain is made public, the two chains would have different difficulties. Here we explore how this changes the operation and incentives of an attack.

A simple conjecture might be that if difficulty adjusts immediately, something we can theorize about even though it cannot occur in practice, the two competing forks' difficulties would adjust and either one could be the longest chain and the attack chain, despite having more hash power, would have no advantage. However, the longest chain rule is not, in fact, purely a convention based on the longest chain. Instead, when two competing fork branches have different difficulties, the generalized longest chain rule becomes the *heaviest chain* rule. That is, the number of blocks is weighted by the difficulty so that the branch with the most

computational power behind it is chosen.[7]

In Bitcoin, difficulty adjusts every 2600 blocks. To adjust, the algorithm takes the average (reported) clock time that it has taken to mine these 2600 blocks, and if that number is different than 10 min, it adjusts the difficulty to such a number that would yield 10 min given the power used over the past 2600 blocks. Note that this adjustment is purely retrospective and does not account for the difference in more recent changes vs old changes. Therefore, if twice as much hashing power was present over the last ten blocks, the algorithm won't assume this is the hashing power going forward. It will take the average over the 2600 blocks.

Now, suppose that the attack started d blocks before the change of difficulty. On average, it took $\tau(2600-d)$ time to mine the blocks on the benchmark blockchain. Then, the average time to mine a block on the attacking branch with power h_A and difficulty $D = H\tau$ is $Y_A = \frac{1}{\theta_A} = \frac{D}{h_A}$. So at the time of the difficulty adjustment, the total average time it took to mine 2600 blocks on this branch is $\tau(2600-d) + d\tau\frac{H}{h_A} = \tau\big((2600-d) + d\frac{H}{h_A}\big)$.

With that, the new difficulty is

$$D'_A = D\frac{2600}{(2600-d) + d\frac{H}{h_A}}\,.$$

Note that $D'_A < D$ when $h_A < H$ and $D'_A > D$ when $h_A > H$. Moreover,the new expected time between blocks is

$$Y'_A = \frac{D'_A}{h_A} = \tau\frac{H}{h_A}\frac{2600}{(2600-d) + d\frac{H}{h_A}}\,.$$

When $d = 2600$, then $Y'_A = \tau$. But when $d < 2600$, then $Y'_A > \tau$ when $h_A < H$ and $Y'_A < \tau$ when $h_A > H$.

Whenever $h_A \geq \underline{h}_A$, then A's attack eventually succeeds with probability 1. This is because, by the definition of $\underline{h}_A, h_A \geq \underline{h}_A$ will make the attacking branch heavier, even if not longer. Therefore, we can express the clock-counted expected length of the attack as $\mathcal{L} = \mathcal{L}_{preD} + \mathcal{L}_{postD}$, where $\mathcal{L}_{preD}$ is the expected length of the attack before the difficulty change ($\mathcal{L}_{preD} = d\tau$) and $\mathcal{L}_{postD}$ is the expected length of the attack after the difficulty change.

In such a case, the cost of the attack is

$$C_{attack}(h_A) = \mathcal{L}_{preD}[c_A(h_A) - \frac{h_A}{D}(R + \tilde{\Phi})] + \mathcal{L}_{postD}[c_A(h_A) - \frac{h_A}{D'_A}(R + \tilde{\Phi}_{postD})]$$

where D'_A is the difficulty on the attacking branch after adjustment, and $\tilde{\Phi}_{postD}$ are the

[7]The difficulty of each fork is observable to all miners as it is captured in the level of the computational problem each faces. See e.g., https://learnmeabitcoin.com/technical/longest-chain

expected fees per block on the attacking branch after the adjustment.

So, if $h_i^A < H$, then $D'_A < D$ and the attacking branch mines more blocks than before the adjustment change, $\frac{h_A}{D'_A} > \frac{h_A}{D}$. Thus, the attacker gets more block rewards per unit of time after the difficulty adjustment. But the expected fees are lower per block after the adjustment than before, $\tilde{\Phi}_{postD} < \tilde{\Phi}$. Nonetheless, they are still (weakly) higher than on the benchmark blockchain, $\tilde{\Phi}_{postD} \geq \Phi$, because $Y'_A \geq \tau$, i.e., the blocks are created more slowly, and thus the capacity of the blockchain on the attacking branch is weakly smaller than on the benchmark blockchain. It is strictly smaller, and $Y'_A > \tau$ when $d < 2600$. Altogether, $\frac{h_A}{D'_A}(R + \tilde{\Phi}_{postD}) > \frac{h_A}{D_A}(R + \tilde{\Phi}_{postD}) > \frac{h_A}{D_A}(R + \Phi)$. Therefore, the cost of attacking for $h_A^* > 0$ is lower than the opportunity cost even with difficulty adjustment:

$$\min_{h_A} C_{attack}(h_A < H) \leq C_{attack}(h_A^*) < \left(c_A(h_A^*) - \frac{h_A^*}{D}(R + \Phi) \right) \mathcal{L}\,,$$

confirming our base result.

If $h_A > H$, then $D'_A > D$, which means that the number of blocks the attacker mines after the difficulty change is lower than before the difficulty change, $\frac{h_A}{D'_A} < \frac{h_A}{D}$. Nonetheless, the new expected time between blocks is still lower than on the benchmark blockchain, $Y'_A < \tau$, which means $\tilde{\Phi}_{postD} < \Phi$. With that,

$$C_{attack}(h_A > H) > \left(c_i(h_A) - \frac{h_A}{D}(R + \Phi) \right) \mathcal{L}\,.$$

The cost of attack is positive when $c_A(h_A) \geq \frac{h_A}{D}(R + \Phi)$. With free entry condition, it holds when the attacker is weakly less efficient than the marginal miner, e.g., when the cost of mining is linear $c_A(h) = c\,h$.

6.3 Miners adjusting their participation

For our earlier results, we assumed that an attack was short-run and, thus, the number of honest miners was held constant. Thus, there was no entry or exit of honest miners. Accounting for changes in the participation constraint of honest miners impacts the results in an important way.

Many potential miners may be ready to join if the mining becomes marginally more profitable. At the outset, prior to an attack, the blockchain profit of a marginal miner is zero, and thus additional miners do not enter. In case of an external attack, the observable honest branch does not differ from the benchmark blockchain, and hence there is no additional entry.

In the case of an internal attack, the honest branch of the fork slows down, increasing mining rewards as fees per block increase (whether adjusted or not), and block reward stays the same. That will encourage new mining power to enter.

If the potential entrant miners on the fringe have at least the same mining efficiency as the marginal miner in the benchmark blockchain, then the honest branch of the blockchain will grow up to H in mining power. That changes the requirement on $\underline{h}_A$, and basically requires that $\underline{h}_A \gtrsim H$, making it a fully external attack.

If the entry of these new, equally efficient miners happens immediately, the cost of the attack is larger than the opportunity cost. This is because the cost of the attack is positive (because it's effectively a fully external attack), and the opportunity cost is weakly negative (the miner had non-negative payoff from mining on the benchmark blockchain).

If the entry of these new, equally efficient miners is delayed, the cost of the attack is negative until the honest branch reaches H. This is because A can conduct a majority attack with $h_A < H$. Once the honest branch reaches H, the attacker needs to deploy $h_A > H$, and the cost of the attack becomes positive. Whether the total cost of the attack is positive or negative depends on the length of the attack. Since the speed at which the new miners enter the honest branch is independent of h_A, it may be optimal to increase h_A to increase the chance that the attack finishes before the honest branch reaches H and the cost of the attack becomes positive.

If the newly entering miners on the honest branch are less efficient than those on the benchmark blockchain, the honest branch will not reach H before the marginal miner breaks even. With less than H on the honest branch, A can conduct a successful majority attack with $h_A < H$, and thus with the cost of an attack less than the opportunity cost.

7 Conclusion

In designing the Bitcoin network, Nakamoto (2008) concluded that "[a]ny needed rules and incentives can be enforced with this [Proof of Work] consensus mechanism." While it was thought that an entity that controlled the majority of miners could attack the blockchain, the assumption was that this would be costly. This paper has demonstrated that this contention does not hold. Using within-protocol mechanisms only but for the special case of a purely external attacker, an attack can be carried out with, at best, zero costs and, more likely negative costs. Thus, what regulates the security of the network is purely external.

There are many candidates for external regulation, but these are often contingent on the precise motives for an attack. For example, a double-spend attack by which an attacker censors past transactions in order to re-spend tokens may require an agreed-upon escrow

period (something Nakamoto (2008) suggested). However, even here, a negative cost attack could be carried out indefinitely. In other cases where the attacker remains invested in the network post-attack (say, directly or indirectly because of the ownership of specialized processors), an anticipated fall in the value of crypto-tokens could deter an attack. Again, how this would operate is external to the protocol and, at present, the evidence that it is a regulating device is mixed.[8] This points to future research focussing on the nature of V_{attack} and developing an understanding of the external institutional and social mechanisms that may reduce or eliminate it.

[8]See Kwon et al. (2019), Ramos et al. (2021) and Shanaev et al. (2019).

A Appendix: Proof of Stake with Nakamoto Consensus

@articlehalaburda2022microeconomics, @articlehalaburda2022microeconomics, In this appendix, we demonstrate that the same outcomes as those derived for a majority attack on Proof of Work blockchains also apply to permissionless Proof of Stake blockchains with Nakamoto Consensus.[9]

A.1 Equilibrium without an Attack

Let $\mathcal{V}$ be the set of validators with individual validator $i \in \mathcal{V}$. The stake, in blockchain-native coins) of i is s_i. Staking involves a cost associated with locking up capital in the network. Let $r(s_i) = r\, s_i e_s$ – average cost of staking (locking up the capital) per unit of clock time, where e_s is the (expected) exchange rate and r could be interpreted as an interest rate.

The staking process and block proposer selection proceeds as follows:

- the (approximate) time between the blocks is set by the protocol to be τ_s
- the protocol calls validator i to mint a block with probability proportional to their stake; each draw independent
  - every τ_s, validator i mints a block with probability $\frac{s_i}{S}$
  - within a time $K = k\,\tau_s$ validator i expects to mint $k\frac{s_i}{S}$ block, since each block's draw independent
- if a validator is called but does not propose a block, that is, they are a no-show, the block missing, next node selected according to staking proportion for next τ_s
- if a validator is called and proposes a valid block they potentially receive newly minted tokens and transaction fees with expected values in fiat currency of R_s and Φ_s respectively. (Transaction fees are determined in the same way as under Proof of Work.)

Thus, each validator can choose a stake level that gives them a higher probability of being the stake proposer. That is, if the total amount staked is $S = \sum_{i\in\mathcal{V}} s_i$, then the probability of being selected to propose a stake in any epoch is $\frac{s_i}{S}$.

[9]There are two broad variants of Proof of Stake. The first one to emerge was based on Nakamoto consensus whereby, if there were forks, the chain that validators extended would be the one with the most blocks. This was the consensus mechanism of PeerCoin. This is not the most widely used variant today. That is based on Byzantine Fault Tolerance (BFT) and is not vulnerable to the majority attacks as analyzed in this paper. For more on the distinction between these approaches see Gans (2023). Amoussou-Guenou et al. (2019), Auer et al. (2021) and Halaburda et al. (2021) discuss attack vulnerabilities of BFT blockchains.

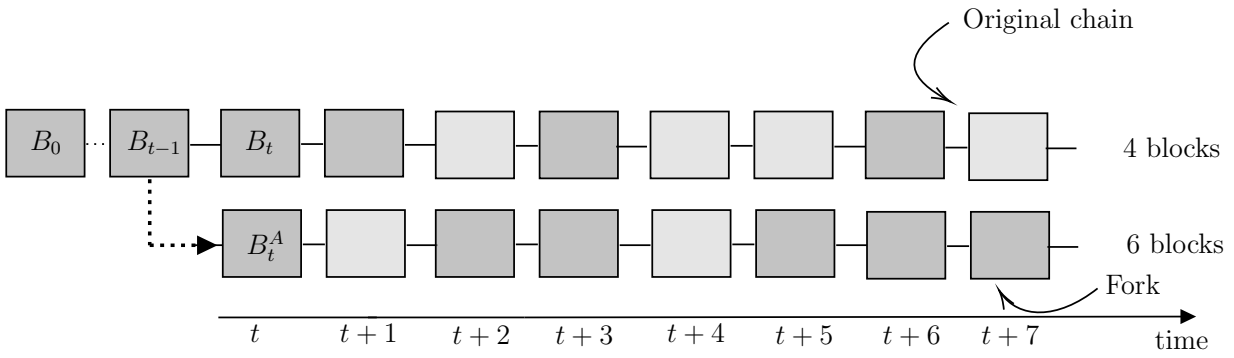

Figure 2: **Majority Attack under Proof of Stake**

Given this, for a clock interval $K = k\,\tau_s$, each validator i chooses s_i to maximize expected profits:

$$k\frac{s_i}{S}(R_s + \Phi_s) - k\,\tau_s\,r\,s_i e_s$$

If s_i^* is the maximized stake amount, then the participation constraint for each i is:

$$\frac{1}{S}(R_s + \Phi_s) \geq \tau_s\,r\,e_s$$

Let $\mathcal{V}' \subseteq \mathcal{V}$ be the set of validators who satisfy this participation constraint. Given the linear costs that arise under Proof of Stake, free entry implies that the participation constraint will bind for all $i \in \mathcal{V}'$ so that:

$$\frac{1}{S}(R_s + \Phi_s) = \tau_s\,r\,e_s \implies R_s + \Phi_s = \tau_s\,r\,e_s\,S$$

A.2 Attack by a Majority Staker

As validators must be accepted into the protocol, the relevant mode of attack is an inside attack.[10] We now describe that attack as depicted in Figure 2.

Suppose that an entity that controls majority of the stake, $s_A > \frac{1}{2}S$ creates a fork. Under PoS, a validator cannot attach a block to the blockchain unless they are "called by the protocol" to do it. Let B_{t-1} be the last block before the fork. To create the fork, A accepts B_{t-1}, but thereafter does not accept any block created by other validators than themself. If the protocol running on his local machine calls for a validator other than A, there is no response, and after τ_s the protocol calls another validator. Since the stake in the protocol has not changed, A is called by the protocol with probability $\frac{s_A}{S}$ every interval τ_s, and only these blocks are added to the attacking branch.

[10]Previous analyses of Proof of Stake attacks assumed an outside attack; e.g., Gans and Gandal (2021), Halaburda et al. (2022a).

At the same time, A withholds their blocks from the honest branch. Thus, during the attack, a block is added to the honest branch with probability $1 - \frac{s_A}{S}$ every interval τ_s. Note that with $s_A > \frac{1}{2}S$, the attack will succeed with probability 1 at some point. Let $\mathcal{L} = l\,\tau_s$ be the expected length of the attack in clock time. During $\mathcal{L}$, attacker A adds in expectation $l\,\frac{s_A}{S}$ blocks, and these are all the blocks on the attacking blockchain. Without the attack, over $\mathcal{L}$, A would also add $l\,\frac{s_A}{S}$ blocks in expectation. Thus, A receives proposes the same number of blocks with and without the attack.

If V_{attack} is the net present discounted expected payoff to A from a successful attack, then the incentive compatibility constraint is:

$$k\frac{s_A}{S}(R_s + \Phi_s) - k\,\tau_s\,r\,s_A e_s \geq V_{attack} - C_{attack}$$

Note that, by free entry, the left-hand side of this inequality is 0 so the constraint collapses to $C_{attack} \geq V_{attack}$. Here, C_{attack} is:

$$C_{attack}(\mathcal{L}) = r\,s_A e_s \mathcal{L} - \frac{s_A}{S}(\tilde{R}_s + \tilde{\Phi}_s)$$

where $\tilde{R}_s$ and $tilde\Phi_s$ are the block rewards and transaction fees earned during an attack.

Note that, as in Proof of Work, it is likely that $\tilde{R}_s = R_s$. In addition, using the free entry condition, C_{attack} becomes $\frac{s_A}{\tau_s S}\big(\Phi_s - \tilde{\Phi}_s\big)\mathcal{L}$. Thus, the incentive compatibility condition becomes:

$$\frac{s_A}{\tau_S S}\big(\Phi_s - \tilde{\Phi}_s\big)\mathcal{L} \geq V_{attack}$$

By the same argument as for Proof of work (that is, Lemma 1), it is easy to see that $\tilde{\Phi}_s \geq \Phi_s$. Thus, we have demonstrated the following result:

Proposition 3 *An equilibrium without a majority attack does not exist if there exists a majority validator even if* $V_{attack} = 0$.

Note that this is a stronger result than under Proof of Work due to the symmetric and linear costs associated with being a validator.

References

Amoussou-Guenou, Y., Biais, B., Potop-Butucaru, M., and Tucci-Piergiovanni, S. (2019). Rationals vs byzantines in consensus-based blockchains. *arXiv preprint arXiv:1902.07895*.

Auer, R., Monnet, C., and Shin, H. S. (2021). Permissioned distributed ledgers and the governance of money.

Bakos, Y. and Halaburda, H. (2021). Tradeoffs in permissioned vs permissionless blockchains: Trust and performance. *NYU Stern School of Business working paper*.

Biais, B., Bisiere, C., Bouvard, M., and Casamatta, C. (2019). The blockchain folk theorem. *The Review of Financial Studies*, 32(5):1662–1715.

Bonneau, J. (2016). Why buy when you can rent? bribery attacks on bitcoin-style consensus. In *International Conference on Financial Cryptography and Data Security*, pages 19–26. Springer.

Budish, E. B. (2022). The economic limits of bitcoin and anonymous, decentralized trust on the blockchain. *University of Chicago, Becker Friedman Institute for Economics Working Paper*, (83).

Chiu, J. and Koeppl, T. V. (2022). The economics of cryptocurrency: Bitcoin and beyond. *Canadian Journal of Economics/Revue canadienne d'économique*, 55(4):1762–1798.

Eyal, I. and Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7):95–102.

Gans, J. S. (2023). *The Economics of Blockchain Consensus*. Palgrave.

Gans, J. S. and Gandal, N. (2021). Consensus mechanisms for the blockchain. In *The Palgrave Handbook of Technological Finance*, pages 269–286. Springer.

Halaburda, H., Haeringer, G., Gans, J., and Gandal, N. (2022a). The microeconomics of cryptocurrencies. *Journal of Economic Literature*, 60(3):971–1013.

Halaburda, H., He, Z., and Li, J. (2021). An economic model of consensus on distributed ledgers. Technical report, National Bureau of Economic Research.

Halaburda, H., Sarvary, M., and Haeringer, G. (2022b). *Beyond bitcoin*. Springer.

Huberman, G., Leshno, J. D., and Moallemi, C. (2021). Monopoly without a monopolist: An economic analysis of the bitcoin payment system. *The Review of Economic Studies*, 88(6):3011–3040.

Kwon, Y., Kim, H., Shin, J., and Kim, Y. (2019). Bitcoin vs. bitcoin cash: Coexistence or downfall of bitcoin cash? In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 935–951. IEEE.

Leshno, J. D. and Strack, P. (2020). Bitcoin: An axiomatic approach and an impossibility theorem. *American Economic Review: Insights*, 2(3):269–86.

Ma, J., Gans, J. S., and Tourky, R. (2018). Market structure in bitcoin mining. Technical report, National Bureau of Economic Research.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260.

Ramos, S., Pianese, F., Leach, T., and Oliveras, E. (2021). A great disturbance in the crypto: Understanding cryptocurrency returns under attacks. *Blockchain: Research and Applications*, 2(3):100021.

Shanaev, S., Shuraeva, A., Vasenin, M., and Kuznetsov, M. (2019). Cryptocurrency value and 51% attacks: evidence from event studies. *The Journal of Alternative Investments*, 22(3):65–77.