Blockchain Without Waste: Proof-of-Stake

Article *in* SSRN Electronic Journal · January 2018

DOI: 10.2139/ssrn.3183935

Fahad Saleh
Wake Forest University

Blockchain Without Waste: Proof-of-Stake[*]

Fahad Saleh[†]

Abstract

Permissionless blockchains require a protocol to generate consensus. Many prominent permissionless blockchains employ Proof-of-Work (PoW) for that purpose, but PoW possesses significant shortcomings. Various alternatives have been proposed. This paper provides the first formal economic model of the most famous alternative, Proof-of-Stake (PoS), and establishes conditions under which PoS generates consensus. A sufficiently modest reward schedule not only implies the existence of an equilibrium in which consensus obtains as soon as possible but also precludes a persistent forking equilibrium. The latter result arises because PoS, unlike PoW, requires that validators are also stakeholders.

Keywords: Blockchain, Consensus, Proof-of-Stake, FinTech

JEL Classification: C70, G00

[*] I am especially grateful to Robert Engle, Joel Hasbrouck, Kose John, Thomas Philippon, Rangarajan Sundaram, and David Yermack for extensive support and guidance. I am also grateful to Wei Jiang (Editor) and two anonymous referees for many helpful suggestions. Moreover, I thank Farshid Abdi, Yakov Amihud, Bruno Biais, Matthieu Bouvard, William Cong, Manasa Gopal, Hanna Halaburda, Christopher Hennessy, Franz Hinzen, Burton Hollifield, Gur Huberman, Jiasun Li, Dmitry Orlov, Andreas Park, Walter Pohl, Ioanid Rosu, Jacob Sagi, Noah Stoffman, Matt Weinberg, and Baozhong Yang and participants at Bank of Canada, the Bergen FinTech 2018 Conference, Carnegie Mellon University, the FTG Summer 2018 Conference, the GSU/RFS FinTech Conference 2019, Indiana University, London Business School, McGill University, the NFA 2018 Conference, NYU Stern, University of North Carolina, the WEAI 2018 Conference, and the WFA 2018 Conference for valuable comments.

[†] Wake Forest University. Email: salehf@wfu.edu.

A blockchain is a distributed ledger that records transactions across a network. The members of this network not only store but also update the ledger and are referred to as *validators*. The ledger's usefulness depends on validators agreeing on its contents because the ledger is supposed to provide transaction settlement and such settlement corresponds to all copies of the ledger recording the transaction. For that reason, agreement across validators, referred to as *consensus*, is a key issue for the viability of blockchain. This paper studies consensus within the context of a particularly important class of blockchain known as a *Proof-of-Stake (PoS)* blockchain.

In principal, consensus may be achieved on a blockchain by appointing a central authority to determine which transactions have settled. Such a protocol describes a restricted blockchain. Philippon (2016, p. 16) argues that "a restricted blockchain could in fact be used by incumbents to deter entry and stifle innovation" thereby "increase[ing] the rents of incumbents" but recognizes that "blockchain technology could improve ... efficiency" otherwise. Thus, blockchain's potential to improve financial market efficiency hinges on the viability of permissionless (i.e., unrestricted) blockchains; in fact, Chiu and Koeppl (2019) and Cong and He (2019) provide specific applications in which a permissionless blockchain improves efficiency. As discussed subsequently, this paper's topic of study, a PoS blockchain, is an important example of a permissionless blockchain.

A permissionless blockchain lacks a central authority and thus attainment of consensus is a nontrivial issue for such a blockchain. Nakamoto (2008) alleged to resolve the issue of consensus within a permissionless blockchain by employing an economic protocol known as Proof-of-Work (PoW). PoW requires validators to compete to update the blockchain. The competition consists of solving a trivial puzzle so that success probabilities depend on raw computational power only. Validators largely compete through energy expenditures, which have led to exploding levels of energy consumption.[1] de Vries (2018, p. 801) argues that Bitcoin, the most prominent PoW blockchain, consumes "at least 2.55 gigawatts of electricity

[1] Chiu and Koeppl (2017), Ma, Gans, and Tourky (2019), Saleh (2019), and Cong, He, and Li (2020) discuss the economic implications of this energy consumption.

currently, and potentially 7.67 gigawatts in the future, making it comparable with countries such as Ireland (3.1 gigawatts) and Austria (8.2 gigawatts)." Moreover, Mora et al. (2018, p. 931) highlight the unsustainability of PoW blockchains, noting that "Bitcoin... could alone produce enough CO2 emissions to push warming above 2 [degrees Celsius] within less than three decades."

In hopes of creating a sustainable permissionless blockchain (i.e., one that does not expend an exorbitant amount of energy), the blockchain community has bandied about several alternatives to PoW. As demonstrated by Irresberger, John, and Saleh (2020), the most widely used alternative is PoS; this paper theoretically studies PoS blockchains, blockchains that employ PoS as the economic protocol. PoS replaces PoW's competition by offering a randomly selected stakeholder the authority to update the blockchain. As such, PoS omits any incentive for validators to engage in a computational arms race. Many, however, remain skeptical regarding PoS's long-run viability because they fear that PoS fails to generate consensus. PoS, like PoW, offers a validator an explicit monetary reward, referred to as a *block reward*, to update the blockchain, but PoS, unlike PoW, does not require validators to incur an explicit monetary cost (such as that incurred from solving PoW's puzzle) to gain the authority to update the blockchain. Detractors assert that this lack of an explicit cost coupled with the explicit benefit of the block reward implies that a validator will always update the ledger whenever given the opportunity even if the update necessarily perpetuates disagreement. This assertion is known as the *Nothing-at-Stake problem,* and it nullifies the viability of PoS if true because it implies that an initial disagreement persists indefinitely. I detail the Nothing-at-Stake problem within Section 1.3.3.

This paper's first contribution is to establish that the Nothing-at-Stake problem is not valid in general. A blockchain possesses a native coin that facilitates exchange on that blockchain. A stakeholder of a given blockchain is an individual holding some native coins of that blockchain, and stake references the native coin holding of a stakeholder. The Nothing-at-Stake problem implicitly makes a price-taking assumption so that a validator does not internalize the effect of her decisions on the value of the blockchain's native coin. This assumption is not appropriate for a validator in a PoS setting. If a validator appends to the blockchain in a way that perpetuates disagreement, then she imposes a cost on all stakeholders because her action undercuts the ability of users to exchange the native coin and thereby lowers its value. PoS grants authority to update the blockchain to only stakeholders; thus, within PoS, a validator imposes a cost on herself if she updates the blockchain in a manner that persists disagreement.

Invalidity of the Nothing-at-Stake problem, however, need not imply that PoS generates consensus. As such, this paper takes the next step and provides conditions under which PoS generates consensus. That demonstration constitutes this paper's second contribution and one of the paper's main results (Proposition 2). Formally, I show that restricting access to update the blockchain to sufficiently large stakeholders induces an equilibrium that generates consensus expediently. This result arises because the cost of updating the blockchain in a manner that persists disagreement increases with a validator's stake. That cost increases in a validator's stake because, as previously referenced, persisting disagreement reduces coin value and stake references the number of coins held. For a sufficiently large stakeholder, the cost of persisting disagreement outweighs the benefit from the block reward and thus a sufficiently high minimum stake requirement to update the ledger generates an equilibrium in which validators coordinate on generating consensus.

Proposition 2 also implies that sufficiently low block rewards help generate consensus. In fact, as demonstrated by Corollary 3, a zero block reward generates consensus without a restriction on the set of stakeholders that may update the blockchain. This result arises because the zero block reward removes the only potential gain from propagating disagreement. Nonetheless, never updating any branch is also costly for stakeholders because the blockchain ledger must grow for the native coin to ever be exchanged, and the value of the native coin depends on its ability to be exchanged. Thus, with nothing to gain from updating multiple branches (zero block reward) and a cost incurred from updating no branches, a zero block reward induces an equilibrium in which all stakeholders coordinate to update a single branch of the blockchain.

Another important result within this paper, Proposition 5, demonstrates that a sufficiently modest block reward schedule implies that disagreement resolves eventually with probability one within any equilibrium. This result arises because stakeholder incentives involve two components: initial coin holdings and block rewards. When the latter component, block rewards, is sufficiently small, then the former component, initial coin holdings, becomes the dominant incentive. Since disagreement undermines the blockchain's native coin value, a sufficiently modest block reward schedule drives each stakeholder to prioritize generating consensus regardless of the strategies of other stakeholders. In turn, this behavior generates consensus eventually in any equilibrium.

The ability to generate consensus with low block rewards distinguishes PoS from PoW. The reason for this difference arises because PoS requires all validators to hold native coins. That requirement ensures that low block rewards shift validator incentives toward maximizing the value of native coins, which, in turn, encourages validators to seek consensus. Within PoW, validators need not hold any native coins so that reducing block rewards does not align incentives toward maximizing the value of native coins; rather, reducing block rewards undermines the incentive for validators to participate in the validation process at all.

Concretely, this paper offers two pieces of economic guidance. Specifically, this paper demonstrates benefits for PoS protocols imposing a validator minimum stake requirement and employing modest block reward schedules. A minimum stake requirement requires that the PoS protocol restrict access to update the ledger to sufficiently large stakeholders. A modest block reward schedule requires that block rewards offered to validators for updating the ledger be kept small. Both the discussed measures help generate consensus within a PoS protocol as demonstrated by Proposition 2, Corollary 3, and Proposition 5.

Computer science contains a large literature that studies consensus. That literature dates back to Lamport, Shostak, and Pease (1982). More recent papers within that literature, with relevance to permissionless blockchains, include Miller and LaViola (2014), Chen and Micali

(2016), Daian, Pass, and Shi (2016), and Kiayias, Russell, David, and Oliynykov (2017). This paper differs from those works that rely on exogenous behavioral assumptions by employing the standard economic paradigm of implying agent behavior based on preferences, payoffs, and equilibrium analysis. Some computer science papers, such as Eyal and Sirer (2014) and Nayak, Kumar, Miller, and Shi (2015), explicitly consider incentives but do not analyze equilibrium outcomes. Carlsten, Kalodner, Weinberg, and Narayanan (2016) conduct an equilibrium analysis but do so for PoW only.

This paper also relates to a large practitioner literature that proposes PoS protocols, which are typically implemented in live blockchains. That literature began with King and Nadal (2012), who put forth a hybrid PoS-PoW protocol that Peercoin implemented. Vasin (2013) and Nxt (2018) followed with pure PoS protocols implemented by Blackcoin and Nxt, respectively. This literature overlaps with the computer science consensus literature as Cardano implements Kiayias et al. (2017), and Algorand implements Chen and Micali (2016). Prominent examples of PoS protocols proposed by practitioners that await implementation include Zamfir (2017) and Buterin and Griffith (2017). All the discussed papers provide PoS implementations that differ from each other, and this paper does not seek to match any single implementation. Rather, this paper studies a chain-based PoS protocol that captures the most economically relevant aspects of an arbitrary PoS protocol.

A large literature now studies blockchain economics. This paper relates to both the literature examining economic limitations of PoW and that studying the economics of PoS. Prominent papers in the former literature include Chiu and Koeppl (2017), Budish (2018), Arnosti and Weinberg (2018), Benetton, Compiani, and Morse (2019), Biais, Bisière, Bouvard, and Casamatta (2019), Saleh (2019), Alsabah and Capponi (2020), Cong et al. (2020), and Hinzen, John, and Saleh (2020). With regard to the latter literature, Cong et al. (2020) discuss staking pools; Fanti, Kogan, and Viswanath (2020) study valuation of a PoS cryptocurrency; Irresberger et al. (2020) highlight the recent growth of PoS cryptocurrencies empirically; and Rosu and Saleh (2020) examine wealth concentration for a PoS cryptocurrency. This paper differs from the aforementioned PoS papers by focusing on generating consensus within a PoS blockchain.

1 Background

1.1 Blockchain

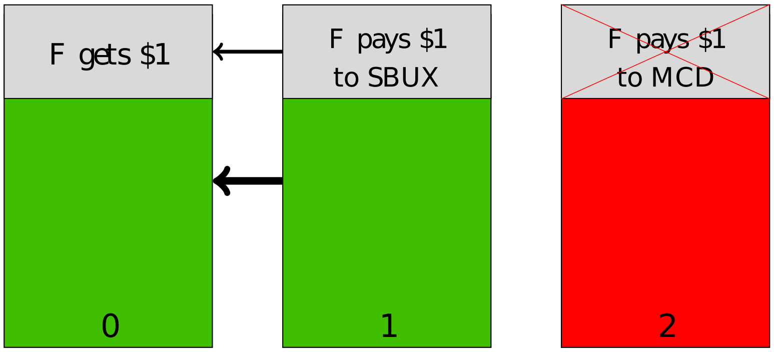

Figure 1: Blockchain with a single history

A blockchain is a virtual chain of ordered blocks. Each block contains transactions and a reference to the previous block. Any transaction in a block must be valid given the preceding set of blocks. Such validity is publicly verifiable. The blockchain receives an update only when a new block is appended. New transactions enter the blockchain only by being included in a block that enters the blockchain. Transactions on the blockchain are typically denominated in the currency of a native coin.[2]

Figure 1 offers an example of a blockchain. The first block, Block 0, consists of only a single entry that records the wealth level of some agent, F. The following block, block 1, then records an entry in which F spends F's only dollar. This second entry points back to the first entry as a manner of validating that F has the wealth to pay SBUX. If F attempts to make a subsequent $1 payment to MCD, then the blockchain will not accept that transaction

[2] For example, the currency of Bitcoin is bitcoin coins ("bitcoins"), whereas that of Ethereum is ether coins ("ether").

because F cannot validate the transaction with a previous entry within the blockchain. The blockchain in Figure 1 depicts a unique history.

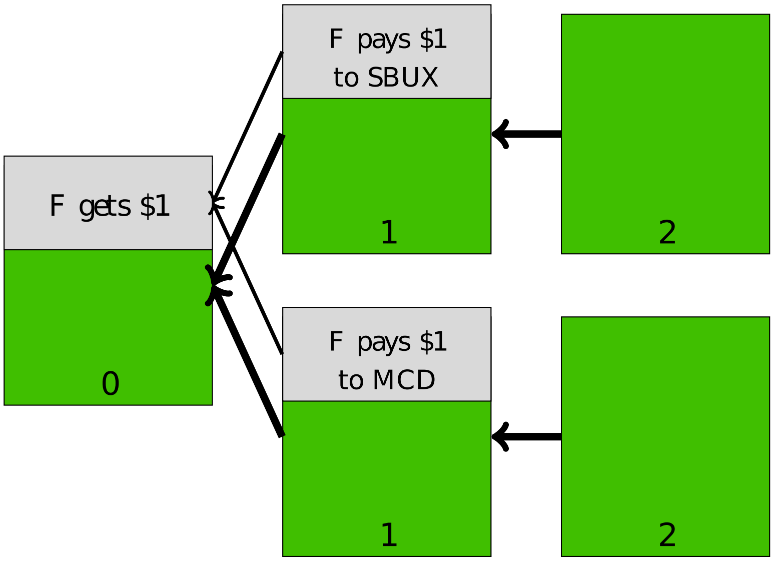

Figure 2: A forked blockchain

The blockchain community refers to a network state in which validators perceive different histories as a blockchain fork. Figure 2 depicts a fork with the same transactions as Figure 1 but two histories. On the upper branch's history, F paid SBUX $1; on the lower branch's history, F paid MCD $1. Both histories acknowledge that F had only $1 initially, so neither branch will accept the transaction from block 1 of the other branch.

1.2 Proof-of-Work

Forks may arise for a variety of reasons, and a fork may appear without intent.[3] Thus, ensuring that transactions eventually become part of a blockchain's unique accepted history necessitates having an effective consensus achievement process. To many in the blockchain community, PoW constitutes the only such viable process.

PoW dates back to Dwork and Naor (1992), Jakobsson and Juels (1999), and Back (2002). Later, Nakamoto (2008) popularized the concept across a broader audience by employing it

[3] Decker and Wattenhofer (2013) demonstrate that most Bitcoin forks arise without intent. Hinzen et al. (2020) study the related economic implications.

to allegedly achieve consensus within a permissionless blockchain. Nakamoto (2008) provides heuristic arguments whereas Biais et al. (2019) put forth a formal economic analysis of PoW consensus. Biais et al. (2019) demonstrate existence of both an equilibrium in which PoW induces consensus and an equilibrium in which PoW generates persistent disagreement. Budish (2018), Saleh (2019), and Hinzen et al. (2020) also raise concerns regarding PoW's economic viability, but those concerns arise from aspects other than consensus.

PoW extends beyond the scope of this paper, so I provide only a high-level discussion of the protocol.[4] PoW confers on a validator the authority to update the blockchain ledger on the basis of a competition to solve a trivial puzzle. Any validator solving this puzzle receives the opportunity to update the ledger. Solving the referenced puzzle involves significant computational expense, which, in turn, involves a significant economic cost. To compensate PoW validators to bear such a cost, PoW blockchains offer any validator solving the puzzle a block reward consisting of some native coins.

The referenced incentive structure has triggered a computational arms race among PoW validators. That arms race manifests in PoW blockchains expending an exorbitant level of energy as shown by de Vries (2018). This level of energy expenditure is socially costly (see Benetton et al. 2019; Saleh 2019; Cong et al. 2020) and, as documented by Irresberger et al. (2020), has contributed to a shift from PoW blockchains toward PoS recently.

1.3 Proof-of-Stake

PoS attempts to solve the energy expenditure problem created by PoW. To do so, PoS replaces PoW's competition by randomly selecting stakeholders to append to the blockchain. The simplest implementation of PoS, known as the Follow-the-Satoshi (FTS) algorithm, involves each blockchain branch selecting uniformly and randomly from the universe of native coins.[5] The owner of the selected coin receives the opportunity to append to the branch that selected

[4] Biais et al. (2019) describe the PoW protocol in depth.

[5] Xiao, Zhang, Lou, and Hou (2019) survey distributed consensus protocols and discuss the FTS algorithm within the PoS section.

her coin and simultaneously collect a block reward. This protocol succeeds in reducing energy expenditure to negligible levels, but in doing so, PoS faces an objection known as the Nothing-at-Stake problem. The remainder of this subsection provides practical context regarding PoS, a high-level description regarding the PoS protocol (as implemented by the FTS algorithm), a description of the Nothing-at-Stake Problem, and a brief discussion regarding other PoS implementations.

1.3.1 PoS in practice

King and Nadal (2012) put forth the first PoS proposal, which was implemented in the Peercoin blockchain in a hybrid scheme with PoW. PoS was later deployed as a stand-alone protocol in the Nxt blockchain. Since then, PoS has been deployed again as a stand-alone protocol in several instances. As documented by Irresberger et al. (2020), 14 PoS blockchains were announced prior to 2015 as compared to 32 PoW blockchains.[6] In recent years, the number and the relevance of PoS blockchains has grown, both in absolute terms and relative to PoW blockchains. Irresberger et al. (2020) find that over 50 PoS blockchain projects have been announced since 2015 and that deployed PoS blockchains cumulatively currently are more used than deployed PoW blockchains. This usage largely arises from decentralized applications (dApps) relating to finance, such as betting markets and decentralized exchanges. Moreover, recently launched PoS blockchains, such as Algorand, offer the potential for further applications, such as "tokenization and issuance of any type of asset . . . in a standard way" and a "faster and more secure clearing and settlement processes." These applications become feasible partially because the "proof-of-stake protocol . . . scales to billions of users without incurring significant computational or financial costs."[7]

[6] These numbers pertain to blockchains with native coins possessing in excess of US$1 million as of June 6, 2019.

[7] Sources: https://www.algorand.com/resources/blog/the-borderless-economy-is-here, https://www.algorand.com/use-cases

1.3.2 PoS protocol

The FTS algorithm partitions time into discrete periods known as time slots. The algorithm imposes that no more than one block may be added to any blockchain branch in any given slot so that no blockchain branch may grow faster than one block per the time length of a slot. During each time slot, each blockchain branch draws a unit of the native coin with each unit being equally likely to be drawn. The holder of the drawn coin then receives the opportunity to append to the associated branch. The implication of this structure is that the probability that a given player has the opportunity to append to a branch equals the proportion of coins that she owns. As in PoW, a validator appending to the blockchain is compensated with a block reward, which consists of some newly created native coins.

As a concrete example, if player A holds 8 coins and the blockchain branch has issued 10 coins to date, then player A has a $\frac{8}{10} = 80\%$ probability of receiving the opportunity to append to that branch. The random draws are independent, conditional on all information available at the time including the current coin holdings. To understand the significance of this conditioning, I extend the example to consider the next draw. For exposition and for the sake of this example only, I assume that each block possesses a block reward of one unit of the native coin and that validators always append when given the opportunity. Following the earlier discussion, player A receives the opportunity to append to the blockchain and does so with 80% probability so that player A holds $8 + 1 = 9$ coins with 80% probability after the first draw. Player A does not receive the opportunity to append to the blockchain with the complementary $100\% - 80\% = 20\%$ probability and thus holds $8 + 0 = 8$ coins after the first draw with 20% probability. Regardless of whether player A receives the opportunity to append to the blockchain, some validator receives the opportunity to append to the blockchain and exercises that option thereby receiving 1 newly circulated coin and increasing the total number of coins in circulation to $10 + 1 = 11$ after the first draw. Accordingly, after the first draw, player A's coin share equals $\frac{9}{11}$ with 80% probability and $\frac{8}{11}$ with 20% probability.

Continuing the example beyond the first time slot, the second time slot begins, and a second draw occurs. The second draw consists of a coin being selected among the now 11 coins in circulation. As with the first draw and all future draws, each coin in circulation is equally likely to be selected, and player A receives the opportunity to append if and only if one of her coins is drawn. Since player A's coin holding after the first draw is a random variable so too is the probability that player A is selected on the second draw. If player A was selected on the first draw, which occurs with $\frac{8}{10} = 80\%$ probability, then she has a $\frac{9}{11} = 81.\overline{81}\%$ probability of being selected by the second draw because she holds 9 coins in that case among a total circulation of 11 coins. In contrast, if player A was not selected on the first draw, which occurs with a $1 - \frac{8}{10} = 20\%$ probability, then she has a $\frac{8}{11} = 72.\overline{72}\%$ probability of being selected on the second draw because she holds 8 coins in that case among a total circulation of 11 coins. Unconditionally, the coin selections from the first two time slots are not independent because the first draw affects the probability of the second draw selecting player A. However, conditional on coin holdings after the first draw, the second is draw is constructed to be independent of all else and thus the probability a player is selected is independent, conditional on all information available at the time. In the event that multiple branches exist on a PoS blockchain, each branch evolves separately as previously described for a single branch.

1.3.3 Nothing-at-Stake

The Nothing-at-Stake problem assumes an initial disagreement and argues that this disagreement will persist indefinitely for a PoS protocol. More formally, the setting assumes a fork exists on the blockchain and argues that each branch of the fork will receive a new block during each time period. The fact that the blockchain branches evolve in this way is taken as persistent disagreement because multiple branches are receiving regular updates, and each branch represents a different ledger.

The argument supporting the Nothing-at-Stake problem centers around the incentive problem of a player who has received the opportunity to append a block on a branch of the blockchain. The argument highlights that a player receives a block reward if she appends the block and nothing if she does not append the block. The argument also asserts that the player faces no cost from appending to the ledger, implicitly invoking a price-taking assumption. Thus, the argument finds that appending a block to the blockchain is a weakly dominant strategy because not appending a block when given the opportunity amounts to foregoing the block reward with no off-setting gain. Since this argument asserts that appending a block is a weakly dominant strategy and since some player receives the opportunity to append to each branch during each time slot, the argument concludes that any existing branch will receive a new block in every time slot so that any initial fork will persist indefinitely.

This paper highlights that the Nothing-at-Stake problem's price-taking assumption is not appropriate for the setting. The existence of multiple blockchain branches raises doubt regarding the rightful owners of the units of the native coin because each branch corresponds to a separate ledger with different balances. In practice, merchants compensate for the risk of acquiring a coin on a branch that will not eventually become the only branch receiving updates by waiting. That waiting, in turn, undermines the ability to easily exchange the native coin and thereby lowers its value. When a player perpetuates disagreement by contributing to multiple branches being extended, this action induces delays, which in turn reduce native coin value. That value reduction is costly for holders of the native coin, and these holders are precisely the set of players that PoS grants the authority to add blocks to the blockchain. Thus, equilibrium behavior under a PoS protocol is not as suggested by the Nothing-at-Stake problem, and this paper offers a set of conditions under which PoS generates consensus in the presence of an initial disagreement.

To support the aforementioned premise that the value of a blockchain's native coin depends on the ability to exchange the native coin, I provide an example from the history of Bitcoin. In March 2013, two branches on the Bitcoin blockchain arose due to technical reasons. These branches persisted for hours, and "Mt. Gox, the leading Bitcoin exchange, announced that it was suspending Bitcoin transactions shortly afterwards." Consequently, "from a high of more than $48 earlier Monday, the value of Bitcoins plummeted to less than $37 around 10 PM Central time on Monday evening, a 23 percent decline." This example demonstrates that persistent forks undermine the ability of users to exchange the native coin and thereby impair native coin value.[8]

1.3.4 Other PoS protocols

Different than the implementation discussed within Section 1.3.2, some PoS implementations (e.g., Ethereum's Casper) combat the Nothing-at-Stake problem with explicit punishment schemes. Those implementations require the ability for one branch to detect behavior on other branches, but Brown-Cohen, Narayanan, Psomas, and Weinberg (2018) establish that detection of the Noting-at-Stake problem requires making PoS vulnerable to the Double-Spending Attack.[9] Consequently, developing a PoS incentive structure that overcomes the Nothing-at-Stake problem without explicit punishment is important, and this paper does precisely that by providing restrictions on PoS protocol design parameters that overcome the Nothing-at-Stake problem.

2 Model

In line with the previous description of PoS (Section 1.3.2) and the Nothing-at-Stake problem (Section 1.3.3), I consider the following model.

2.1 Environment

I model an extensive form game with periods $t \in \mathbb{N}$. The game involves $N \geqslant 2$ players with $N \in \mathbb{N}$ and $I \equiv \{1, ..., N\}$. Player i holds $\pi_i \in (0, 1)$ proportion of coins within the system at $t = 0$ with $\sum_{i=1}^{N} \pi_i = 1$. I let $S \in \mathbb{R}_{++}$ denote the total coin stock at $t = 0$.

[8] Source: https://arstechnica.com/information-technology/2013/03/major-glitch-in-bitcoin-network-sparks-sell-off-price-temporarily-falls-23/

[9] See Section 4 for a discussion of the double-spending attack.

2.2 PoS protocol

I assume the blockchain employs a PoS protocol. All periods $t \geqslant 1$ begin with each branch simultaneously and randomly selecting a coin from the set of coins owned by players. Any player who owns a drawn coin receives the option to append a block to the associated branch. All players receiving an option to extend a branch within any period act simultaneously during that period. Any player who receives that option in period t earns $R_t \geqslant 0$ native coins in that period if she exercises her option. R_t is referred to as the block reward, and these coins are newly created coins that are given to the block producer through a special transaction referred to as a coin-base transaction. I assume that $R \equiv \sup_{t \in \mathbb{N}} R_t < \infty$. No other activity occurs within a period, and the subsequent period commences immediately.

2.3 A fork

I assume that a fork arises at $t = 0$. No further action occurs at $t = 0$. Thus, the extant fork consists of two equally long branches at the end of $t = 0$.

2.4 Strategy space

Each player's strategy space consists entirely of whether or not that player opts to update the blockchain whenever given the option. As discussed within Section 1.3.2, each branch selects a coin from the coins in circulation as recorded by the particular branch. The owner of the selected coin then receives the option to append to the branch on which the coin was selected.[10]

More formally, player i's strategy space, $\mathcal{A}_i$, may be characterized by $\mathcal{A}_i \equiv \mathcal{A}_{1,i} \times \mathcal{A}_{2,i}$ with $\mathcal{A}_{b,i}$ denoting player i's actions when called on to act by branch b. In turn, $\mathcal{A}_{b,i} \equiv \{f_b : \mathcal{H}_{b,i} \mapsto \{0,1\}\}$ with $\mathcal{H}_{b,i}$ denoting the set of states in which branch b draws one of player i's coins. For any such state, player i must decide between one of two actions: do not add a block to

[10] The action of appending a block to a branch may be likened to voting for that branch. In that sense, the environment studied here is similar to the sequential voting environment examined by Dekel and Piccione (2000).

the branch or add a block to the branch. I represent the action of not adding a block to the branch by 0 and the action of adding a block to the branch by 1 so that f maps into $\{0, 1\}$. For any state, $h \in \mathcal{H}_{b,i}$, on which player i is given the option to append to branch b, $f_b(h) = 0$ corresponds to player i not appending to branch b in state h whereas $f_b(h) = 1$ corresponds to player i appending to branch b in state h.

On a given date t, I specify that the model's state, $h \in \mathcal{H}_{b,i}$, consists of two pieces of information. The first piece of information is the history of PoS players drawn inclusive of the current date, $\{(i_{1,s}, i_{2,s})\}_{s=1}^{t} \subseteq I^2$; $i_{b,s} \in I$ denotes the player selected by branch b in time period s. The second piece of information is the history of actions by the selected players, $\{(a_{1,s}, a_{2,s})\}_{s=1}^{t-1} \subseteq \{0, 1\}^2$; $a_{b,s} \in \{0, 1\}$ denotes whether or not the selected player appended to blockchain branch b in period s.

For concreteness, I illustrate the players' information environment at $t = 2$. At $t = 2$, the state consists of six elements. Four elements, $\{(i_{1,s}, i_{2,s})\}_{s=1}^{2}$, correspond to players that the PoS protocol selected, and two elements, $\{(a_{1,s}, a_{2,s})\}_{s=1}^{1}$, correspond to past player actions. The former consists of the players drawn by each branch in periods $t = 1$ and $t = 2$. The latter consists of the two player actions from $t = 1$. In the context of the referenced information environment, the two players selected at $t = 2$ act simultaneously, deciding whether to append to the blockchain branch. The action of not appending is represented by 0 whereas the action of appending is represented by 1, hence f_b mapping into $\{0, 1\}$.

2.5 Probability space

The model's only source of randomness arises from each branch randomly selecting a player at each time step. The probability that branch b selects player i in period t equals the proportion of coins player i holds on branch b as of the end of time $t - 1$. The proportion of coins held by player i on branch b is a discrete stochastic process affected by not only which player is given the opportunity to add a block to a branch but also whether that player actually appends a block. Thus, the proportion of coins held by player i on branch b at time t depend on not only i, b, and t but also the strategies employed by all players. Accordingly, I let $\pi^{\sigma}_{i,b,t}$ denote the proportion of coins within the system that player i holds on branch b at the beginning of period t with $\sigma \in \mathcal{A} \equiv \bigtimes_{i} \mathcal{A}_i$ denoting the strategies played by all players.

The number of coins held by player i changes only through the block reward. If player i receives the opportunity to add a block to a branch at time t and does so, then the number of coins she holds on that branch increases by R_t. The proportion of coins held by player i on a branch equals the number of coins she holds at the time divided by the total number of coins from that branch in circulation. For any branch, the number of coins in circulation is a weakly increasing sequence that increases only if a player appends a block to that branch. If a player appends a block to the branch in period t, then the total number of coins from that branch in circulation increases by R_t.

$$\pi^{\sigma}_{i,b,t} = \frac{\pi_i \times S + \sum\limits_{s=1}^{t-1} R_s A^{\sigma}_{b,s,i}}{S + \sum\limits_{s=1}^{t-1} R_s A^{\sigma}_{b,s}} \tag{1}$$

$$A^{\sigma}_{b,t} \equiv \sum_{i=1}^{I} A^{\sigma}_{b,t,i} \tag{2}$$

Equation (1) states the evolution of the proportion of coins held by player i on branch b at the beginning of period t, $\pi^{\sigma}_{i,b,t}$. $A^{\sigma}_{b,t}$ denotes an indicator of the event that player i is selected by branch b during period t and opts to append. Equation (2) defines $A^{\sigma}_{b,t}$ as an indicator of the event that branch b receives a new block in period t.

As indicated in Section 1.3.2, each blockchain branch operates independently of the other in the sense that the randomized coin selection for branch 1 is independent of that for branch 2. Formally, letting $X^{\sigma}_{b,t}$ denote the player selected on branch b at time t, $\mathbb{P}\{X^{\sigma}_{1,t} = i_1, X^{\sigma}_{2,t} = i_2 | \mathcal{F}_{t-1}\} = \mathbb{P}\{X^{\sigma}_{1,t} = i_1 | \mathcal{F}_{t-1}\} \times \mathbb{P}\{X^{\sigma}_{2,t} = i_2 | \mathcal{F}_{t-1}\} = \pi^{\sigma}_{i,1,t} \times \pi^{\sigma}_{i,2,t}$ with $\mathcal{F}_{t-1}$ denoting all information available at the end of period $t - 1$. Section 1.3.2 provides a concrete example as clarification.

2.6 Achieving consensus

Narayanan (2016, pp. 37, 173) assert that forks resolve once "one of the two [branches] is viewed as more legitimate." They assert that a branch gains legitimacy by becoming sufficiently longer than all other branches because the "chance that the shorter branch . . . will catch up to the longer branch becomes increasingly tiny as [the long branch] grows longer than any other branch." Accordingly, I assume that the blockchain achieves consensus if and only if one branch exceeds the other branch by at least length $k \in \mathbb{N}$.

I let $l_b^\sigma(t)$ denote the length of branch b at the end of period t when players play σ. I define $\Delta^\sigma(t) \equiv l_1^\sigma(t) - l_2^\sigma(t)$ so that $\Delta^\sigma(t)$ represents the gap between branches 1 and 2 at the end of period t. Then, $\tau^\sigma \equiv \inf\{t \in \mathbb{N} : |\Delta^\sigma(t)| \geqslant k\}$ represents the time at which the blockchain achieves consensus. If $\Delta^\sigma(\tau^\sigma) = k$, then I reference branch 1 as the winning branch since branch 1 possesses k more blocks than branch 2 at time τ^σ. Alternatively, if $\Delta^\sigma(\tau^\sigma) = -k$, then I reference branch 2 as the winning branch since branch 2 possesses k more blocks than branch 1 at time τ^σ.[11]

2.7 Salient strategies

I define two particular strategies before proceeding. The first strategy, referred to as the Longest Chain Rule (hereafter referenced as LCR), corresponds to a player appending only to the longest branch whenever feasible with the longest chain being defined as branch 1 whenever both branches possess the same length. The second strategy, referred to as the Nothing-at-Stake strategy (hereafter referenced as NSS), corresponds to a player appending a block whenever given the opportunity in line with the Nothing-at-Stake problem. Hereafter, I let $LCR_i^{\sigma_i}$ and $NSS_i^{\sigma_i}$ reference strategies under which player i follows σ_i until period t and LCR and NSS, respectively, thereafter. Moreover, I define $LCR_{-i}^{\sigma_{-i}} \equiv \bigtimes_{j \in I : j \neq i} LCR_j^{\sigma_j}$ and $NSS_{-i}^{\sigma_{-i}} \equiv \bigtimes_{j \in I : j \neq i} NSS_j^{\sigma_j}$.

[11] If $\tau^\sigma = \infty$, then $\Delta^\sigma(\tau^\sigma)$ is not defined, so neither branch is the winning branch.

2.8 Preferences, payoffs, and equilibrium

I specify that all players possess risk-neutral preferences with a discount factor $\delta \in (0,1)$. I assume that each coin earned on the winning branch confers on the owner one consumption unit once the blockchain achieves consensus, whereas each coin generated on a losing branch confers zero consumption value.[12]

$$V_i^{(\sigma_i,\sigma_{-i})} = \sum_{b=1}^{2}\sum_{t=1}^{\infty} R_t Y_{b,t,i}^{(\sigma_i,\sigma_{-i})} A_{b,t,i}^{(\sigma_i,\sigma_{-i})} \delta^{\tau^{(\sigma_i,\sigma_{-i})}} \mathcal{I}_{t\leqslant\tau^{(\sigma_i,\sigma_{-i})}=\tau_b^{(\sigma_i,\sigma_{-i})}} + \delta^{\tau^{(\sigma_i,\sigma_{-i})}}\pi_i S \tag{3}$$

$$Y_{b,t,i}^{(\sigma_i,\sigma_{-i})} \equiv \mathcal{I}(X_{b,t}^{(\sigma_i,\sigma_{-i})} = i) \tag{4}$$

$V_i^{(\sigma_i,\sigma_{-i})}$ constitutes the pathwise discounted payoff for player i when she plays σ_i and other players play $\sigma_{-i} \equiv \mathop{\times}\limits_{j\in I:j\neq i} \sigma_j$. I assume that each player sells her stake and consumes it on consensus. $Y_{b,t,i}^{(\sigma_i,\sigma_{-i})}$ is defined by Equation (4) and references the event that branch b selects player i at time t. Per Section 2.5, $A_{b,t,i}^{(\sigma_i,\sigma_{-i})}$ reflects player i's choice if selected by branch b in period t; $A_{b,t,i}^{(\sigma_i,\sigma_{-i})} = 1$ only if branch b selects player i's coin at time t, and player i opts to append. I let $\tau_b^{(\sigma_i,\sigma_{-i})}$ denote the period in which consensus obtains if branch b wins when player i plays σ_i and all other players play σ_{-i}.[13]

$$\sigma_i^* \in \operatorname*{arg\,sup}_{\sigma_i\in\mathcal{A}_i} \mathbb{E}[V_i^{(\sigma_i,\sigma_{-i}^*)}] \tag{5}$$

Player i's payoff equals $\mathbb{E}[V_i^{(\sigma_i,\sigma_{-i})}]$ when player i plays σ_i and all other players play σ_{-i}. Accordingly, $\{\sigma_i^*\}_{i=1}^N$ constitutes an equilibrium if σ_i^* satisfies Equation (5) with $\sigma_{-i}^* \equiv \mathop{\times}\limits_{j\in I:j\neq i} \sigma_j^*$ for all players.

[12] The analysis in this paper abstracts from the price dynamics of blockchain native coins. For an analysis of such dynamics, the interested reader should consult Cong, Li, and Wang (2019) or Fanti et al. (2020).

[13] $\tau_b^{(\sigma_i,\sigma_{-i})} \equiv \infty$ if branch b does not win.

3 Main Results

This section provides the main results of this paper in Propositions 2 and 5. Proposition 2 establishes the existence of an equilibrium in which PoS obtains consensus. Proposition 5 provides conditions under which all equilibria achieve consensus eventually with probability one.

For each branch b, time t and set of strategies σ, I define $P^{\sigma}_{b,t} \equiv \mathbb{E}[\delta^{\tau^{\sigma}-t}\mathcal{I}_{\tau^{\sigma}=\tau^{\sigma}_{b}}|\mathcal{F}_t]$ if consensus occurs after time t. If consensus has already obtained, I take $P^{\sigma}_{b,t}$ as one if branch b is the winning branch and zero otherwise. Then, $P^{\sigma}_{b,t}$ constitutes the "ex-dividend" present value of a coin on branch b at the end of time t if players play σ. I reference $P^{\sigma}_{b,t}$ as ex-dividend because PoS potentially entitles the purchaser of a coin to additional coins in probability, and I term those additional coins as dividends of the purchased coin. For simplicity, I reference the ex-dividend present value as the coin value without any qualification hereafter.

I also define $P^{\sigma}_{t} \equiv P^{\sigma}_{1,t} + P^{\sigma}_{2,t}$. P^{σ}_{t} represents the value of a coin held on both branches whereas $P^{\sigma}_{b,t}$ represents the value of a coin held only on branch b. Coins held as part of initial stake possesses value P^{σ}_{t} because such coins are held on both branches. In contrast, a coin held from a block reward earned on branch b possesses value $P^{\sigma}_{b,t}$.

Proposition 1. *Not Nothing-at-Stake*
$\forall i \in I, (\sigma_i, \sigma_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}$, *at any time* $t < \tau^{(\sigma_i,\sigma_{-i})}$:
$P_t^{(\sigma_i,\sigma_{-i})} \leqslant P_t^{(LCR_i^{\sigma_i},LCR_{-i}^{\sigma_{-i}})}$
and
$0 = P_t^{(NSS_i^{\sigma_i},NSS_{-i}^{\sigma_i})} < P_t^{(NSS_i^{\sigma_i},LCR_{-i}^{\sigma_i})} < P_t^{(LCR_i^{\sigma_i},LCR_{-i}^{\sigma_i})}$

Proposition 1 establishes that coin value obtains a maximum if all players follow LCR. This result obtains because coins derive value from their ability to be easily exchanged, which in turn depends on consensus obtaining expediently. Therefore, delaying consensus reduces coin values. As the blockchain achieves consensus at the earliest possible time when all players follow LCR, coin values achieve a maximum in this case. Proposition 1 also states that following NSS instead of LCR when all other players follow LCR strictly reduces coin value.

This finding highlights that the Nothing-at-Stake problem does not apply in general since NSS reduces coin value when all other players follow LCR. Since any player taking any action must hold native coins by PoS's construction, Proposition 1 establishes that playing NSS is costly when all other players play LCR. In Proposition 2, I demonstrate that this cost can be made sufficiently large to induce a symmetric equilibrium in which all players follow LCR.

Proposition 2. *Immediate Consensus*

If $\min_{i \in I} \pi_i \times S \geqslant \frac{R}{\delta^k(1-\delta)^2}$*, then there exists an equilibrium in which each player follows the longest chain rule. In such an equilibrium, the fork resolves at* $t = k$*.*

Corollary 3. *No Block Reward*

If $\forall t \in \mathbb{N} : R_t = 0$*, then there exists an equilibrium in which each player follows the longest chain rule.*

If a player appends to the blockchain, she receives a block reward. This block reward possesses nonnegative value, but appending to the blockchain may defer consensus and thus decrease coin value. A myopic player with no coins always appends to the blockchain when given the option if the block reward takes a strictly positive value.[14] Alternatively, a player with a large stake opts not to append to the blockchain when doing so defers consensus due to the prohibitive cost incurred via her stake being devalued. Thus, an equilibrium in which all players follow the LCR exists if each player holds a sufficient stake. Proposition 2 formalizes that result.

Adding a block to the shorter branch on the blockchain has two effects for the player appending the block. The first effect is that the player receives a block reward in terms of coins on that branch; this first effect creates an incentive for the player to append the block. The second effect is that the value of all coins falls; this second effect competes with the first effect and discourages the player from adding a block to the shorter branch. For an equilibrium to arise in which all players play LCR, the second effect must always dominate. The second

[14] A PoS protocol only selects players who have a stake in the blockchain, not those without any stake.

effect's magnitude increases in a player's coin holding and thus restricting access to updating the ledger to those with a sufficient coin holding induces the desired equilibrium.

Proposition 2 provides guidance to developers regarding designing a viable PoS blockchain. This proposition indicates that developers should restrict players with small stakes from appending to the blockchain. Fortunately, some existing PoS proposals already use similar methods. As an example, Ethereum's Casper PoS protocol proposes a minimum stake of 32 ether for validators.[15]

The condition given by Proposition 2 also may be read as a restriction on block rewards instead of a restriction on eligible stakeholders. In that context, Proposition 2 highlights that low block rewards facilitate achieving consensus within a PoS protocol. This result arises because block rewards serve as a perverse incentive to delay consensus. A player delaying consensus devalues her own wealth, but the player may incur that cost for a sufficiently high block reward from the shorter branch. This argument relies on the player holding some stake and thereby distinguishes PoS from PoW as PoW validators need not have a stake in the blockchain.

As noted, adding a block to the shorter branch has two effects. The first effect, gain of value through receipt of the block reward, encourages the player to add such a block whereas the second effect, reduction of value through coin devaluation, discourages the action. A minimum stake threshold precludes adding to the shorter branch in equilibrium by ensuring the second effect is sufficiently strong. However, developers may similarly generate an equilibrium in which all players follow LCR by ensuring that the first effect is sufficiently small. As the block reward goes to zero, the first effect vanishes so that modest block rewards also aid in delivering an equilibrium in which all players follow LCR.

Corollary 3 provides an important insight: a PoS blockchain obtains consensus without further conditions if the blockchain possesses no block reward. This result arises because a player incurs a cost for delaying consensus but receives no off-setting reward for appending to the blockchain's shorter branch. If a player refuses to append to the blockchain's longer

[15] Source: https://docs.ethhub.io/ethereum-roadmap/ethereum-2.0/proof-of-stake/.

branch when all other players play LCR, then she delays consensus and thereby reduces her own wealth. If a player appends to the blockchain's shorter branch when all other players play LCR, then she also delays consensus and thereby reduces her own wealth; a zero block reward ensures that she receives no block reward to counteract that wealth reduction. Thus, absent block rewards, an equilibrium in which all players follow LCR obtains without restricting the set of players with access to append to the blockchain.

In the context of the two effects discussed, Corollary 3 considers a case in which the first effect is entirely absent and thus the second effect dominates. A zero block reward implies that a player's gain from adding to the shorter branch is bounded above by zero when other players play LCR. Moreover, in that case, adding to the shorter branch reduces coin value. Thus, since a PoS protocol requires a strictly positive coin holding for any player given the opportunity to add a block, a zero block reward implies that payoffs strictly decrease when a player appends to the shorter branch.

Nxt blockchain employs a PoS protocol with a zero block reward. Nxt's online blockchain explorer provides no indication of prolonged forks.[16] Moreover, Nxt's developers assert, "the Nxt network does not experience long blockchain forks, and the low block reward does not provide a strong profit incentive" so that the Nothing-at-Stake strategy is "currently not practical."[17] The results of this paper are consistent with the aforementioned facts.

Proposition 4. *Subgame Perfection*

The LCR described within Proposition 2 constitutes a subgame perfect equilibrium.

Proposition 4 strengthens Proposition 2 by establishing that the equilibrium in which all players play LCR constitutes a subgame perfect equilibrium. This result arises because stake becomes a more powerful incentive as a possible fork resolution draws near. Accordingly, working to resolve the fork initially (via LCR) implies incentive compatibility to follow through and the LCR equilibrium does not require a commitment device.

[16] See https://nxtportal.org/monitor/.

[17] Source: https://nxtwiki.org/wiki/Whitepaper:Nxt#Nothing_at_Stake

The aforementioned results establish existence of an equilibrium in which consensus obtains under PoS. Those results, however, do not preclude the existence of another equilibrium in which a fork persists indefinitely. I therefore turn next to studying conditions that preclude such an equilibrium.

Proposition 5. *Eventual Consensus*
Let $\sigma^* \equiv \{\sigma_i^*\}_{i=1}^N \in \mathcal{A}$ *denote an equilibrium. If* $\sum_{t=1}^{\infty} R_t < \infty$*, then* $\mathbb{P}\{$*Consensus Never Obtains*$\} \equiv \mathbb{P}\{\tau^{\sigma^*} = \infty\} = 0$.

Proposition 5 establishes that a sufficiently modest block reward schedule precludes a persistent forking equilibrium. This result distinguishes PoS from PoW as a similar reward structure within PoW does not preclude a persistent forking equilibrium.

Biais et al. (2019) study forking within a PoW protocol and establish the existence of a persistent forking equilibrium. Their result arises due to different validators acquiring block rewards on different branches. When a validator acquires block rewards on a given branch, that validator possesses an economic incentive to perpetuate the branch because perpetuating the branch helps preserve the value of block rewards on that branch. Similarly, when one set of validators accrues block rewards on one branch and another set of validators accrues block rewards on a different branch, then each set possesses an incentive to perpetuate the branch on which it earned block rewards. Biais et al. (2019) demonstrate that these incentives can become strong enough in equilibrium such that each set propagates a different branch perpetually so that a fork persists indefinitely.

The referenced persistent PoW fork arises due to block rewards. Proposition 5 precludes such a fork within PoS by minimizing the role of block rewards. The design choice of low block rewards precludes persistent forks within a PoS setting, but not within a PoW setting. To intuit this distinction between PoS and PoW, I contrast the case of zero block rewards within PoS with that of zero block rewards within PoW. Zero block rewards within a PoS protocol imply that a PoS validator's objective function entirely consists of the value of her stake, which in turn depends on coin value. Formally, invoking Equation (3), $\forall t : R_t = 0$ implies

$\mathbb{E}[V_i^{(\sigma_i,\sigma_{-i})}] = P_t^{(\sigma_i,\sigma_{-i})}\pi_i S$. Since forks erode coin value, the referenced incentive structure induces each validator to work to resolve forks within a PoS protocol. That desire for each validator to resolve forks, in turn, precludes persistent forks as demonstrated by Proposition 5. I contrast PoW with PoS: since PoW does not require validators to hold coins, zero block rewards imply that validator payoffs do not depend on coin value within a PoW setting. Thus, in the PoW setting, validators have no particular interest in working to resolve forks so that a persistent forking equilibrium may yet arise.[18]

This section provides important economic guidance to PoS developers. To generate consensus more easily, PoS protocols should impose minimum stake requirements and maintain low block reward schedules. This guidance does not apply to PoW protocols.[19]

The Internet Appendix extends the aforementioned analysis by allowing trading prior to $t = 0$. Allowing trading allows free entry among players and renders the initial coin shares endogenous. All results discussed within this section qualitatively hold under the referenced extension.

4 Additional Discussion

This section discusses additional concerns regarding PoS aside from the Nothing-at-Stake problem and clarifies why these concerns do not preclude the viability of PoS. Section 4.1 explains why PoS validators cannot commit fraud and transfer to themselves coins from other users on the blockchain. Section 4.2 highlights that PoS, contrary to conventional wisdom, does not generate wealth concentration. Section 4.3 provides guidance for insulating PoS coins from double-spending attacks and highlights that such guidance is consistent with practice. Section 4.4 discusses staking pools.

[18] Formally, defining $G(\cdot)$ such that $\forall M : G(M) = 0$ in the model of Biais et al. (2019) implies zero payoffs for all validators regardless of strategy and thus admits a persistent forking equilibrium.

[19] In contrast, Chiu and Koeppl (2017) and Budish (2018) highlight that low rewards could be problematic for a PoW blockchain.

4.1 Validator fraud

Like PoW blockchains, PoS blockchains employ a technical solution to preclude validators from transferring coins from other users to themselves. This technical solution is known as a Digital Signature Scheme (DSS). The remainder of this section provides context regarding DSS and explains how this system precludes validators from fraudulently stealing coins from users.

The U.S. Department of Homeland Security (DHS) describes DSS as "a mathematical algorithm routinely used to validate the authenticity and integrity of [data] (e.g., an email, a credit card transaction, or a digital document)." DHS notes that DSSs "increase the transparency of online interactions and develop trust between customers, business partners, and vendors."[20] DSSs are widely used in practice, well beyond blockchain settings. Subsequently, I provide some technical details of DSS and clarify how these details prevent validators from stealing funds on a blockchain.

A DSS consists of three functions: a generation function, a signing function and a verification function. The generation function enables an individual to create an identity within the DSS system. The signing function enables an individual who has generated an identity to produce a digital signature of any data associated with the aforementioned identity. The verification function enables a third party to verify that a given digital signature is actually associated with particular data and a particular identity.

Within a DSS, an identity consists of two components: a public key and a secret key. The public key is public information and serves as a public identifier for the identity. In contrast, the secret key should be kept private. The need to keep the secret key private arises because the signing function takes the secret key as an input so that knowledge of the secret key enables any individual to digitally sign for that identity.

The signing function takes two inputs and produces a digital signature. The two inputs are arbitrary data and a secret key. The data input is arbitrary to allow the user freedom to

[20] Source: https://www.us-cert.gov/ncas/tips/ST04-018.

convey any particular content as having come from the user. The second input is the secret key to enable the user that generated the identity to produce a digital signature. As discussed, the user should keep the secret key private so that no other user may also digitally sign with the user's identity.

The verification function takes three inputs and outputs either $True$ or $False$. The three inputs are a public key, arbitrary data, and a digital signature. A third party may verify whether a specific digital signature corresponds to a specific identity digitally signing particular data by using the identity's public key, the particular data and the specific digital signature as inputs to the verification function. If the verification function outputs $True$, then the third party may safely consider the digital signature as evidence that the specific identity's secret key was used to sign the particular data to produce the digital signature. Assuming that this private key is known only to the user that generated the identity, the third party may also safely conclude that the user who generated the identity was responsible for signing the particular data.

The DSS is useful for a blockchain precisely because it precludes one user from stealing the funds of another user. On a blockchain, each user possesses an identity in a DSS system. Blockchain coin balances are held under these identities. Moreover, for any transaction to end up on the blockchain, the sender of that transaction must sign the transaction. As discussed, since any third party may verify whether this signature occurred properly, even validators cannot steal funds of a user. If a validator attempts to include a transaction in which she steals the funds of another user on a block, that transaction would not verify under the DSS, and all other users and validators would reject the block as invalid for that reason.

For concreteness, I offer an example. Assume that there exists a balance of \$100 on the blockchain under an identity that corresponds to public key pk and secret key sk. If a validator seeks to gain that \$100, this validator must include a transaction in which this \$100 is transferred to the validator. As the balance is held by the identity associated with pk as the public key, this transaction must also involve a digital signature from the identity associated with pk. As noted, such a digital signature requires the usage of the signing function, which, in turn, requires the identity's secret key, sk, as an input. As sk is known only to the user that generated the identity, the validator cannot use this as an input and thus cannot produce the digital signature necessary to transfer the \$100 to the validator. As such, the validator cannot steal the \$100.

4.2 Wealth concentration

Some detractors of PoS argue that it leads to wealth concentration. For example, one editorial asserts, "PoS systems [lead] to a rich-get-richer effect, causing dramatic concentration of wealth."[21] However, Rosu and Saleh (2020) study wealth concentration within a PoS blockchain and demonstrate that, contrary to the aforementioned assertion, PoS does not induce wealth concentration. Subsequently, I provide some intuition for that finding.

Rosu and Saleh (2020) highlight that PoS does not cause a rich-get-richer effect; rather, PoS wealth shares neither rise nor fall in expectation. To understand that finding, one must recognize that while PoS is likely to select a validator with a large wealth share, such a selection would not increase the wealth share of the validator as much as not being selected would reduce that validator's wealth share. Incorporating both the probability that a validator is selected and how such a selection or nonselection affects the validtor's wealth share leads to the conclusion that PoS wealth shares neither rise nor fall in expectation.

To solidify this point, I revisit the example from Section 1.3.2. In that example, player A begins with 8 of 10 native coins and thus has an overwhelming $\frac{8}{10} = 80\%$ probability of being selected by the PoS protocol. If selected, she earns the right to receive the block reward, which is set to one for the example. As such, if player A is selected and appends to the blockchain, then her coin share increases by less than 2% to $\frac{8+1}{10+1} = \frac{9}{11} = 81.\overline{81}\%$. In contrast, if player A is not selected, then her wealth share falls by more than 7% to $\frac{8+0}{10+1} = \frac{8}{11} = 72.\overline{72}\%$. In this example, player A's expected wealth share equals $\frac{8}{10} \times \frac{9}{11} + \frac{2}{10} \times \frac{8}{11} = \frac{8}{10}$, which is also player

[21] Source: https://coingeek.com/proof-work-vs-proof-stake/.

A's initial wealth share. Thus, player A's wealth share neither rises nor falls on average. Rosu and Saleh (2020) demonstrate this stability property holds in general for a PoS blockchain, thereby undermining the notion that PoS induces wealth concentration.[22]

4.3 Double spending

Nakamoto (2008) argues that PoW overcomes the double-spending attack under specific conditions. Some argue that PoS must meet a similar standard to be deemed viable. This section demonstrates that PoS does meet such a standard. Section 4.3.1 discusses the double-spending attack and the associated analysis of Nakamoto (2008), clarifying the sense in which Nakamoto (2008) demonstrates that PoW overcomes the double-spending attack. Section 4.3.2 extends the model of Section 2 and provides an analysis of double-spending attacks for PoS. This analysis provides two key results, Propositions 6 and 7. Proposition 6 establishes that PoS overcomes the double-spending attack in a similar sense that Nakamoto (2008) demonstrates PoW overcomes the double-spending attack. Proposition 7 moves beyond that sense, establishing that there exist conditions under which a double-spending attack is not profitable on a PoS blockchain. Section 4.3.3 highlights that measures taken in practice align with the analysis provided in Nakamoto (2008) and this paper. Appendix B studies a similar attack known as the Long-Range Attack and provides conditions under which this attack is not profitable.

4.3.1 PoW and double spending

A double-spending attack refers to an instance in which a malicious agent, referred to as an attacker, uses native coins to purchase a physical good but then attempts to reverse the transaction to regain possession of those coins. If successful, the double-spending attack yields the attacker both the physical good purchased and the coins.

Nakamoto (2008) envisioned a double-spending attack as follows. First, an attacker and a merchant agree to an exchange in which the attacker pays the merchant in native coins and

[22] Formally, Rosu and Saleh (2020) demonstrate that PoS wealth shares exhibit a martingale property.

the merchant gives the attacker a physical good. As one leg of the exchange, the merchant and attacker put a transaction on the blockchain in which the attacker transfers native coins to the merchant. As the second leg, the merchant transfers physical possession of the good to the attacker. Once the second leg is complete, the attacker needs to regain possession of the coins to successfully complete the attack. Nakamoto (2008) assumes that all validators follow LCR. Therefore, regaining possession of the coins requires the attacker to generate a blockchain branch, which omits her payment to the merchant and is longer than the main blockchain branch. This new branch must be longer than the main blockchain branch because LCR implies that validators will then follow the new branch thus making the new branch the accepted transaction history. The new branch must omit the payment to the merchant because that omission effectively erases the transaction and thereby returns possession of the coins to the attacker.

Nakamoto (2008) does not argue that arbitrary behavior on the part of the merchant overcomes the double-spending attack. Rather, he requires that the merchant "wait" to turn over physical possession of a good. Nakamoto (2008) states that the merchant wait "until the transaction has been added to a block and z blocks have been linked after it" because "the probability [the attack succeeds] drop[s] off exponentially with z" under certain conditions. Under the analysis of Nakamoto (2008), the double-spending attack always succeeds with some probability, but the attack's success probability can be made arbitrarily small as the merchant waits sufficiently long. This ability to generate arbitrarily small attack success probabilities is the standard by which Nakamoto (2008) demonstrates that PoW overcomes the double-spending attack. Section 4.3.2 demonstrates that not only does PoS attain this standard but also the attack becomes unprofitable under certain conditions.

4.3.2 PoS and double spending

This section extends the model of Section 2 and examines the double-spending attack on a PoS blockchain. The associated analysis provides conditions under which this attack is overcome in the sense given by Nakamoto (2008) and then conditions under which the attack is not profitable.

I assume that the fork discussed within Section 2 has resolved according to the equilibrium discussed within Proposition 2. At that point, the validators discussed within Section 2 sell their stakes to new validators that enter the model. I assume that these new validators follow LCR akin to the analysis of Nakamoto (2008). As the fork has resolved, there exists only one branch initially within this section's analysis.

As with the model of Nakamoto (2008), this section's extension includes an attacker. The attacker selects a number of native coins to purchase and then selects an amount of those coins to sell in an exchange with a merchant for a physical good. The purchase requires the attacker to place a transaction on the blockchain that sends the merchant some coins. Per Nakamoto (2008), I allow that the merchant waits for "z blocks [to] have been linked after [the transaction]" before transferring possession of the physical good. This wait implies that any new branch omitting the transaction must lag the initial branch by z blocks at first. Once the attacker receives the physical good, she then mounts a double-spending attack by creating a new branch, which omits the transaction. The new branch lags the initial branch by z blocks at first, but the attacker seeks to extend the branch to become at least as long as the initial branch. If the new branch ever becomes at least as long as the initial branch, then, per LCR, all validators switch to adding blocks only to the new branch and thus the double-spending attack has succeeded. In that case, the attacker regains possession of the coins used in the exchange.

Each blockchain branch evolves as discussed within Section 1.3.2 and as modeled within Section 2. In each period, each blockchain branch selects a coin uniformly among all native coins. The owner of the selected coin then receives the opportunity to add a block to the associated branch. This structure implies that the probability an individual receives the opportunity to add a block to a blockchain branch equals the proportion of coins she holds on that branch. Akin to the analysis of Nakamoto (2008), all validators follow LCR. As the attacker seeks to execute a double-spending attack, she attempts to extend the length of the new branch beyond that of the old branch and thus adds blocks to the new branch whenever possible but never adds blocks to the old branch.

For exposition, I assume that the blockchain provides no block rewards. This assumption simplifies the mathematical analysis. Also, for exposition, I specify the analysis in terms of the proportion of native coins held by the attacker rather than the number of such coins.

I incorporate financing costs and financing constraints. I assume that the attacker must finance her purchase of coins by paying interest rate $r > 0$ per unit time. I also assume that the attacker faces a borrowing limit, $B > 0$.

$$\begin{aligned}
&\max_{\rho,\rho'} \rho' M \times \mathbb{P}(T < \infty) - r \times \mathbb{E}[T] \times \rho M - \rho(M - M') \\
&s.t. \\
&\qquad 0 \leqslant \rho \leqslant 1 \\
&\qquad 0 \leqslant \rho' \leqslant \rho \\
&\qquad \rho M \leqslant B
\end{aligned} \tag{6}$$

Within the described setting, Problem 6 provides the attacker's incentive problem. The attacker initially purchases $\rho \in [0, 1]$ proportion of the native coins. Those coins cost her ρM with $M > 0$ denoting the market capital of the native coin before the attack. The attacker finances this purchase by borrowing at interest rate $r > 0$ per unit time, which implies that she bears expected interest costs $r \times \mathbb{E}[T] \times \rho M$ over the duration of the attack with T denoting the random attack duration. The attacker's borrowing and thus maximal coin holding is restricted by a borrowing limit, $B > 0$. The attacker transfers $\rho' \in [0, \rho]$ proportion of coins to a merchant as part of a mutual exchange. This transfer implies that the attacker possesses $\rho - \rho'$ proportion of coins on the initial blockchain branch but ρ proportion of coins on any new branch that omits the transaction. In return for the coins, the merchant transfers to the attacker possession of a physical good worth the $\rho' M$ paid by the attacker. As discussed, the attack consists of the attacker creating a new branch, which omits her transaction to the merchant and seeking to extend that branch to become as long as the initial branch. If the attacker succeeds in her task (i.e., $T < \infty$), the new branch becomes accepted by all validators and thus the attacker effectively regains possession of the ρ' proportion of coins she paid the merchant. In such a case, the attacker possesses $(\rho - \rho') + \rho' = \rho$ proportion of coins on the branch all validators follow and thus the attack has succeeded. If such a point ever occurs, the attacker sells her full coin holdings for $\rho M'$ and repays the principal of ρM from her borrowing with M' denoting the market value of native coins after initiating the attack. Following the analysis from previous sections, $M' \leqslant M$ because the attack involves creating disagreement on the blockchain although the results of this section hold even when $M' = M$.

The expected profit from the attack equals $\rho' M \times \mathbb{P}(T < \infty) - r \times \mathbb{E}[T] \times \rho M - \rho(M - M')$. The first term represents the expected value of the physical good the attacker receives. This gain is incurred only if the attack succeeds; otherwise, the gain is offset by the purchase of the native coins used to acquire the good. The second term represents the interest cost of the attack. The third term represents the difference between the value of coins regained from the attack and their value when initially purchased to execute the attack. All results hold when $M' = M$ in which case the third term disappears.

A key economic point within this analysis is that financing costs, $r > 0$, and financial constraints, $B > 0$, serve as key drivers to protect a PoS blockchain from a double-spending attack. This contrasts with a PoW blockchain, which is protected from the same attack primarily by computational costs. More precisely, Nakamoto (2008) demonstrates that PoW overcomes the double-spending attack in the sense that the probability of a successful attack approaches zero as z diverges under the condition that the validator network possesses more computational power than the attacker. Proposition 6 provides a parallel result for PoS, thereby demonstrating that PoS overcomes the double-spending attack in a sense similar to PoW.

Proposition 6. *PoS Overcomes Double-Spending Attack*

If $M > 2B$, then the double-spending attack succeeds with vanishing probability as z diverges (*i.e.*, $\lim_{z \to \infty} \mathbb{P}(T < \infty) = 0$).

Proposition 6 highlights that a sufficiently high market capital ensures that the probability of a successful double-spending attack approaches zero as z diverges. This result arises because the attack succeeding requires that the new branch become at least as long as the initial branch. The new branch advances relative to the initial branch in a given period only if the attacker is selected by both branches because other validators follow LCR and therefore add only to the initial branch. The attacker's likelihood of being selected on a given branch equals her proportion of coins on that branch, so she must hold a large proportion of coins on both branches for the attack to succeed. To acquire a large proportion of coins, the attacker must borrow funds that equate with a commensurately large share of the coin's market capital. However, as the attacker faces a borrowing constraint, she cannot acquire such a large proportion of coins if the market capital is sufficiently large. Accordingly, a sufficiently large market capital enables a PoS blockchain to overcome the double-spending attack just as sufficiently high computational power enables a PoW blockchain to overcome the double-spending attack.

To better understand the aforementioned result, I consider an example with market capital sufficiently high as indicated by Proposition 6. In particular, I suppose that the market capital, M, is sufficiently high that the borrowing limit, B, is only 1% of it (i.e., $\frac{B}{M} = .01$). Then the attacker cannot acquire more than 1% of coins on the initial branch of the blockchain (i.e., $\rho \leqslant \frac{B}{M} = .01$). In such a case, the attacker holds no more than 1% of coins on each branch since her proportion on the initial branch equals $\rho - \rho' \leqslant \rho \leqslant .01$ and her proportion on the new branch equals $\rho \leqslant .01$. Accordingly, the probability she is selected by both branches is no more than $1\% \times 1\% = .01\%$. The new branch advances on the initial branch only if the attacker is selected on both branches, so the last calculation implies that the new branch advances on the initial branch in a period with probability not exceeding .01%. In contrast, the probability that the attacker is selected by neither branch is at least $99\% \times 99\% = 98.01\%$. As validators play LCR and hence add blocks only to the initial branch, the last calculation implies that the initial branch extends its lead on the new branch in a given period with probability at least 98.01%. As the initial branch extends its lead on the new branch with overwhelming probability ($\geqslant$ 98.01%) while the new branch reduces that lead with infinitesimal probability ($\leqslant$ .01%), the new branch is unlikely to ever catch up to the initial branch. Moreover, as demonstrated by Proposition 6, the likelihood that the new branch ever catches up decays to zero as the merchant waits longer (i.e., as $z \to \infty$) before transferring possession of the physical good.

Proposition 7. *PoS Double Spending Not Profitable*
For a sufficiently large z, the double-spending attack is not profitable.

Although the probability that an attempted double-spending attack succeeds may not be made arbitrarily small if the market capital is low (i.e., if $M \leqslant 2B$), it is still nonetheless not profitable for a sufficiently large z and thus will not arise in equilibrium given that merchants take appropriate caution.[23] Proposition 7 formalizes that assertion. This result arises because larger values of z increase the attack duration, which in turn increases the cost of the attack. For a sufficiently large z, the attack becomes sufficiently prolonged that it is no longer profitable.

An increase in z increases the attack duration because the new branch begins behind by at least z blocks and thus must overcome a larger deficit against the main branch as z increases. The attacker must maintain her position in native coins for the attack duration and that position requires financing through interest payments. Accordingly, interest costs increase with the attack duration so that a sufficient duration renders the attack unprofitable.

The attack duration is endogenous and depends on not only z but also the attacker's coin holding. As the attacker selects her coin holding, she may reduce the attack duration through that choice. The attack duration monotonically decreases in her coin holding because a larger coin holding makes the attacker more likely to be selected by the PoS protocol. Thus, the attacker could increase her coin holding to reduce the attack duration. However, for sufficiently

[23] Formally, the meaning of unprofitable is that the attacker's payoff is not positive.

large z, this choice fails to make the attack profitable because a larger coin holding requires a higher borrowing principal, which, in turn, induces a higher interest cost. For sufficiently large z, this interest cost is prohibitively high, and the double-spending attack is not profitable.

4.3.3 Double spending in practice

Double-spending attacks are a concern for entities that trade blockchain assets. However, the risk posed by such attacks do not preclude such entities from trading PoS coins. Rather, these entities insulate themselves from these attacks by following a policy consistent with the guidance given within Section 4.3.2. As such, the potential for double-spending attacks does not preclude the viability of PoS.

As cryptocurrency exchanges constitute the largest traders of blockchain assets and publicly provide their trading policies, I use these entities as an example to highlight the typical policy that entities employ to protect themselves from double-spending attacks. Cryptocurrency exchanges "avoid the risks of double spending" by requiring a transaction "receive a number of confirmations on its blockchain" before the exchange takes the transaction as settled.[24] The number of confirmations refers to the number of blocks linked on top of the initial transaction and thus corresponds to z within the analysis of both Nakamoto (2008) and Section 4.3.2 so that the described policy aligns with the guidance provided by Section 4.3.2.

The aforementioned policy for avoiding the risks of double-spending applies to both PoS and PoW coins. For example, the cryptocurrency exchange Bittrex sets $z = 8$ for PoS coin Nxt and $z = 2$ for PoW coin bitcoin.[25] The interested reader may consult Irresberger et al. (2020) for further detail regarding the values of z across coins and exchanges.

[24] This quote comes from the website of a major exchange, Kraken.
Source: https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times.

[25] Source: https://bittrex.com/api/v1.1/public/getcurrencies.

4.4 Staking pools

Recently, the notion of staking pools has become more popular among PoS blockchains. A staking pool is an organization in which individual stakeholders partner together and share rewards from validation. Cong et al. (2020) and Lehar and Parlour (2020) study an analogous phenomenon for PoW blockchains. Lehar and Parlour (2020) use Bitcoin data to argue that this phenomenon facilitates collusion among Bitcoin miners, whereas Cong et al. (2020) theoretically demonstrate that it "severely escalate[s] the arms race in PoW blockchains." Lehar and Parlour (2020) do not relate their findings to staking pools, whereas Cong et al. (2020) note that their analysis applies to "PoS systems equally well . . . except that in PoS the consensus generation process does not necessarily incur a high energy consumption."

The PoW arms race involves PoW validators increasing energy expenditure to increase their likelihood of acquiring PoW block rewards. The analog of this phenomenon for PoS blockchains is that PoS validators would increase their PoS native coin holdings to increase their likelihood of acquiring PoS block rewards. Thus, the PoW arms race translates not to increased energy expenditure for PoS but rather to a positive demand shock that puts upward pressure on the market capital of a PoS coin.

Empirically, PoS coins have exhibited growth in market capital relative to PoW coins (see Irresberger et al. 2020). Although it is difficult to tie that growth to staking pools, some of that growth has come from a breed of PoS protocols that generate pooling by the construction of the protocol. Specifically, a breed of PoS protocol known as Delegated Proof-of-Stake (DPoS) has recently become popular. In DPoS, a small group of stakeholders, known as delegates, serve as the validators for the PoS blockchain. These delegates are elected by the entire population of stakeholders. Although the delegates may be replaced, the set of delegates exhibits little variability over time as might be expected of the industrial organization of staking pools.

While formal modeling of staking pools is beyond the scope of this paper, this paper's results suggest that staking pools could strengthen the security of a PoS blockchain. As noted, the analysis of Cong et al. (2020) suggests that staking pools would increase demand for a PoS coin. In turn, that increased demand would generate an increase in the market capital of the PoS coin. Then, given Proposition 6, such an increase in market capital would translate to better security against the double-spending attack for a PoS blockchain.

5 Conclusion

This paper provides the first formal economic analysis of PoS consensus. I provide conditions under which PoS generates consensus. Therefore, my work highlights that developers may implement a viable permissionless blockchain without prohibitive energy consumption documented within de Vries (2018) and Mora, Rollins, Taladay, Kantar, Chock, Shimada, and Franklin (2018).

This paper establishes two design choices that PoS developers may employ to generate consensus within PoS blockchains. One such design choice is that developers may impose a minimum stake threshold for validators. A minimum stake threshold restricts the set of validators that may update the blockchain to those holding at least some minimum amount of native coins. The other design choice is that developers may impose a modest block reward schedule. A block reward refers to the number of native coins given to a validator for updating the blockchain with a new block.

Even setting aside exorbitant energy expenditure levels, Hinzen et al. (2020) raise concerns regarding the economic viability of PoW blockchains. Such concerns highlight the need for research regarding the economic viability of non-PoW protocols. This paper begins to fill that need.

References

Alsabah, H., and A. Capponi. 2020. Pitfalls of Bitcoin's Proof-of-Work: R&D Arms Race and Mining Centralization. *Working Paper* .

Arnosti, N., and S. M. Weinberg. 2018. Bitcoin: A Natural Oligopoly. *CoRR* abs/1811.08572. URL http://arxiv.org/abs/1811.08572.

Back, A. 2002. Hashcash - a denial of service counter- measure. *http://www.hashcash.org/papers/hashcash.pdf* .

Benetton, M., G. Compiani, and A. Morse. 2019. CryptoMining: Local Evidence from China and the US. *Working Paper* .

Biais, B., C. Bisière, M. Bouvard, and C. Casamatta. 2019. The Blockchain Folk Theorem. *Review of Financial Studies* 32(5):1662–1715.

Brown-Cohen, J., A. Narayanan, C. Psomas, and S. M. Weinberg. 2018. Formal Barriers to Longest-Chain Proof-of-Stake Protocols. *CoRR* abs/1809.06528. URL http://arxiv.org/abs/1809.06528.

Budish, E. 2018. The Economic Limits of Bitcoin and the Blockchain. *NBER Working Paper* .

Buterin, V., and V. Griffith. 2017. Casper the Friendly Finality Gadget. *CoRR* abs/1710.09437. URL http://arxiv.org/abs/1710.09437.

Carlsten, M., H. Kalodner, S. M. Weinberg, and A. Narayanan. 2016. On the Instability of Bitcoin Without the Block Reward. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* pp. 154–167.

Chen, J., and S. Micali. 2016. ALGORAND: The Efficient and Democratic Ledger. *CoRR* abs/1607.01341. URL http://arxiv.org/abs/1607.01341.

Chiu, J., and T. Koeppl. 2017. The Economics of Cryptocurrencies – Bitcoin and Beyond. *Working Paper* .

Chiu, J., and T. Koeppl. 2019. Blockchain-based Settlement for Asset Trading. *Review of Financial Studies* 32:1716 – 1753.

Cong, L. W., and Z. He. 2019. Blockchain Disruption and Smart Contracts. *Review of Financial Studies* 32:1754–1797.

Cong, L. W., Z. He, and J. Li. 2020. Decentralized Mining in Centralized Pools. *Review of Financial Studies* Forthcoming.

Cong, L. W., Y. Li, and N. Wang. 2019. Tokenomics: Dynamic Adoption and Valuation. *Working Paper* .

Daian, P., R. Pass, and E. Shi. 2016. Snow White: Provably Secure Proofs of Stake. Cryptology ePrint Archive, Report 2016/919. http://eprint.iacr.org/2016/919.

de Vries, A. 2018. Bitcoin's Growing Energy Problem. *Joule* 2:801–805.

Decker, C., and R. Wattenhofer. 2013. Information Propagation in the Bitcoin Network. *IEEE P2P 2013 Proceedings* .

Dekel, E., and M. Piccione. 2000. Sequential Voting Procedures in Symmetric Binary Elections. *Journal of Political Economy* 108:34–55. URL https://doi.org/10.1086/262110.

Dwork, C., and M. Naor. 1992. Pricing via processing or combatting junk mail. *In 12th Annual International Cryptology Conference* pp. 139–147.

Eyal, I., and E. G. Sirer. 2014. Majority is not enough: Bitcoin mining is vulnerable. In *Eighteenth International Conference on Financial Cryptography and Data Security (FC'14)*.

Fanti, G., L. Kogan, and P. Viswanath. 2020. Economics of Proof-of-Stake Payment Systems. *Working Paper* .

Hinzen, F., K. John, and F. Saleh. 2020. Bitcoin's Fatal Flaw: Proof-of-Work's Limited Adoption Problem. *NYU Stern Working Paper* .

Irresberger, F., K. John, and F. Saleh. 2020. The Public Blockchain Ecosystem: An Empirical Analysis. *NYU Stern Working Paper* .

Jakobsson, M., and A. Juels. 1999. Proofs of Work and Bread Pudding Protocols. In *Proceedings of the IFIP TC6/TC11 Joint Working Conference on Secure Information Networks: Communications and Multimedia Security*, CMS '99, pp. 258–272. Deventer, The Netherlands, The Netherlands: Kluwer, B.V. URL http://dl.acm.org/citation.cfm?id=647800.757199.

Kiayias, A., A. Russell, B. David, and R. Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pp. 357–388. Springer.

King, S., and S. Nadal. 2012. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. White paper: https://peercoin.net/assets/paper/peercoin-paper.pdf.

Lamport, L., R. Shostak, and M. Pease. 1982. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems* 4/3:382–401. URL https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/.

Lehar, A., and C. Parlour. 2020. Miner Collusion and the BitCoin Protocol. *Working Paper* .

Ma, J., J. Gans, and R. Tourky. 2019. Market Structure in Bitcoin Mining. *Rotman School of Management Working Paper* .

Miller, A., and J. J. LaViola. 2014. Anonymous Byzantine Consensus from Moderately-Hard Puzzles: A Model for Bitcoin. *http://nakamotoinstitute. org/research/anonymous-byzantine-consensus* .

Mora, C., R. L. Rollins, K. Taladay, M. B. Kantar, M. K. Chock, M. Shimada, and E. C. Franklin. 2018. Bitcoin emissions alone could push global warming above 2 C. *Nature Climate Change* 8:931–933.

Nakamoto, S. 2008. Bitcoin: A peer-to-peer electronic cash system. *https://bitcoin.org/bitcoin.pdf* .

Nayak, K., S. Kumar, A. Miller, and E. Shi. 2015. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. Cryptology ePrint Archive, Report 2015/796. http://eprint.iacr.org/2015/796.

Nxt. 2018. Whitepaper:Nxt. White paper: https://nxtwiki.org/wiki/whitepaper:nxt.

Rosu, I., and F. Saleh. 2020. Evolution of Shares in a Proof-of-Stake Cryptocurrency. *Management Science* Forthcoming.

Saleh, F. 2019. Volatility and Welfare in a Crypto Economy. *Working Paper* .

Vasin, P. 2013. BlackCoin's Proof-of-Stake Protocol v2. White paper: https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf.

Xiao, Y., N. Zhang, W. Lou, and Y. T. Hou. 2019. A Survey of Distributed Consensus Protocols for Blockchain Networks. *CoRR* abs/1904.04098. URL http://arxiv.org/abs/1904.04098.

Zamfir, V. 2017. Casper the Friendly Ghost A Correct-by-Construction Blockchain Consensus Protocol. White paper: https://github.com/ethereum/research/blob/master/papers/caspertfg/caspertfg.pdf.

Appendices

Appendix A. Proofs

A.1 Proof of Proposition 1

Proof. $\forall \omega \in \Omega, i \in I, (\sigma_i, \sigma_{-i}) \in \mathcal{A}_i \times \mathcal{A}_{-i}$:

$t < \tau^{(\sigma_i,\sigma_{-i})}$ implies $\tau^{(\sigma_i,\sigma_{-i})} \geqslant \min\{k - \Delta^{(\sigma_i,\sigma_{-i})}(t), \Delta^{(\sigma_i,\sigma_{-i})}(t) + k\} + t = \tau^{(LCR_i^{\sigma_i}, LCR_{-i}^{\sigma_{-i}})}$, which implies $P_t^{(\sigma_i,\sigma_{-i})} \leqslant P_t^{(LCR_i^{\sigma_i}, LCR_{-i}^{\sigma_{-i}})}$.

Further, $t < \tau^{(\sigma_i,\sigma_{-i})}$ implies $\tau^{(NSS_i^{\sigma_i}, NSS_{-i}^{\sigma_{-i}})} = \infty$, $\mathbb{P}\{\tau^{(NSS_i^{\sigma_i}, LCR_{-i}^{\sigma_{-i}})} > t + k|\mathcal{F}_t\} > 0$ and $\mathbb{P}\{\tau^{(NSS_i^{\sigma_i}, LCR_{-i}^{\sigma_{-i}})} < \infty|\mathcal{F}_t\} > 0$ so that $R < \infty$ yields $0 = P_t^{(NSS_i^{\sigma_i}, NSS_{-i}^{\sigma_i})} < P_t^{(NSS_i^{\sigma_i}, LCR_{-i}^{\sigma_i})} < P_t^{(LCR_i^{\sigma_i}, LCR_{-i}^{\sigma_i})}$ as desired. □

A.2 Proof of Proposition 2

Proof. $\forall i \in I, \sigma_i \in \mathcal{A}_i$: I let $d_i^{\sigma_i}$ denote a random variable that equals the first period that σ_i differs from LCR_i on the path of play when all other players play LCR. $\forall i \in I : \forall \sigma_i \in \mathcal{A}_i$: $V_i^{(LCR_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} > k} = V_i^{(\sigma_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} > k}$ so that $\mathbb{E}[V_i^{(LCR_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k}] \geqslant \mathbb{E}[V_i^{(\sigma_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k}]$ suffices to prove the desired conclusion. $\forall i \in I, \sigma_i \in \mathcal{A}_i$: I define $\Lambda_i^{\sigma_i} \equiv \{\omega \in \Omega : \{Y_{1,1,i}^{(\sigma_i,LCR_{-i})} Y_{2,1,i}^{(\sigma_i,LCR_{-i})} A_{2,1,i}^{(\sigma_i,LC}$
$A_{1,1,i}^{(\sigma_i,LCR_{-i})}) = 1\} \bigcap \bigtimes_{t=2}^{k} \{Y_{1,t,i}^{(\sigma_i,LCR_{-i})} A_{1,t,i}^{(\sigma_i,LCR_{-i})} = 0\} \bigcap \bigtimes_{t=2}^{k} \{Y_{2,t,i}^{(\sigma_i,LCR_{-i})} (1 - A_{2,t,i}^{(\sigma_i,LCR_{-i})}) = 0\}\}$ and $\Lambda_i \equiv \{\omega \in \Omega : Y_{1,1,i}^{(LCR_i,LCR_{-i})} Y_{2,1,i}^{(LCR_i,LCR_{-i})} = 1\}$.

Then,

$\mathbb{E}[V_i^{(\sigma_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k}]$

$\leqslant \mathbb{E}[V_i^{(\sigma_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k \bigcap \Lambda_i^{\sigma_i}}] + (\frac{R}{1-\delta} + \delta^{k+1} \pi_i S) \mathbb{P}\{\{d_i^{\sigma_i} \leqslant k\} \bigcap (\Lambda_i^{\sigma_i})^c\}$

$\leqslant \delta^k \mathbb{E}[\sum_{t=1}^{k} R_t Y_{2,t,i}^{(\sigma_i,LCR_{-i})} A_{2,t,i}^{(\sigma_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k \bigcap \Lambda_i^{\sigma_i}}] + \delta^k \pi_i S \mathbb{P}\{d_i^{\sigma_i} \leqslant k\}$.

The second inequality follows from $\min_{i \in I} \pi_i \times S \geqslant \frac{R}{\delta^k (1-\delta)^2}$.

Additionally,

$\mathbb{E}[V_i^{(LCR_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k}]$

$\geqslant \mathbb{E}[V_i^{(LCR_i,LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k \ \bigcap \ \Lambda_i}] + \delta^k \pi_i S \mathbb{P}\{\{d_i^{\sigma_i} \leqslant k\} \bigcap \Lambda_i^c\}$

$\geqslant \delta^k \mathbb{E}[\sum_{t=1}^{k} R_t Y_{1,t,i}^{(LCR_i, LCR_{-i})} A_{1,t,i}^{(LCR_i, LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k \bigcap \Lambda_i}] + \delta^k \pi_i S \mathbb{P}\{d_i^{\sigma_i} \leqslant k\}$.

If $\Lambda_i^{\sigma_i} = \varnothing$, then the result follows immediately. Otherwise, let $\sigma_i' \in \mathcal{A}_i$ denote a strategy such that player i follows σ_i at $t = 1$ and then LCR thereafter. Then,
$\mathbb{E}[\sum_{t=1}^{k} R_t Y_{2,t,i}^{(\sigma_i, LCR_{-i})} A_{2,t,i}^{(\sigma_i, LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k \bigcap \Lambda_i^{\sigma_i}}] = \mathbb{E}[\sum_{t=1}^{k} R_t Y_{2,t,i}^{(\sigma_i, LCR_{-i})} A_{2,t,i}^{(\sigma_i, LCR_{-i})} \mathcal{I}_{\Lambda_i^{\sigma_i}}]$
$\leqslant \mathbb{E}[\sum_{t=1}^{k} R_t Y_{2,t,i}^{(\sigma_i', LCR_{-i})} A_{2,t,i}^{(\sigma_i', LCR_{-i})} \mathcal{I}_{\Lambda_i^{\sigma_i'}}] = \mathbb{E}[\sum_{t=1}^{k} R_t Y_{1,t,i}^{(LCR_i, LCR_{-i})} A_{1,t,i}^{(LCR_i, LCR_{-i})} \mathcal{I}_{\Lambda_i}]$
$= \mathbb{E}[\sum_{t=1}^{k} R_t Y_{1,t,i}^{(LCR_i, LCR_{-i})} A_{1,t,i}^{(LCR_i, LCR_{-i})} \mathcal{I}_{d_i^{\sigma_i} \leqslant k \bigcap \Lambda_i}]$, which completes the proof. □

A.3 Proof of Proposition 4

Proof. Proposition 2 implies that LCR constitutes a best response for any subgame in which both branches possess the same length when all other players play LCR. Then, via symmetry, demonstrating that LCR constitutes a best response to all other players playing LCR when the branches possess different lengths suffices to establish the result.

For exposition, let $\Delta > 0$ denote the absolute gap between branch 1's length and branch 2's length at the beginning of the subgame. If a player deviates from LCR on the equilibrium path only after $k - \Delta$ periods, then LCR trivially produces a higher payoff. Otherwise, the associated strategy's payoff is bounded above by $\frac{R}{1-\delta} + \delta^{k-\Delta+1} \pi_i S$ whereas LCR produces a payoff at least as large as $\delta^{k-\Delta} \pi_i S$. Then, $\forall \pi_i, \Delta : \frac{R}{1-\delta} + \delta^{k-\Delta+1} \pi_i S \leqslant \delta^{k-\Delta} \pi_i S \Leftrightarrow \pi_i S \geqslant \frac{R}{\delta^{k-\Delta}(1-\delta)^2}$ constitutes a sufficient condition to establish subgame perfection. Then, $\pi_i S \geqslant \min_i \pi_i S \geqslant \frac{R}{\delta^k (1-\delta)^2} \geqslant \frac{R}{\delta^{k-\Delta}(1-\delta)^2}$ completes the proof. □

A.4 Proof of Proposition 5

Proof. $\forall i \in I, t \in \mathbb{N}$, I define strategy $\sigma_{i,t} \in \mathcal{A}_i$ such that player i follows σ_i^* until period t and LCR thereafter. Then, $\forall i \in I, t \in \mathbb{N} : \mathbb{E}[V_i^{(\sigma_i^*, \sigma_{-i}^*)}] \geqslant \mathbb{E}[V_i^{(\sigma_{i,t}, \sigma_{-i}^*)}]$. $\forall i \in I, t \in \mathbb{N}$: $V_i^{(\sigma_i^*, \sigma_{-i}^*)} \mathcal{I}_{\tau^{\sigma^*} \leqslant t} = V_i^{(\sigma_{i,t}, \sigma_{-i}^*)} \mathcal{I}_{\tau^{\sigma^*} \leqslant t}$ so that $\mathbb{E}[V_i^{(\sigma_i^*, \sigma_{-i}^*)} \mathcal{I}_{\tau^{\sigma^*} > t}] \geqslant \mathbb{E}[V_i^{(\sigma_{i,t}, \sigma_{-i}^*)} \mathcal{I}_{\tau^{\sigma^*} > t}]$. Direct verification reveals that $\mathbb{E}[V_i^{(\sigma_i^*,\sigma_{-i}^*)}\mathcal{I}_{\tau^{\sigma^*}>t}] \leqslant \delta^t \times \mathbb{P}\{t < \tau^{\sigma^*} < \infty\} \times (\sum_{s=1}^{t} R_s + \frac{R\delta}{1-\delta} + \pi_i S)$ and $\mathbb{E}[V_i^{(\sigma_{i,t},\sigma_{-i}^*)}\mathcal{I}_{\tau^{\sigma^*}>t}] \geqslant \delta^{t+k} \times \mathbb{P}\{\tau^{\sigma^*} > t\}\tilde{\pi}_i^{2k} \times \pi_i S$ with $\tilde{\pi}_i \equiv \frac{\pi_i S}{S+\sum_{t=1}^{\infty} R_t}$. Thus, $\forall i \in I, t \in \mathbb{N}$: $\mathbb{P}\{\tau^{\sigma^*} > t\} \leqslant \frac{\mathbb{P}\{t<\tau^{\sigma^*}<\infty\}\times(\sum_{s=1}^{t} R_s+\frac{R\delta}{1-\delta}+\pi_i S)}{\delta^k\tilde{\pi}_i^{2k}\times\pi_i S}$ so that taking limits as $t \to \infty$ on both sides yields $\mathbb{P}\{\tau^{\sigma^*} = \infty\} = 0$ as desired. □

A.5 Proof of Proposition 6

Proof. Note that $\rho - \rho' \leqslant \rho \leqslant \frac{B}{M} < \frac{1}{2}$ so that $1 - \rho - (\rho - \rho') > 0$, which implies $(1-\rho)(1-(\rho-\rho')) > \rho(\rho-\rho')$.

Then,

$$\lim_{z\to\infty} \mathbb{P}(T < \infty)$$
$$= \lim_{z\to\infty} (\tfrac{\rho(\rho-\rho')}{(1-\rho)(1-(\rho'-\rho))})^z$$
$$= 0$$ □

A.6 Proof of Proposition 7

Proof. Let $z > \frac{1}{r}$. Then,

$$\max_{\rho,\ \rho'\leqslant\rho} \rho' M \times \mathbb{P}(T < \infty) - r \times \mathbb{E}[T] \times \rho M - \rho(M - M')$$
$$\leqslant \max_{\rho,\ \rho'\leqslant\rho} \rho M - r \times \mathbb{E}[T] \times \rho M$$
$$\leqslant \max_{\rho} \rho M - r \times z \times \rho M$$
$$= \max_{\rho} \rho M \times (1 - r \times z)$$
$$\leqslant 0$$

The inequality is strict if $\rho > 0$ so that any attempt at a double-spending attack is unprofitable. $\rho = 0$ corresponds to not attempting a double-spending attack. □

Appendix B. Long-Range Attack

Another infamous attack, similar to double spending, is known as the long-range attack. This attack assumes that the attacker held some coins long ago and that she wishes to regain possession of those coins. To regain possession of her coins, the attacker must fork the blockchain from before the block in which she gives up possession of her coins and then make the new branch the longest branch. The attack is similar to a double-spending attack in the sense that this attack also requires the attacker to create a new branch of the blockchain and the attack succeeds only if the new branch can be made the longest branch. Subsequently, I demonstrate that the long-range attack is not profitable for a PoS blockchain for which the native coin possesses a sufficiently large market capital. Moreover, even with a low market capital and zero interest rates, an attacker must hold at least 50% of native coins for the long-range attack to become profitable.

$$\begin{aligned}
&\max_{\rho}\ \rho_0 M' \times \mathbb{P}(T < \infty) - r \times \mathbb{E}[T] \times \rho M - \rho M \\
&s.t. \\
&\qquad 0 \leqslant \rho \leqslant 1 \\
&\qquad \rho M \leqslant B
\end{aligned} \tag{B7}$$

Problem B7 provides the problem of the attacker. The problem resembles that of the attacker in a double-spending attack except that the attacker holds an exogenous proportion of coins, $\rho_0 \in (0, 1)$, from $\zeta > 0$ blocks ago with ζ being some large integer. The attacker forks the blockchain from ζ blocks ago when she held ρ_0 proportion of coins and seeks to make this new branch longer than the main branch. For the attack to succeed, the attacker must not only extend the new branch but also slow the main branch; otherwise, her new branch cannot catch up to the main branch. The attacker can slow the main branch only if she purchases some native coins on that branch. Accordingly, the attacker's problem is to select a proportion of coins, $\rho \in [0, 1]$, to purchase from the main branch such that her expected profits are maximized. Those coins cost her ρM with $M > 0$ denoting the market capital of the native coin before the attack. The attacker finances this purchase by borrowing at interest rate $r > 0$ per unit time, which implies that she bears expected interest costs $r \times \mathbb{E}[T] \times \rho M$ over the duration of the attack with T denoting the random attack duration. The attacker's borrowing and thus maximal coin holding is restricted by a borrowing limit, $B > 0$.

As discussed, the attack consists of the attacker creating a new branch by forking the blockchain from ζ blocks in the past. The attacker then seeks to extend that new branch to become as long as the main branch. If the attacker succeeds in her task (i.e., if $T < \infty$), the new branch becomes accepted by all validators and thus the attacker effectively regains possession of the ρ_0 proportion of coins she held ζ blocks ago. If the attack succeeds, the attacker sells her full coin holdings for $\rho_0 M'$ and pays back the principal of ρM from her borrowing with M' denoting the market value of native coins after initiating the attack. Following the analysis from previous sections, $M' \leqslant M$ because the attack involves creating disagreement on the blockchain although the results of this section hold even when $M' = M$.

Proposition 8. *Unprofitable Long-Range Attack*
If $M > \frac{B}{1-\rho_0}$, *then the long-range attack is not profitable.*

Proposition 8 demonstrates that PoS blockchains with large native coin market capitals are protected against long-range attacks in the sense that the long-range attack is not profitable for the attacker. Since the attacker must slow the main branch to successfully execute a long-range attack, the attacker must acquire a sufficiently large coin share on the main branch for the attack to succeed. The market value of the necessary coin share increases with the market capital of the native coin. For a sufficiently large market capital, the purchase of the necessary coin share to successfully execute the attack with probability one exceeds the borrowing limit and thus the attacker cannot always execute a successful attack. As the attack cannot always succeed, the expected interest rate costs from launching such an attack become unbounded and the long-range attack becomes unprofitable.

To clarify the aforementioned result, I contrast two cases. In the first case, I assume that the PoS native coin possesses a relatively small market capital of \$1 million. In the second case, I assume that the PoS native coin possess a relatively large market capital of \$100 billion. In both cases, I assume that the attacker held 1% of native coins ζ blocks in the past (i.e., $\rho_0 = .01$) and that the borrowing limit equals \$1 million. Direct verification reveals that the attacker must hold strictly more than 99% of the coins on the main branch to keep expected interest rate costs finite (i.e., to keep $\mathbb{E}[T] < \infty$). In the case of the relatively small market capital of \$1 million, such a purchase is feasible because the \$1 million borrowing constraint enables the attacker to purchase the necessary 99% of the native coins, which have a total value less than \$1 million. Nonetheless, in the case of the relatively large market capital for the native coins, the attacker must expend $.99\times$ \$100 billion = \$99 billion to purchase 99% of the native coins and such a cost far exceeds the borrowing limit thereby ensuring $\mathbb{E}[T] = \infty$ and thus unbounded expected interest rate costs. Those unbounded expected interest rate costs in turn render the attack unprofitable.

Proposition 9. *Unprofitable Long-Range Attack II*
If $\rho_0 < \frac{1}{2}$, then the long-range attack is not profitable regardless of the PoS native coin market capital, the borrowing constraint and the interest rate.

Proposition 9 establishes that even a PoS blockchain with an arbitrarily small native coin market capital is protected against any attacker who did not hold a majority of native coins. This result arises because the native coin holding on the main branch necessary for the attack to succeed with probability one is sufficiently large that the attacker would no longer benefit from the attack succeeding. For concreteness, I reconsider the previous example of the PoS blockchain with a relatively small native coin market capital of \$1 million. Recall that, in that example, the attacker possesses 1% of coins ζ blocks ago. Moreover, for the attacker to not incur unbounded expected interest rate costs (i.e., to keep $\mathbb{E}[T] < \infty$), the attacker must acquire at least 99% of the coins on the main branch. However, if the attacker holds 99% of the coins on the main branch and only 1% of the coins on the new branch, then the attacker would not want the attack to succeed as such a course of action would render coins on the main branch worthless and thus render her 99% coin share on the main branch worthless. Proposition 9 generalizes this example, highlighting that an attacker must necessarily hold a majority of coins on the new branch for the attack to be profitable regardless of the PoS native coin market capital, the borrowing constraint and the interest rate.

B.1 Proof of Proposition 8

Proof. The borrowing constraint implies $\rho \leqslant \frac{B}{M} < 1 - \rho_0$.

Then, for all feasible values of $\rho \neq 0$, we have that $\mathbb{P}(T < \infty) < 1$ and $\mathbb{E}[T] = \infty$. In turn, for all feasible values of $\rho \neq 0$, we have that $\rho_0 M' - r \times \mathbb{E}[T] \times \rho M - \rho M = -\infty$, which completes the proof.

Thus, any attempt of a long-range attack generates infinite negative utility thereby completing the proof. I exclude $\rho = 0$ from the aforementioned analysis because that choice corresponds to the case that no attack is attempted. □

B.2 Proof of Proposition 9

Proof. If $0 < \rho \leqslant 1 - \rho_0$, then $\mathbb{E}[T] = \infty$ so that $\rho_0 M' \times \mathbb{P}(T < \infty) - r \times \mathbb{E}[T] \times \rho M - \rho M = -\infty$.

In contrast, if $\rho > 1 - \rho_0$, then $\rho > \frac{1}{2} > \rho_0$. Then,

$\rho_0 M' \times \mathbb{P}(T < \infty) - r \times \mathbb{E}[T] \times \rho M - \rho M$

$\leqslant (\rho_0 - \rho) M$

< 0

Thus, any attempt of a long-range attack is strictly unprofitable. I exclude $\rho = 0$ from the analysis because that choice corresponds to the case that no attack is attempted. □