Pricing Security in Proof-of-Work Systems

George Bissias★, Rainer Böhme†, David Thibodeau★, Brian Levine★
★ University of Massachusetts Amherst, USA
† Universität Innsbruck, Austria

A key component of security in decentralized blockchains is proof of opportunity cost among block producers. In the case of proof-of-work (PoW), currently used by the most prominent systems, the cost is due to spent computation. In this paper, we characterize the security investment of miners in terms of its cost in fiat money. This enables comparison of security allocations across PoW blockchains that generally use different PoW algorithms and reward miners in different cryptocurrency units. We prove that there exists a unique allocation equilibrium, depending on market prices only, that is achieved by both strategic miners (who contemplate the actions of others) and by miners seeking only short-term profit. In fact, the latter will unknowingly compensate for any attempt to deliberately shift security allocation away from equilibrium.

Our conclusions are supported analytically through the development of a Markov decision process, game theoretical analysis, and derivation of no arbitrage conditions. We corroborate those results with empirical evidence from four years of blockchain and price data. Overall agreement is strong. We find that from the beginning of 2018 until the end of 2021, market prices predicted security allocation between Bitcoin and Bitcoin Cash with error less than 1%; the error was consistently less than 0.25% after 2018. These results are further corroborated by our establishment of Granger-causality between change in market prices and change in security allocation. From the beginning of October, 2019, until mid April, 2021, market prices also predicted security allocation between Bitcoin and Litecoin with error no greater than 0.55%. We show that the equilibrium subsequently breaks down due to the influence of merged mining in Litecoin.

To demonstrate the practicality of our results, we describe a trustless oracle that leverages the equilibrium to estimate the price ratios of PoW cryptocurrencies from on-chain information only.

Additional Key Words and Phrases: blockchain security, proof-of-work, no arbitrage

1 INTRODUCTION

Cryptocurrencies such as Bitcoin [1] have emerged as an intriguing complement to state-backed fiat currencies. We analyze the security of blockchains in the form of distributed systems with decentralized control and weak identification of participants (i.e., Sybil attacks [2] are possible and require mitigation). Typically, cryptocurrencies are implemented using a blockchain data structure, with each block containing a set of transactions. Although there are many aspects of blockchain security, in this paper we focus on security as it relates to consensus on the contents of blocks.

A blockchain is secure only to the extent that consensus emerges from the entire set of participants rather than an individual or subgroup. For blockchains with open membership, consensus is based on one of several different mechanisms including *proof-of-work (PoW)* [1] and *proof-of-stake (PoS)* [3], which are the two most popular choices. To gain the authority to record transaction history to the blockchain, the former requires participants to demonstrate use of computational resources, while the latter requires that participant funds be locked for a fixed period of time. Because participants could invest their resources elsewhere, participation in consensus, and thus the basis of blockchain security, is the summed opportunity cost of all participants. This cost is offset by a reward paid in cryptocurrency, which has a market-driven fiat value. Blockchains are secure when the opportunity cost cannot be borne by one participant or group that seeks to control consensus. Thus, the relative security of cryptocurrencies can be determined from the fiat value of their opportunity costs.

PoW blockchains, particularly Bitcoin, have become a major source of controversy due to their negative environmental impact [4, 5] as well as association with criminal activity [6, 7]. In this paper, we do not attempt to diminish any of these concerns. Nevertheless, it is our view that PoW blockchains continue to gain in economic importance as evidenced by their increasing market capitalization. Therefore, understanding their security, particularly insofar as it relates to market price fluctuations, is of paramount importance.

The relationship between the resources that participants choose to allocate among chains and the market-based fiat exchange value of each cryptocurrency is fundamental to the amount of security provided, and yet it is not very well understood. In this paper, we provide novel analysis of this relationship. Earlier work offers an incomplete understanding of this relationship. Spiegelman et al. [8] predicted the existence of stable equilibria among resource allocations, and Kwon et al. [9] subsequently identified multiple Nash equilibria, with one being closely observed in practice. Indeed, there exists evidence that some blockchain participants are already aware of this equilibrium [10].

Although Kwon et al. [9] take an important first step, we feel their model is too limited and their conclusions are informal and partly misleading. Their utility function is not parsimonious, relying on multiple miner strategies unnecessarily so that the regime for each equilibrium cannot be determined without unobservable information. This precluded critical tests of the theory in their work and even led to what we show is an inaccurate conclusion regarding the viability of minority hash rate blockchains. The following questions remain unanswered in their work. Why does one equilibrium dominate all others in practice? How do nonstrategic agents *find* this equilibrium. And how does the equilibrium change with protocol details such as cryptocurrency issuance rate or choice of PoW algorithm? These questions are critical to understanding PoW blockchain security, and to the best of our knowledge, the present work is the only one that provides formal answers.

A secondary goal of our work is to bridge the gap between techniques familiar to computer scientists and those more commonly applied in the field of economics and finance. We believe that both communities can benefit from this synthesis. Indeed, the present venue has explored the economic aspects of PoW several times in the past [11–13]. In this paper, we demonstrate how fundamental concepts from economics and finance can be used to develop a robust and highly accurate theory of security in PoW blockchains. For example, we show that between January 1, 2018 and January 1, 2022, cryptocurrency prices alone are sufficient to predict resource allocation between Bitcoin and Bitcoin Cash with root mean squared error at most 0.91%; the maximum error after 2018 was no more than 0.22%. Moreover, during overlapping periods in 2017, the error in our model is roughly three times lower than prior work. And beginning October 2019, until April 15, 2021, we show that a combination of cryptocurrency and hash price data is sufficient to predict resource allocation between Bitcoin and Litecoin (which do not share a PoW algorithm) with error of 0.55%.

The implications of our findings are profound for the blockchain ecosystem. They provide insight into the motivations and reasoning of PoW *miners*, the typically reticent participants responsible for securing many of the most prominent blockchains, including Bitcoin. For at least the past four years, the scope of our data, change in currency price alone has proven to be a remarkably accurate predictor of change in miner resource allocation. Our results suggest that this connection is typically Granger-causal [14]: changes in the fiat value of a cryptocurrency will tend to result in a rapid change to investment in its security.

In sum, we make the following contributions.

(1) We use a multi-method approach that spans solution concepts established in computer science, economics, and finance. Specifically, we use a Markov decision process (MDP) to analyze basic resource allocation dynamics, competitive game theory for multi-miner interaction, and consolidate our theory in no-arbitrage conditions [15], a powerful solution concept in finance that is more general than MDPs and requires fewer assumptions than Nash

equilibria within game theory. This analysis yields a single equilibrium allocation that we show to be an attractor; every other allocation will tend to rebalance toward it.

(2) We evaluate the strength of this attractor on four years of historical blockchain and price data for many of the most popular PoW blockchains including Bitcoin, Ethereum, Bitcoin Cash, Ethereum Classic, and Litecoin. We show that actual resource allocation among blockchains that share the same PoW algorithm follows extremely close to the equilibrium; those that do not share a PoW algorithm also follow closely, but less so due to market inefficiencies. To the best of our knowledge, the latter analysis is the first quantitative comparison of security between blockchains using different PoW algorithms.

(3) Using Granger-causality, we show that, on a systematic, hourly basis, change in the fiat value of a cryptocurrency typically elicits a change in the resources a miner allocates to securing its blockchain. We also show that the opposite link is generally rare, but has manifested with weak significance occasionally.

(4) We leverage the correlation between actual and predicted resource allocations to describe how to develop a trustless exchange price ratio oracle between pairs of PoW cryptocurrencies sharing the same PoW algorithm. Its susceptibility to manipulation is limited relative to other decentralized solutions. And, to the best of our knowledge, it is the first to use only on-chain information in a way that allows for quantification of manipulation cost.

We conclude with a comparison to related work.

2 SECURITY IN POW BLOCKCHAINS

A distributed and decentralized *blockchain* [1] (or *chain* for brevity) is a data structure, formed among autonomous peers having weak identities [2], who assemble *blocks* in a hash-linked list. Each block contains a set of *transactions*, which can be simple account updates or more complex state changes in smart contracts. Transactions are *confirmed* once they appear in a block on the chain.

Mining. PoW *miners* achieve consensus through a block *mining* process. The purpose of the mining process is to compensate for the absence of strong identities by requiring each peer participating in the blockchain to provide evidence of computation. In its simplest form, each miner applies a cryptographic hash algorithm [16] to the metadata associated with the block called the *block header*, randomly varying a nonce in that header. If the resulting hash value is less than a known *target*, then the miner is considered to have mined the block and it is awarded a portion of cryptocurrency (or *coin* for brevity): some is newly minted to form a base reward and the rest is derived from transaction fees. Coins carry an exchange value in *fiat currency* (a state-backed currency such as USD), which is established by exchanges that facilitate trade. *Difficulty* is a quantity inversely related to the target by a constant. It is essentially the expected number of hashes required to mine a block, and we treat it as such unless otherwise indicated. (Most blockchains actually define the difficulty somewhat differently, but our definition is similar in spirit.) The difficulty (and therefore the target) is updated via a protocol-defined algorithm called a *difficulty adjustment algorithm* (DAA) so that all miners, working independently, are expected to mine a block in a fixed time (e.g. 600 seconds in Bitcoin).

Threat model. Blockchain security is multifaceted [17]; vulnerabilities can arise at the network [18, 19], protocol [20], consensus [1, 21], or application [22] layers. But perhaps the most fundamental attack on PoW blockchains is the *51% attack*, which arises when the computational resources of a nefarious individual or organization exceed those of the remaining participants. In this work, as is common in related works [9, 21], we assume that attackers cannot break primitives or exploit network or cryptographic vulnerabilities [23] and that they have potentially substantial but ultimately limited resources. Because attacker hash rate is assumed to be limited, risk of a 51% attack is lowest on blockchains where absolute hash power is highest [24, 25].

DEFINITION 1: The *security metric* for PoW blockchains is hash rate.

Hash markets. PoW mining constitutes a bona fide cost to miners in terms of both capital outlay and expended electricity [26]. The majority of work performed on all major PoW blockchains uses application specific integrated circuits (ASICs). Purchasing ASICs constitutes a significant capital expenditure and also creates lock-in because these devices can typically only be used to execute a single PoW algorithm. Yet some blockchains, such as Bitcoin and Bitcoin Cash, use the same PoW algorithm. In this case, the cost to move mining resources to the other chain is negligible. This creates an economic tension between such blockchains whereby the incentive to mine on a given chain vacillates depending on the relative fiat value of reward per hash at any given moment. Moreover, there exist markets [27] for renting time on ASICs, which allow miners to effectively purchase reward on blockchains implementing a PoW algorithm that they cannot mine directly themselves, or sell excess capacity and thus amortize capital they have invested in ASICs.

3 A MOTIVATING EXAMPLE

In this section, we illustrate by example how any miner, given the choice between two blockchains, will allocate hash rate to each in proportion to its share of the total reward so as to optimize profit. This principle is carried forward throughout the paper.

Imagine a simplified blockchain ecosystem where there exist only two chains A and B, each implementing the same PoW algorithm and each aiming to produce blocks at the same average rate of $T = 2$ seconds. The coins issued by A carry 2 units of fiat value while those issued by B carry only 1 unit. There exists a single miner who must decide how to allocate his available hash rate of $H = 6$ hashes per second among the two chains so as to maximize profit. We assume that each chain's DAA fully adjusts to the hash rate applied to that chain after a single block.

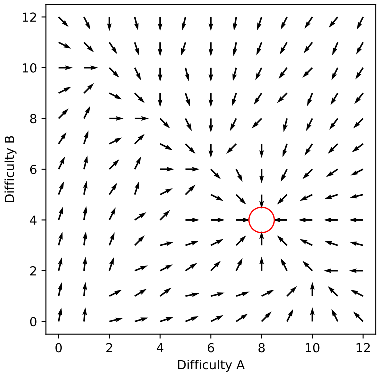


Fig. 1. *Direction of optimal allocation rebalancing of a total of 6 hashes per second among chains A and B for given difficulties (expected hashes per block). Each point in the plot represents difficulties for chains A and B. The direction of arrows indicates how to optimally rebalance the allocation at the given difficulties. The concentration point for this miner, the only state where remaining stationary is optimal, is indicated by the red circle.*

Suppose that initially the miner's hash rate is split evenly among the two chains. What is the miner's optimal hash rate *rebalancing* given the initial difficulty on each chain? To answer this question, we use a Markov decision process (MDP). States in the MDP correspond to the difficulty associated with each chain, which we measure in terms of the expected number of hashes required to mine a block. Actions correspond to the miner's hash rate *allocation* among the two chains. And transitions occur from one state to another when a block is mined using the hash rate given by the current action. Further MDP details can be found in Appendix B. Figure 1 shows the optimal policy for the miner in a grid where each column represents the current difficulty on chain A and each row represents the current difficulty on chain B. The direction of the arrow at each grid point indicates the direction of optimal hash rate rebalancing among the two chains.

The figure reveals that the optimal action for the miner is not simply to allocate all hash rate to chain A, which offers the highest reward per block. This is because shifting all hash rate to A forces the difficulty higher, raising cost. *This tension captures the essence of an equilibrium that forms between any two PoW blockchains.*

Upon closer inspection, it appears that *the optimal strategy is for the miner to allocate his hash rate on each chain in proportion to the chain's relative share of fiat reward.* The MDP confirms that this property holds for other reward proportions, so it does not appear to be a coincidence. This result implies that the fiat value of a blockchain's native coin has a direct and quantifiable impact on its security relative to another blockchain. In the remainder of this document we explain why this is the case and both generalize and formalize the result. Going forward, we model multiple competing miners who allocate resources among blockchains generally having different block times and PoW algorithms. In this broader context, we observe the formation of an allocation *equilibrium* that forms at precisely the same point achieved by the lone miner in this example.

4 FRAMEWORK

In this section, we construct an analytical framework, which generalizes familiar blockchain concepts and enables presentation of novel concepts in subsequent sections.

We consider two blockchains A and B, each generally using different PoW algorithms ALG_A and ALG_B. Having different PoW algorithms, we imagine that the sets of miners M_A and M_B of each coin are generally disjoint, but in the special case where $\mathsf{ALG}_A = \mathsf{ALG}_B$ or when the algorithms are supported by the same mining hardware, their intersection can be non-empty. We denote the set of all miners by $M = M_A \cup M_B$.

We denote the *hash rate* (hashes per second) for miner m by $H(m)$, and with H_A and H_B we denote the *aggregate* hash rate of all miners on chains A and B, respectively. We assume that the hash rate for each miner remains constant over time as does the total hash rate H. Through secondary *hash rate markets* such as NiceHash [27], it is possible for a miner $m \in M_A$ to trade hash power in A (through a series of trades) for hash power in B. Thus, the sets M_A and M_B are fluid, i.e. miners can readily move between sets. By T_X we denote the *target average block inter-arrival time* for chain $X \in \{A, B\}$. Let τ represent time and define $t_X(\tau)$ as the *actual* inter-arrival time for the last block before time τ from chain $X \in \{A, B\}$.

DEFINITION 2: The *spot hash price* at time τ, denoted $\sigma_X(\tau)$, is the fiat price of a single hash using PoW algorithm ALG_X.

The spot hash price on a given blockchain is simply the cost to purchase hash power on that chain. Using Definition 2, and assuming a perfectly efficient hash rate market, we can quantify the fiat value of hash power devoted to securing a given blockchain.

DEFINITION 3: The *actual security investment* in blockchain X, denoted $s_X(\tau) = H_X(\tau)\sigma_X(\tau)$, is the actual fiat value of hash power devoted by miners to mining on chain X for 1 second at time τ.

The actual security investment definition abstracts the conventional concept of hash rate by converting hashes per second to fiat per second.

DEFINITION 4: The *security allocation* at time τ, denoted by vector $\boldsymbol{w}(\tau)$, is the fraction of the total actual security investment applied to each chain:

$$\boldsymbol{w}(\tau) = (w_A(\tau), w_B(\tau)) = \frac{1}{s_A(\tau) + s_B(\tau)}(s_A(\tau), s_B(\tau)). \tag{1}$$

We often refer to a security allocation as simply an allocation for brevity, and we also drop τ from the notation when time is either unimportant or clear from context. Notice that the security allocation to chain $X \in \{A, B\}$ is equivalent to the fraction of total actual security investment allocated to chain X. Thus, when chains A and B share the same PoW algorithm, $\sigma_A = \sigma_B$ and $\boldsymbol{w}$ gives the share of total hash rate allocated to each chain. At times we consider the relative security attributable to miner m_i, which we denote by $\boldsymbol{w_i} = (w_{iA}, w_{iB})$, where

$$\sum_{i, m_i \in M} \boldsymbol{w_i} = \boldsymbol{w}. \tag{2}$$

The fiat value of the coinbase reward plus average fees for chain X is given by $V_X(\tau)$. Coinbase value decomposes into $V_X(\tau) = k_X(\tau)P_X(\tau)$, where $k_X(\tau)$ is the quantity of X coins (from base reward and average transaction fees) paid out per block, and $P_X(\tau)$ is the fiat value of each coin from chain X at time τ. Finally, define the relative reward of the two chains by $R(\tau) = \frac{V_A(\tau)}{V_A(\tau)+V_B(\tau)}$.

DEFINITION 5: The *target security investment* $S_X(\tau)$ for a blockchain X at time τ is the fiat value of hash power that must be applied to chain X by miners, for each second beginning at time τ, to produce a block in expected time T_X.

Recall from Section 2 that the difficulty of a blockchain measures the expected number of hashes required to mine a block. The target security investment abstracts the difficulty by converting hashes per block to fiat per second. Contrast actual and target security investments s and S with conventional quantities hash rate and difficulty, H and D. Quantities s and H are controlled by the miner, they reflect actual resources devoted to mining, while quantities S and D are set by the blockchain protocol, they reflect prescribed mining resources.

4.1 Inferring security

Meeting target security S_X is required to produce blocks on chain X in expected time T_X. Thus, the rate of coin issuance is tied directly to the relative difference between actual and target security. To maintain a desired block time, blockchains attempt to tune S_X to match actual security s_X as closely as possible. However, in PoW blockchains, s_X cannot be determined from on-chain information alone. So PoW blockchain protocols must implement methods for inferring security.

DEFINITION 6: For a given blockchain, a *security adjustment algorithm* (SAA) is any algorithm that adjusts its baseline security S_X so that it tends toward s_X.

The SAA is simply an abstraction of the DAA described in Section 2. To be clear, blockchains implement DAAs, but we choose to describe them as SAAs to emphasize that they are changing the target security investment. A conventional SAA measures average block time $\bar{t}(\tau)$ over a given window and adjusts $S_X(\tau)$ in the direction of $S_X(\tau)\frac{T_X}{\bar{t}(\tau)}$. When $\bar{t}(\tau) = T_X$ we say the SAA is *at rest.*

Blockchains record their security in terms of difficulty D. Therefore, empirical analysis requires that we express security allocation in terms of the difficulty. Section 2 describes the difficulty as the expected number of hashes required to mine a block. Thus, for chain $X \in \{A, B\}$, and when the DAA is at rest, we have that $H_X(\tau) \approx D_X(\tau)/T_X$. Finally, according to Definitions 3 and 4,

$$\boldsymbol{w}(\tau) \approx \frac{1}{\hat{s}_A(\tau) + \hat{s}_B(\tau)}(\hat{s}_A(\tau), \hat{s}_B(\tau)), \text{ where } \hat{s}_X(\tau) = \frac{\sigma_X(\tau)D_X(\tau)}{T_X}. \tag{3}$$

5 NASH EQUILIBRIUM FOR SECURITY ALLOCATION

A relatively simple game is capable of describing (and generalizing) the concentration point observed by the MDP in Section 3. We introduce the *Security Allocation Game* among two blockchains A and B (not necessarily sharing the same PoW algorithm), which is a one-shot game with N homogeneous miners (the homogeneity assumption applies only in this section). Following conventions in the game theory literature, we distinguish an arbitrary miner m_i from all the others, which are labeled $m_{\text{-}i}$. The miner strategy space comprises all mixed allocations among chains A and B.

Recall from Section 4 that the security allocation across chains A and B for miner m_i is given by $\boldsymbol{w}_i = (w_{iA}, w_{iB})$, and $\boldsymbol{w}_{\text{-}i}$ is similarly defined for $m_{\text{-}i}$. Thus, given miner m_i and the group of other miners $m_{\text{-}i}$, the overall allocation is fully specified by $[\boldsymbol{w}_i, \boldsymbol{w}_{\text{-}i}]$. We assume unit aggregate security investment, which is completely allocated among the two chains, i.e. $|\boldsymbol{w}_i| = \frac{1}{N}$ and $|\boldsymbol{w}_{\text{-}i}| = \frac{N-1}{N}$. Being homogeneous, miners have the property that $(N-1)s(m_i) = s(m_{\text{-}i})$, i.e. each makes the same contribution to total security investment.

Assuming that the SAA for each chain is at rest, the total available payoff per second is given by

$$\boldsymbol{\pi}_i = \left(\frac{V_A}{T_A}, \frac{V_B}{T_B}\right). \tag{4}$$

Now define

$$\boldsymbol{u}_i = \left(\frac{w_{iA}}{w_{iA} + w_{\text{-}iA}}, \frac{\frac{1}{N} - w_{iA}}{1 - w_{iA} - w_{\text{-}iA}}\right), \tag{5}$$

which is miner m_i's share of the reward on each chain. The payoff for m_i is equal to $\boldsymbol{\pi}_i^{\mathrm{T}}\boldsymbol{u}_i$. Payoff has one term per chain and reflects the fact that reward is distributed to miners (in expectation) proportionally to the security they allocate to each chain. We search for a pure-strategy Nash equilibrium that leverages the payoff function in Eq. 4 and the miner homogeneity assumption. The existence of a symmetric pure strategy equilibrium is not remarkable, but it is instructive to show that such an equilibrium matches the main equilibrium discovered by Kwon et al. [9].

THEOREM 1: *The following allocation is a symmetric equilibrium for the Security Allocation Game:*

$$[\boldsymbol{w}_i^*, \boldsymbol{w}_{-i}^*] = \left[\frac{1}{N}(c, 1-c), \frac{n}{N}(c, 1-c)\right],$$

where $n = N - 1$ *and* $c = \frac{T_B R}{T_B R - T_A R + T_A}$. *When* $T_A = T_B$ *the equilibrium simplifies to* $c = R$. *(Proof in Appendix E.)*

The equilibrium specified by Theorem 1 coincides with the concentration point identified in Figure 1. This result tells us that a relatively simple game theoretical model explains the behavior observed in the optimal solution to a specific hash rate allocation problem, but with greater generality. However, the game theoretical approach also carries significant limitations. First, it relies on the homogeneity of hash power among miners. Second, it assumes that all miners have the same utility (optimizing Eq. 4), which is unrealistic because miners face variable costs and they may accept losses to promote a chain of their liking. Third, our simple game assumes that miners have no outside options, which exist in the real world by abstaining, mining on a third chain, or selling excess mining capacity. Fourth, the game does not consider higher moments of the payoff distribution (beyond expected value): miner risk appetite might result in different adjustments to obtain their individual objective function. Fifth, the approach is not exhaustive. It is difficult to completely eliminate the possibility of other equilibria that might arise asymmetrically or in mixed strategies. Kwon et al. [9] also developed a game theoretical model, which identifies the same symmetric Nash equilibrium and a number of others. However, that model suffers from the same limitations listed above, and others as well, which we discuss in Section 10.

In the next section, we introduce an approach that captures the uniqueness of the equilibrium described in Theorem 1. We use the technique to show that this equilibrium is unique under much weaker assumptions, eliminating the list of limitations above.

6 ARBITRAGE CONDITIONS AT EQUILIBRIUM

The game theoretical equilibrium of Section 5 is important because it explains the behavior of a group of strategic miners who understand how mining profitability changes in a competitive environment. In the case where all miners achieve this level of sophistication, and subject to the assumptions of the section, the equilibrium of Theorem 1 will be achieved. However, it is unlikely that all miners currently are strategic and implausible that all assumptions are met. Given these limitations, we seek to understand how security allocation is affected when most of the assumptions in Section 5 are relaxed.

The finance literature has studied *no arbitrage* (NA) conditions, weak conditions that guarantee price equilibrium. Informally, arbitrage occurs when profit is made at zero cost. NA theory posits that agents will change their behavior to exploit arbitrage opportunities when they exist, and will maintain their behavior (forming an equilibrium) when they do not. Figure 2 depicts conventional two-point arbitrage in the context of currency exchange. In this example, an investor sees an opportunity to capitalize on the difference in trade price of Bitcoin on two different exchanges. The existence of arbitrage creates strong incentive for investors to exploit the opportunity until they reach a point of no arbitrage.

6.1 Derivative Markets

In the study of finance, we are often concerned with the payoff of a certain *portfolio* of *financial securities* at a future date. In the simplest model, agent m_i purchases *contingent claims* on securities A and B in quantities $\boldsymbol{c}_i(\tau_1) =$

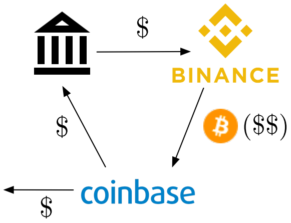


Fig. 2. Exploiting conventional (two-point) arbitrage in currency exchange. *An investor moves fiat currency from a bank to the Binance exchange where she trades the fiat for Bitcoin. She then transfers the Bitcoin to the Coinbase exchange where she is able to sell it for a greater amount of fiat currency than she began with.*

$(c_{iA}(\tau_1), c_{iB}(\tau_1))$ at time τ_1 using an initial *endowment* $e_i(\tau_1)$. The endowment constitutes the resources available to the agent for purchasing contingent claims. A contingent claim is any sort of derivative contract on the security whose payout depends on a future *outcome*, such as an option or futures contract. Contingent claims carry purchase prices $\boldsymbol{p}(\tau_1) = (p_A(\tau_1), p_B(\tau_1))$ at τ_1. Naturally, the *portfolio price*, $(\boldsymbol{c}_i(\tau_1))^{\mathrm{T}}\boldsymbol{p}(\tau_1)$, must not exceed endowment $e_i(\tau_1)$, which is the agent's budget constraint. Contingent claims can be sold at time τ_2 for payoff $\Pi(\tau_2)$, where $\Pi(\tau_2)$ is a matrix with columns corresponding to portfolio components, rows corresponding to individual states, and where each matrix entry corresponds to an outcome, claim pair. The agent seeks to maximize aggregate payoff, $\sum_{X\in\{A,B\}}(\Pi(\tau_2)c_i(\tau_1))_X$.

6.1.1 Notation. In the remainder of this section, we occasionally drop the time argument τ where it can be understood from context, but we reintroduce it in places where time should be emphasized. Also, for ease of exposition, we use *Hadamard notation* for component-wise multiplication and division of vectors u and v: $\boldsymbol{u}\odot\boldsymbol{v} = (u_A v_A, u_B v_B)$ and $\boldsymbol{u}\oslash\boldsymbol{v} = (u_A/v_A, u_B/v_B)$.

6.2 Blockchain Security Market

We define the *Blockchain Security Market* for agent m_i, a miner, as follows. Endowment $e_i(\tau_1)$ is equal to $s(m_i)$, or the fraction of all fiat currency devoted to security across chains A and B at time τ_1 that belongs to m_i. Accordingly, $e_i(\tau_1)$ is also a scalar multiple of $H(m_i)$, the number of hashes that m_i is capable of producing per second. We assume that $e_i(\tau_1)$ remains fixed over time so that the miner consistently operates with the same hash rate.

Price vector $\boldsymbol{p}(\tau_1) = (S_A(\tau_1), S_B(\tau_1))$ is equal to the cost of *purchasing* 1 second worth of expected reward for mining on chains A and B. Each portfolio, $\boldsymbol{c}_i(\tau_1)$, represents a contingent claim on future coinbase from chains A and B, respectively, between times τ_1 and τ_2, where we assume that $\tau_2 - \tau_1 = 1$ second and $e_i(\tau_1) = (\boldsymbol{c}_i(\tau_1))^T\boldsymbol{p}(\tau_1)$. Note that components of claim vector $\boldsymbol{c}_i(\tau_1)$ can exceed 1; i.e., it is possible to purchase more than a single claim each second (which would tend to generate blocks faster than the blockchain's target rate). We consider only one state at time τ_2, having payoff vector

$$\boldsymbol{\pi}(\tau_2) = \left(\frac{V_A(\tau_2)}{t_A(\tau_2)}, \frac{V_B(\tau_2)}{t_B(\tau_2)}\right),$$

which is the total expected fiat value for each chain's block reward during the 1 second time period and is equal to Eq. 4 when SAAs for chains A and B are at rest.

In this definition, there exists no contingency because there is only one possible state at time τ_2. As such, it is possible to guarantee payoff $\boldsymbol{c}_i(\tau_1)^{\mathrm{T}}\boldsymbol{\pi}(\tau_2)$ at τ_2. Finally, we redefine the *security allocation* for miner m_i at time τ_1, in the context of the blockchain market, by

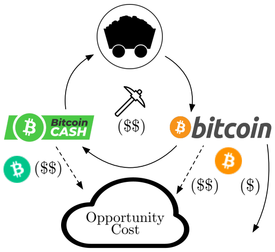


Fig. 3. Exploiting arbitrage between chains sharing the same PoW algorithm in the Blockchain Security Market. *Hashes, having opportunity cost* ($$)*, are generated at a constant rate by the miner. They can be traded off between the Bitcoin Cash and Bitcoin blockchains. In this example, shifting hashes to the Bitcoin blockchain will mine Bitcoin having value* ($) *in excess of the opportunity cost. This results in a Bitcoin payoff of value* ($).

$$\boldsymbol{w}_i(\tau_1) = \frac{1}{e_i(\tau_1)} \boldsymbol{c}_i(\tau_1) \odot \boldsymbol{p}(\tau_1). \tag{6}$$

This allocation corresponds to the fraction of the miner's total security investment devoted to each chain. Throughout, we assume that $|\boldsymbol{w}_i| = 1$, in other words, the miner allocates his resources entirely among the two chains.

6.2.1 Portfolio Rebalancing. We imagine that each miner holds initial claim $\boldsymbol{c}_i(\tau)$ but wishes to *rebalance* to a new claim $\boldsymbol{c}_i(\tau')$ at some future time τ' with the hope of achieving a higher payoff. Prices at time τ', $\boldsymbol{p}(\tau') = (S_A(\tau'), S_B(\tau'))$, correspond to the target security investment (fiat value per second) required to mine a block on each chain in the desired expected time. For the miner to rebalance his claims, he must *sell short* his claim on one chain to increase his claim on another. Thus, to enforce the notion of scarcity in security investment (and ultimately hash rate), we stipulate that $c_{iX}(\tau')p_X(\tau') \leq c_{iX}(\tau)p_X(\tau), X \in \{A, B\}$. This implies that, on any given chain, the miner cannot sell short a claim at price $\boldsymbol{p}(\tau')$ with total fiat value exceeding what he purchased at time τ.

6.2.2 Properties of security allocations. We are primarily interested in the overall effect of miner behavior. therefore, going forward, we consider only *aggregate* allocations.

DEFINITION 7: The *aggregate* claim $\boldsymbol{c}$ and endowment e across multiple miners are given by $\sum_i \boldsymbol{c}_i$ and $\sum_i e_i$, respectively. Aggregate security allocation is $\boldsymbol{w} = \boldsymbol{c} \odot \boldsymbol{p}/e$.

Note that Definition 7 provides a reinterpretation of security allocation w from Definition 4 in terms of portfolio price and claims. The following definitions are useful to us for reasoning about changes in allocation.

DEFINITION 8: The *distance* between two allocations $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ is given by the L1-norm of their difference: $|\boldsymbol{w}_1 - \boldsymbol{w}_2|$.

DEFINITION 9: An *allocation rebalancing* is an allocation $\Delta\boldsymbol{w}$ intended to update existing allocation $\boldsymbol{w}$ to $\boldsymbol{w}' = \boldsymbol{w} + \Delta\boldsymbol{w}$. We say that a rebalancing is *symmetric* when $\Delta\boldsymbol{w} = (\epsilon, -\epsilon)$ for some $\epsilon > 0$.

We are primarily interested in symmetric allocation rebalancings because they maintain constant aggregate resources across both chains.

6.2.3 Portfolio pricing. Typically, security prices emerge when buyers and sellers come to an agreement on an exchange price but the blockchain security market is unique in that prices are set algorithmically by the SAA. At time τ on

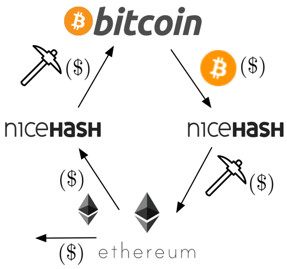


Fig. 4. Exploiting arbitrage between chains having different PoW algorithms in the Blockchain Security Market. *Proceeding clockwise from the upper left, SHA256 hash rate having fiat value* (\$) *is used to mine Bitcoin for zero marginal profit. The Bitcoin is then transferred back to NiceHash and used to purchase Dagger-Hashimoto hash rate at cost* (\$)*. This hash rate is applied to the Ethereum blockchain, which yields ether having fiat value* (\$\$)*, and which is* (\$) *greater than the fiat value of the original SHA256 hash rate.*

blockchain X, the SAA responds to a difference between the baseline security $S_X(\tau)$ and the inferred actual security $s_X(\tau)$ by moving the value of the former closer to the value of the latter. And because $p_X(\tau) = S_X(\tau)$, the action of the SAA has the effect of changing the portfolio price. The following proposition shows how to determine portfolio price when the SAA is at rest, i.e. $S_X(\tau) = s_X(\tau)$.

PROPOSITION 1: For any fixed allocation $\boldsymbol{w}$, after the SAAs on chains A and B come to rest, the portfolio pricing vector will be $\boldsymbol{p} = e\boldsymbol{w}$. (Proof in Appendix E.)

6.2.4 Arbitrage. An *arbitrage opportunity* is formally defined as the circumstance where $\boldsymbol{\pi}^{\mathrm{T}}\Delta\boldsymbol{w} \geq 0$ and $\Delta\boldsymbol{w}^{\mathrm{T}}\boldsymbol{p} \leq 0$, with at least one strict inequality [15]. Less formally, arbitrage is possible any time it is possible to guarantee future payoff at zero cost. We expect that a miner will seek to rebalance his claim to exploit the higher payoff in this circumstance. Figure 3 shows how a miner can rebalance his claim (i.e., his hash rate) between two blockchains sharing the same PoW algorithm to increase his profit. In this case, the opportunity cost of mining is considered a sunk cost, and arbitrage is captured by shifting hash rate to Bitcoin, which is the more profitable chain. In contrast, Figure 4 illustrates how arbitrage can be exploited among blockchains that generally employ different PoW algorithms by an agent owning no hash rate. The agent trades fiat for hash rate in a market such as NiceHash [27], and then distributes those hashes among blockchains. In this example, the fiat reward per unit of cost to secure Ethereum is greater than in Bitcoin. Therefore, it is possible to trade hash rate on Bitcoin for hash rate on Ethereum to boost profits.

We next prove that the equilibrium allocation defined in Theorem 1 is a point of no arbitrage. It is a point where there exists no financial incentive for miners to rebalance their portfolio of security allocations.

> **THEOREM 2:** *Assume any choice of SAA for chains A and B (not necessarily the same). When both SAAs are at rest and the relative reward R is stable, there exists no arbitrage at the following allocation*
>
> $$\boldsymbol{w}_{\mathrm{eq}} = \left(\frac{T_B R}{T_B R - T_A R + T_A}, \frac{T_A(1-R)}{T_B R - T_A R + T_A}\right), \tag{7}$$
>
> *which simplifies to*
>
> $$\boldsymbol{w}_{\mathrm{eq}} = (R, 1-R), \tag{8}$$
>
> *if $T_A = T_B$. (Proof in Appendix E.)*

Notice that the sole requirement for maintaining the equilibrium described by Theorem 2 is for each miner to actively update his allocation so as to maximize profit. **Consequently, the no arbitrage equilibrium is more plausible in practice than a Nash equilibrium because it can be achieved without a complex utility function and without directly contemplating the actions of other miners**.

6.2.5 Uniqueness. Now we establish the uniqueness of the equilibrium defined by Eq. 7 among all potential points of no arbitrage, which further motivates its formation in practice. We begin by proving a lemma that shows portfolio cost remains unchanged by any symmetric rebalancing.

> **LEMMA 1:** *For initial allocation $\boldsymbol{w}$ and price $\boldsymbol{p}$, with SAAs at rest, the claims associated with a symmetric rebalancing $\Delta\boldsymbol{w}$ are given by $\Delta\boldsymbol{c} = \Delta\boldsymbol{w} \oslash \boldsymbol{w}$ and it is always the case that $\Delta\boldsymbol{c}^T\boldsymbol{p} = 0$. (Proof in Appendix E.)*

In the following theorem, we establish that there exists opportunity for arbitrage at *any* allocation that is distinct from the equilibrium defined in Eq. 7. Moreover, we show that the arbitrage can be exploited with a symmetric rebalancing that moves the allocation closer to the equilibrium.

> **THEOREM 3:** *For any allocation $\boldsymbol{w} \neq \boldsymbol{w}_{\text{eq}}$, with SAAs at rest and relative reward R stable, there exists a symmetric allocation rebalancing $\Delta\boldsymbol{w}$, such that $|(\boldsymbol{w} + \Delta\boldsymbol{w}) - \boldsymbol{w}_{\text{eq}}| \leq |\boldsymbol{w} - \boldsymbol{w}_{\text{eq}}|$, which has price zero and strictly positive payoff. (Proof in Appendix E.)*

For allocations not at the equilibrium defined by Eq. 7, we next show that every symmetric rebalancing that moves the allocation away from the equilibrium can only reduce the miner's payoff. Along with Theorem 3, this result establishes that the equilibrium of Theorem 2 is an *attractor*, meaning that off-equilibrium allocations will tend to rebalanced toward it.

> **COROLLARY 1:** *For allocation $\boldsymbol{w} \neq \boldsymbol{w}_{\text{eq}}$, with SAAs at rest and relative reward R stable, any symmetric rebalancing allocation $\Delta\boldsymbol{w}$ such that $|(\boldsymbol{w} + \Delta\boldsymbol{w}) - \boldsymbol{w}_{\text{eq}}| > |\boldsymbol{w} - \boldsymbol{w}_{\text{eq}}|$ has price zero will result in strictly negative payoff. (Proof in Appendix E.)*

6.2.6 Miner behavior. What fraction of miners will act to exploit arbitrage and what are the consequences of this quantity? A specific answer to the former is beyond the scope of our work, so we devote the remainder of this section to answering the latter, subject to various assumptions about the quantity of miners seeking to exploit arbitrage. Toward that end, it is useful to contemplate the alternative behavior.

Define *rogue* mining behavior to be the act of mining on a given chain for reasons other than short term profitability. In general, rogue miners will not act to exploit arbitrage. Note, however, that rogue mining is not necessarily economically *irrational*; it could be motivated by factors exogenous to our model. For example, a miner might have significant coin holdings from chain *B*, in which case she could be incentivized to continue to mine on chain *B* despite it having lower profitability than chain *A*. This behavior might arise if all other miners migrate to chain *A*, leading to an existential crisis for chain *B*. The following corollary makes precise the intuition that, once equilibrium is achieved, SAAs can be expected to remain at rest while rogue miners do not change their allocation. It also helps to establish the plausibility of our assumption, made throughout this section, that SAAs remain at rest.

> **COROLLARY 2:** *Let relative reward ratio R be given. If the security allocation is equal to* $\boldsymbol{w}_{\text{eq}}$ *and SAAs come to rest, then the SAAs will remain at rest so long as R remains fixed and rogue miners do not change their allocation. (Proof in Appendix E.)*

Our final result in this section proves that $\boldsymbol{w}_{\text{eq}}$ constitutes a stable equilibrium when there exists a certain balance between rogue and arbitrage seeking miners.

> **THEOREM 4:** *If* $w_X > 0$ *for* $X \in \{A, B\}$ *and the fraction of security investment devoted to exploiting arbitrage exceeds* $\max\{w_{\text{eq}A}, w_{\text{eq}B}\}$*, then allocation vector* $\boldsymbol{w}$ *will tend toward* $\boldsymbol{w}_{\text{eq}}$ *whenever both SAAs are at rest and relative reward R is stable. (Proof in Appendix E.)*

Theorem 4 proves that if a sufficient quantity of miners act to exploit arbitrage, and some portion remain on each chain regardless of profitability, then the security allocation among those chains will tend toward a single equilibrium at times when both SAAs are at rest and relative reward is stable. This result has important implications for blockchain security and governance. It shows that (i) *a miner allocating hash rate off equilibrium (be it accidental or intentional) will not tend to move the equilibrium because his bias toward one chain will be offset by another miner exploiting the resulting arbitrage opportunity;* (ii) *to boost the proportion of hash rate on a given chain, one must move the market price of the chain's coin, not donate hash power.* These results contrast with those of Kwon et al. [9] who argue that, if a sufficient number of miners act to exploit arbitrage, then ultimately none of those miners will remain on the minority hash rate chain. Indeed, the empirical results of Section 7 support our conclusions by showing that the equilibrium is closely observed in practice.

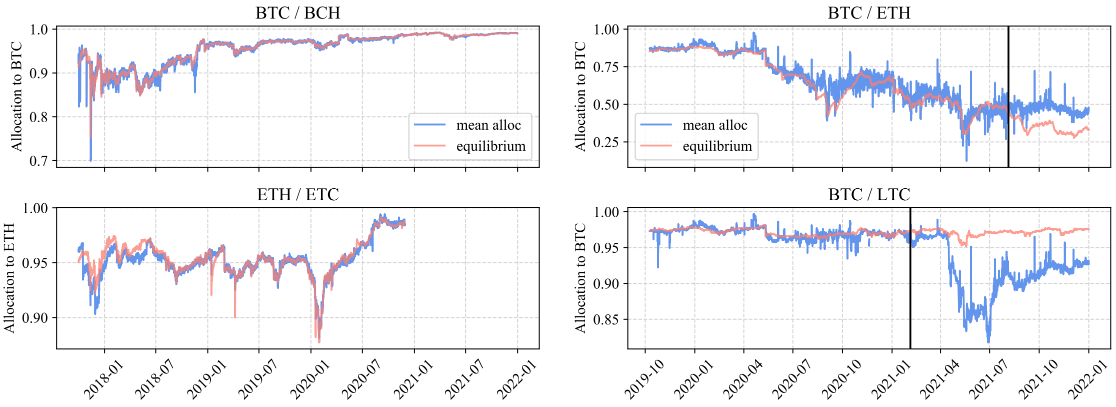
7 EVALUATION

In this section, we demonstrate empirically the formation of the security allocation equilibrium described variously in Sections 3, 5, and 6. Our theory is overwhelmingly supported by data at hourly granularity, with much lower error results than previous work [9] that used less granular data. Moreover, we illuminate security allocation relationships between blockchains previously believed to be unrelated.

Recall that the *actual* security allocation between two blockchains A and B is given by $\boldsymbol{w}$ (see Definition 4). In plain terms, a certain amount of hash rate is applied to both chains cumulatively and this hash rate has a fiat value (as determined by its trade price σ in a marketplace like NiceHash [27]). Vector $\boldsymbol{w}$ captures the relative fiat value devoted to security on each chain. Below, we show that the equilibrium point $\boldsymbol{w}_e$ (Theorem 2) closely matches the actual allocation $\boldsymbol{w}$ for historical data.

7.1 Data collection and preprocessing

We collected historical data for several of the largest PoW blockchains by market capitalization including Bitcoin (BTC), Bitcoin Cash (BCH), Ethereum (ETH), Ethereum Classic (ETC), and Litecoin (LTC). Included in the datasets were hourly fiat / coin exchange prices from the Bitfinex exchange [28] for dates prior to November 15, 2018 and from the Coinbase exchange [29] for dates after. Data from the Bitstamp [30] exchange were used for BCH only for the seven days following a contentious hard fork on November 15, 2018. We used publicly available Blockchain data in the Google BigQuery database [31]. We adjusted Blockchain constants such as target block time and block reward over time to match historical values. We gathered hash price data from NiceHash [27] for dates on or after October 10, 2019. We



(a) Same PoW algorithm.

(b) Different PoW algorithms.

Fig. 5. Actual hash rate allocation (blue) between two cryptocurrencies juxtaposed with the equilibrium allocation (red). ***(a)*** *The plots show strong agreement between the actual allocation and the allocation predicted by the equilibrium, the latter of which is based entirely on expected block times and block rewards and fees. Spikes in ETH / ETC are due to large transaction fees miners couldn't anticipate.* ***(b)*** *Prior to the exogenously induced equilibrium breakdown dates (indicated by vertical black bars), agreement with the equilibrium is strong, albeit with significant bias at times. Spikes in actual allocation are an artifact of sudden changes in hash price in the NiceHash marketplace.*

downloaded the hash price order book from NiceHash every 10 minutes, and used the mean price from those orders as the spot price. Our results are not significantly different when either median or best prices are used instead.

7.1.1 Estimating security allocation $\boldsymbol{w}$. We calculated the security allocation $\boldsymbol{w}$ between pairs of blockchains using Eq. 3. This required calculating $\hat{s}$, the estimated security investment for a given blockchain, whose major components are hash rate H and hash price σ. From the blockchain data, we were able to extract the *nominal* hash rate $H'(\tau)$ at time τ from its difficulty $D(\tau)$. For ETH and ETC, nominal hash rate is simply $H'(\tau) = D(\tau)/T$, where T is the target block time. For BTC, BCH, and LTC it is also necessary to multiply by *pool difficulty* (https://en.bitcoin.it/wiki/Difficulty), so that $H'(\tau) = 2^{32}D(\tau)/T$. H' was a sufficiently smooth estimator for all blockchains except BTC and LTC, which update their difficulty only once every 2016 blocks. For these two chains, we estimated the hash rate at time τ, $\hat{H}(\tau)$, by adjusting a rolling nominal hash rate with a rolling correction term based on observed block times. In particular $\hat{H}(\tau) = \texttt{ewma}(H'(\tau))/\texttt{ewma}(t(\tau))T$, where $t(\tau)$ is the actual block time for the given chain and ewma denotes the exponentially weighted moving average until time τ with 96-hour half-life. The choice of a 96-hour half-life performed best at correcting a small cyclic bias in nominal hash rate over the difficulty adjustment period. Thus, for input to Eq. 3 we use $\hat{s}(\tau) = \texttt{ewma}(\sigma(\tau))\texttt{ewma}(H'(\tau))/\texttt{ewma}(t(\tau))T$.

7.1.2 Estimating equilibrium between chains. We calculated the equilibrium allocation between two blockchains $\boldsymbol{w}_{\text{eq}}$ using Eq. 7. The only variable quantity in that equation is R, which is a function of the fiat value of rewards, $V(\tau)$, paid out each block on either chain. At time τ, fiat reward itself was calculated as the product of coinbase reward plus fees in the native currency, $k(\tau)$, and the fiat exchange rate, $P(\tau)$. For $P(\tau)$ we used the average of the high and low prices for each hour. Reward $k(\tau)$ varied only with transaction fees, which were highly variable for all blockchains. We smoothed fee values for each chain using an exponentially weighted moving average with a half-life of 96 hours. Smoothing fees

		RMSE	MAE	ME	PSNR
2018	BTC/BCH	0.0091	0.0058	−0.0014	40.8666
	ETH/ETC	0.0059	0.0042	−0.0041	44.6022
2019	BTC/BCH	0.0021	0.0016	0.0005	53.5335
	ETH/ETC	0.0038	0.0018	0.0009	48.4710
	BTC/ETH∗	0.0096	0.0071	0.0051	40.3120
	BTC/LTC∗	0.0031	0.0020	−0.0014	50.2742
2020	BTC/BCH	0.0022	0.0015	−0.0004	52.9738
	ETH/ETC*	0.0030	0.0021	−0.0006	50.5195
	BTC/ETH	0.0513	0.0342	0.0274	25.8010
	BTC/LTC	0.0055	0.0039	−0.0011	45.2470
2021	BTC/BCH	0.0012	0.0007	−0.0001	58.7253
	BTC/ETH*	0.0484	0.0388	0.0214	26.2990
	BTC/LTC*	0.0050	0.0034	−0.0030	46.0454
Cmp.	BTC/BCH (ours)	0.0146	0.0074	−0.0016	36.7025
Kwon	BTC/BCH [9]	0.0421	0.0268	−0.0160	27.5224

Table 1. *Equilibrium and actual allocation. Root mean square error, mean average error, mean error (all in allocation units; lower is better) and peak signal-to-noise ratio (in dB; higher is better) for hourly data. Asterisks indicate partial data for the given time period:* (i) *data for the BTC / ETH and BTC / LTC pairs begin on October 9, 2019,* (ii) *the last observation date for the ETH / ETC pair is November 22, 2020, and* (iii) *we truncate data for the pairs BTC / ETH and BTC / LTC in 2021 at the known breakdown dates of August 5 and February 4, respectively.* Cmp. Kwon *compares our results to Kwon et al. [9] for an overlapping period ranging from October 1, 2017 through most of 2018.*

is justified by the fact that miners cannot always redirect their mining resources in time to capitalize on an unusually high transaction fee, so they are more likely to assume average rather than instantaneous fees. Generally, multiple blocks arrived per hour, so to align coinbase reward with coin price (measured hourly), we used the average reward per hour.

7.2 Historical Convergence to Equilibrium

Figure 5a plots the actual security allocation $\boldsymbol{w}$ in blue for pairs of blockchains BTC / BCH (utilizing the SHA256 algorithm) and ETH / ETC (utilizing the DaggerHashimoto algorithm), with one pair per facet, along with the equilibrium allocation $\boldsymbol{w}_{\text{eq}}$, which is plotted in red. Table 1 also shows several error metrics for each pair, broken down by year. Overall agreement between the red and blue curves was excellent in both facets, which indicates convergence to the equilibrium defined by Theorem 2. The most notable deviations from equilibrium in the BTC / BCH plot occur periodically throughout 2017 and during two hard forks on the BCH chain, one late in 2017 and another in late 2018. Both forks resulted in a roughly week-long halt to nearly all fiat exchange of the BCH coin, which likely contributed to the disruption in the equilibrium. Other spikes in 2017 could be related to drastic difficulty changes in BCH brought on by the emergency difficult adjustment algorithm (EDA), which was removed in the hard fork on November, 2017. Figure 7 (see Appendix A) shows the trend in prediction error for the BTC / BCH pair over time. Agreement between actual and equilibrium allocations has tightened considerably since 2019, with bias becoming particularly low.

It can also be seen from Figure 5a that agreement between $\boldsymbol{w}$ and $\boldsymbol{w}_{\text{eq}}$ in the ETH / ETC plot is strong but with noticeable bias in 2017 and 2018. There are two major deviations between actual and equilibrium allocations. First, there are several prominent spikes evident in the equilibrium. These all originate from excessively large transaction fees in ETC, which we believe appeared too suddenly for miners to respond by reallocating hash power. The most prominent occurs around the time of a known attack on ETC [32]. After removing the top 0.1% largest fees, the spikes disappear. Second, agreement is generally poor (typically with equilibrium biased toward ETH) for dates prior to May 29, 2018, the date of a hard fork on the ETC chain, after which any bias abruptly vanishes. The hard fork removed a *difficulty bomb*, a piece of code that intentionally increases the difficulty (and therefore block time), so as to encourage miners to follow the fork. However, the difficulty bomb does not affect the calculation of nominal hash rate (several bombs active for ETH during this time period have no effect), and the bomb was not active before mid February 2018 even though the bias existed earlier than that. So we cannot identify a definitive reason for this early bias. Figure 8 in Appendix A shows the trend in overall error between actual and equilibrium allocation for the ETH / ETC pair. Data for the ETH / ETC plots runs only until November 22, 2020, the last day before a hardfork in ETC that activated ECIP-1099, which is a change to the PoW algorithm that enables lower-memory GPUs and effectively separates the set of devices used to mine ETH and ETC. Hash rate for this new PoW algorithm is not sold on NiceHash, therefore we could perform neither single- nor multi-PoW analysis.

Kwon et al. [9] also considered convergence of security allocation (hash rate) to the equilibrium among blockchains utilizing the same PoW algorithm. However, their analysis was considerably more limited and evaluated only graphically. Their dataset focussed exclusively on the SHA256 PoW algorithm and was limited to dates prior to 2019. Furthermore, it failed to account for some protocol nuances such as transaction fees and bias in BTC's nominal hash rate. As a result, their findings conveyed much looser adherence to the equilibrium. Table 1 (gray) shows the error for BTC / BCH from October 1, 2017, to December 15, 2018, comparing the results of Kwon et al. with ours (the set of dates our data overlapped with theirs). Note that we dropped dates from our data ranging from June 6 through August 12, 2018, because it was missing from their dataset. For root mean squared error, their results incur roughly three times the error of ours. Moreover, mean error for their data reveals strong bias (most likely due to missing transaction fees).

7.2.1 Multiple PoW Algorithms. Miner adherence to the equilibrium is remarkably reliable between blockchains that share the same PoW algorithm. More remarkable still is that the equilibrium continues to hold between blockchains that do not share PoW algorithms.

Similar to Figure 5a, Figure 5b plots actual security allocation $\boldsymbol{w}$, in blue, and equilibrium allocation $\boldsymbol{w}_{\text{eq}}$, in red, this time for pairs of blockchains BTC / ETH and BTC / LTC. Because these plots pair blockchains that do not share a PoW algorithm, arbitrage must be achieved by trading hash rate through a secondary market such as NiceHash. For each facet of the figure, we have delineated cutoff dates with a vertical black bar, which are dates when exogenous factors caused the equilibria to breakdown. For the BTC / ETH pair, this occurred on August 5, 2021 when EIP-1559 was activated on the Ethereum blockchain, which has the effect of burning some portion of fees paid to miners (lowering the value of mining an Ethereum block). And roughly on February 4, 2021 for the BTC / LTC pair, when the price of Doge coin began to skyrocket after Elon Musk first tweeted about it. Doge coin is merge mined with Litecoin, i.e., mining a block in Litecoin can automatically mine a Doge coin block. Therefore, Doge coins exert exogenous influence on the BTC / LTC allocation equilibrium; security implication of merged mining have been explored in the past [33].

The remainder of our discussion of Figure 5b focusses on dates prior to the equilibrium breakdowns indicated by black bars. The plots show generally good agreement, in terms of both magnitude and correlation between curves, but the equilibrium allocation during that time does exhibit significant bias for some months for both blockchain pairs. Table 1 shows that the equilibrium for BTC / LTC typically deviates from the actual allocation with overall error of the same order as was observed for single PoW pairs. In contrast to the BTC / LTC pair, deviation between the equilibrium and actual allocation of BTC / ETH shows error roughly 10 times greater than that of single PoW pairs. Bias is similarly elevated. Nevertheless, both root mean square error and mean absolute error remain below 6%.

Overall, the results suggest that *the Blockchain Security Market seeks the point of no arbitrage even if it can only be accessed through secondary hash rate markets*. We hypothesize that it is inefficiency in the the hash rate market itself that introduces the higher error in the agreement between equilibrium and actual allocations. Finally, we note that Figure 5b also provides a rare opportunity to compare the security investment of blockchains that use different PoW algorithms.

8 CAUSAL ANALYSIS

Section 7 depicts a strong historical correlation between security allocation and the allocation equilibrium predicted by Theorem 2. In this section, we dig deeper into the relationship between actual and equilibrium allocations among chains sharing a PoW algorithm. Specifically, we ask, *to what extent does change in actual allocation invoke change in the equilibrium, and vice versa*? Actual security allocation $\boldsymbol{w}$ is a function of the hash rate that miners devote to each chain, while the equilibrium allocation $\boldsymbol{w}_{\text{eq}}$ is a function of the fiat exchange price of the coins native to those blockchains. Thus, the question asks how hash rate and coin price mutually influence each other.

By evaluating *Granger causality* [14] between these quantities, we find evidence that coin price influences hash rate allocation, which implies that miner security allocation follows the equilibrium. However, the opposite is not typically true: hash rate allocation rarely exhibits a causal effect on coin price. This is not to say that increased hash rate cannot move coin price, only that we find scant evidence for it on a systematic, hourly basis.

These findings have profound implications for blockchain security and governance. First, they imply that blockchains with fixed coin issuance and low coin value are destined to suffer from commensurately low security so long as their coin's price is suppressed. Second, dramatic changes in coin price, which are commonly observed in the cryptocurrency realm, can cause equally sudden changes in security. Third, we find little evidence that security improvements (reductions) are rewarded (punished) by the market. This does not rule out the possibility that it happens occasionally, but the signal is typically very weak.

Granger causality is a method used to establish causality between two time series with the simple rationale that a later event cannot give rise to an earlier one. This notion of causality is weaker than the "gold standard" obtained from controlled experiments, which are very difficult to conduct in real markets. Granger causality assumes that there exists no unobserved third variable influencing events in both series with different latency. With this caveat in mind, we proceed by estimating pairs of regression equations, each with a time-lagged version of the other as a predictor.

Regression on a time series amounts to extracting a stochastic process from temporal data. As a byproduct of the temporal nature of the data, standard regression techniques can often lead to dependent residuals, which imply the process is non-stationary, compromising the validity of statistical inference [34]. The problem manifests with the existence of unit (i.e., trivial) roots in the characteristic regression equation. The standard solution is to differentiate the dependent variables in the equation several times until the unit-root vanishes. Table 3 (see Appendix A) shows that this happens after calculating first differences for all our series of interest. This implies that, while raw security allocations and equilibrium points are not stationary, hourly changes in these variables are. Therefore the analyses in this section refer to first differences of series calculated from empirical data. This transformation does not affect the logic behind Granger causality.

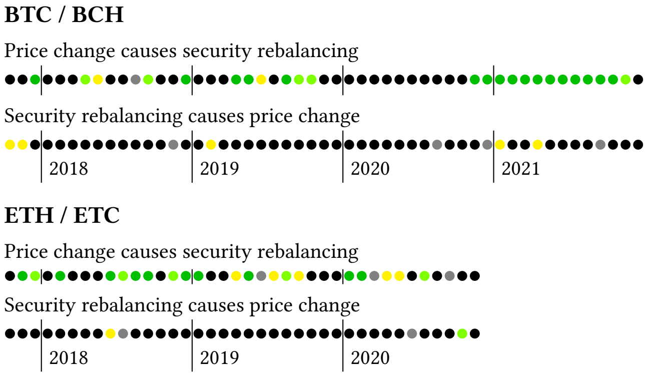


Fig. 6. *Monthly results for the Granger causality analysis. The strength of the evidence for a causal link is color-coded by the statistical significance of the F-test as follows:* ● absent *($p > 0.1$),* ● marginal *($p \leq 0.1$),* ● weak *($p \leq 0.05$),* ● moderate *($p \leq 0.01$), and* ● strong *($p \leq 0.001$). Input data: hourly first differences.*

To determine if change in coin exchange price (labeled *price change*), as the main component of the equilibrium, Granger-causes actual security allocation rebalancing (labeled *security rebalancing*), we fit the following two specifications,

$$\Delta w_t + \varepsilon_t = a + b_1 \cdot \Delta w_{t-1} \tag{9}$$

$$\Delta w_t + \varepsilon_t = a + b_1 \cdot \Delta w_{t-1} + b_2 \cdot \Delta w_{\mathrm{eq}_{t-1}}, \tag{10}$$

and test if the additional term related to coefficient b_2 in Specification 10 improves explanatory power over Specification 9. Likewise, we check if security rebalancing Granger-causes change in equilibrium allocation (a proxy for price change) by fitting

$$\Delta w_{\mathrm{eq}_t} + \varepsilon_t = a + b_1 \cdot \Delta w_{\mathrm{eq}_{t-1}} \tag{11}$$

$$\Delta w_{\mathrm{eq}_t} + \varepsilon_t = a + b_1 \cdot \Delta w_{\mathrm{eq}_{t-1}} + b_2 \cdot \Delta w_{t-1}, \tag{12}$$

and performing the same test. All models are fit by minimizing the squares of the residuals ε_t.

Figure 6 shows the strength of Granger-causal link for blockchain pairs BTC / BCH (top facet) and ETH / ETC (bottom facet) from price change to security rebalancing (top row) and security rebalancing to price change (bottom row). In particular, the figure reports the probability of a type I error in choosing Specification 10 (top row) over 9 and choosing Specification 12 over 11 (bottom row). Dark green dots indicate very low p-value, or a strong Granger-causal link, while black dots indicate a very high p-value, or no Granger-causal link.

On a systematic, hourly basis, we find evidence that security rebalancing tends to follow price change, although the evidence vanishes for months at a time before reappearing. Figure 6 shows that this has been the case for both the BTC / BCH and ETH / ETC pairs. The BTC / BCH pair vacillates between strong and absent significance levels throughout 2018 and 2019. For most of 2020, we find no evidence of Granger causality. We speculate that this could be due to markets rebalancing faster than our hourly data can capture, or due to relative stability of the equilibrium that makes a causal signal difficult to detect. In support of the latter point, Granger causality reemerges strongly in the last months of 2020 and throughout much of 2021, which could be due to the concurrent run-up in the price of both coins that leads to more volatility in the equilibrium.

Evidence that hash rate rebalancing tends to follow price change also exists for the ETH / ETC pair. We see behavior similar to the BTC / BCH pairs throughout 2018 and 2019. The pattern continues through 2020. In particular, months of strong evidence are often followed by months with weak or no evidence of Granger causality. Unfortunately, ETC changed its PoW algorithm at the end of 2020 (see Section 7.2) making it impossible to establish an equilibrium during 2021.

In rare circumstances, evidence supports the conclusion of *reverse causality*: that the market looks to security rebalancing to readjust price. One reason for the rarity of this link might be that a change in security allocation away from the equilibrium is quickly offset by other miners shifting their hash rate in the opposite direction (a behavior predicted by Theorem 4). We fail to see anything more than occasional weak evidence for reverse causality in the BTC / BCH pair. For the ETH / ETC pair, only the month of October, 2020, showed moderate evidence of reverse causality. There is no definitive explanation, but a notable event during that month was a surprise chain spilt [35] that briefly disrupted consensus and transactional throughput on Ethereum.

We leave deeper Granger-causal analysis for future work including more variables (if observable), a broader class of specifications, and investigation of conditions that can lead to causal links. Of particular interest would be an investigation of the impact of equilibrium stability as a possible confounder in the observability of Granger causality.

9 TRUSTLESS PRICE-RATIO ORACLE

Price oracles are a fundamental tool for many popular smart contract applications [36–38], particularly in the space of decentralized finance (DeFi) [39–41]. Oracles typically pull data from trusted, centralized sources [29, 42, 43], decentralized exchanges [44, 45], or from crowds [46, 47]. Centralized sources require trust in a corruptible third-party, while crowd sourcing and decentralized exchanges are subject to manipulation. For example lending platform bZx lost the equivalent of nearly $1M USD due to exchange price manipulation [48].

In this section, we describe a smart contract `Oracle` that leverages the allocation equilibrium described by Theorem 2 to provide an estimate of the fiat *price ratio* of the cryptocurrencies A and B from information contained in block headers only. It can either be used alone or aggregated with existing solutions to increase robustness. An example of a futures contract leveraging `Oracle` appears in Appendix C. In estimating the price ratio of two coins sharing the same PoW algorithm, `Oracle` can be no more easily manipulated than the PoW that secures each chain. Details of an adjustment for coins that use different PoW algorithms are included in the full version of this paper.

Smart contract `Oracle` runs on chain A, returning an estimate of the price ratio P_B/P_A when the two chains are each at a given block height. It does so by implementing a light client for blockchain B. Two public methods are exposed: $\texttt{Update}(h_B)$ and $\texttt{Query}(b_A, b_B, \sigma_\Delta)$ (see Algorithms 1 and 2 in Appendix D). Method $\texttt{Update}(h_B)$ allows any user to update the chain of headers with a new header h_B having the following properties: *(i)* the previous block hash of h_B points to the block hash of the previous header; and *(ii)* the PoW associated with the hash of h_B meets the difficulty implied by earlier headers and chain B's protocol. If either of the conditions are not met, then `Update` returns an error.

Method $\texttt{Query}(b_A, b_B, \sigma_\Delta)$ returns an estimate of the price ratio P_B/P_A at the time when chain A was at block height b_A, chain B was at height b_B, and the ratio of spot hash prices is equal to σ_Δ, i.e., $\sigma_\Delta = \frac{\sigma_B}{\sigma_A}$. If either *(i)* the header at block height b_B is unknown to `Oracle` or *(ii)* the block on chain A at height b_A has not yet been mined, then an error is thrown. We assume any party interested in querying the oracle will be incentivized to run $\texttt{Update}(h_B)$ for all new headers h_B.

The initial state of contract `Oracle` is comprised of list $\texttt{Headers}_B = [h^*_B]$ where h^*_B is the header for the genesis block on chain B. The latest list of headers, $\texttt{Headers}_A$, is native to blockchain A and is therefore assumed to be accessible from within `Oracle`. Furthermore, let $h_X[g]$, $h_X[D]$, and $h_X[P]$ denote the target, difficulty, and hash of the previous block, respectively, specified in header h_X for $X \in \{A, B\}$. Finally, define $\texttt{Headers}_X[\texttt{-1}]$ to be the last item added to list $\texttt{Headers}_X$ and let $\mathcal{H}_X(h_X)$ denote the hash of header h_X.

Using difficulties D_A and D_B, extracted from the headers on chains A and B, and spot hash price ratio σ_Δ, `Oracle` estimates P_B/P_A by equating w_A from Eq. 3 and w_A from Theorem 2: $w_A \approx \hat{s}_A/(\hat{s}_A + \hat{s}_B)$, where w_A denotes the portion of allocation among chains A and B devoted to chain A and $\hat{s}_X = \sigma_X \frac{D_X}{T_X}$. It follows that,

$$\begin{aligned}
&\tfrac{\hat{s}_A}{\hat{s}_A+\hat{s}_B} \approx \tfrac{T_B R}{T_B R - T_A R + T_A} \implies \\
&\tfrac{P_B}{P_A} \approx \tfrac{k_A}{k_B T_A}\left(\tfrac{T_B(\hat{s}_A+\hat{s}_B)}{\hat{s}_A} - T_B + T_A\right) - \tfrac{k_A}{k_B} \qquad (13) \\
&\tfrac{P_B}{P_A} \approx \tfrac{k_A T_B}{k_B T_A}\tfrac{\hat{s}_B}{\hat{s}_A} = \tfrac{k_A}{k_B}\tfrac{D_B}{D_A}\tfrac{\sigma_B}{\sigma_A} = \sigma_\Delta \tfrac{k_A}{k_B}\tfrac{D_B}{D_A}.
\end{aligned}$$

When blockchains A and B use the same PoW algorithm, $\sigma_\Delta = 1$, and Eq. 13 simplifies to $\frac{P_B}{P_A} \approx \frac{k_A}{k_B}\frac{D_B}{D_A}$. In this case, all information required by `Oracle` is either provided to the contract by way of the `Update` method or is accessible natively on chain A.

Figure 5a and the MAE from Table 1 demonstrate that the equilibrium agrees strongly with the actual allocation when chains A and B use the same PoW algorithm. Thus, we can verify price-ratio predictions would have been accurate within less than 1% error.

10 RELATED WORK

Prat and Walter [49] model the impacts of mining difficulty and coin exchange rate on profitability. Ma et al. [50] show that there exists a Nash equilibrium for the computing power allocated by miners given a fixed mining difficulty. Kristoufek [51] derives an equilibrium between miner hash rate production and PoW energy costs in Bitcoin mining. Huberman et al. [52] devise an economic model of the Bitcoin payment system that captures the tension between users who compete for transaction processing capacity provided by miners. Noda et al. [53] argue that the elastic supply of hash rate due to price fluctuations can render the Bitcoin DAA ineffective. Biais et al. [54] identify Markov-perfect equilibria in miner consensus; their analysis is primarily theoretical with only anecdotal supporting evidence.

Huang et al. [55] describe short-term investing and mining strategies for cryptocurrencies relative to base currencies Litecoin and Bitcoin. Nguyen et al. [56] show that new cryptocurrencies have a small but significant negative impact on the price of Bitcoin. Both stop short of identifying hash rate allocation equilibria. Gandal et al. [57] analyzes price manipulation on the Mt. Gox exchange, concluding that it was carried out by a small group. Today there exist many exchanges, centralized and decentralized, which makes such manipulation more difficult.

Meshkov et al. [58] analyze *coin-hopping*, where miners move among blockchains using the same PoW according to which is most profitable; see also [59, 60]. Shibuya et al. [61] provide statistical evidence that mining profitability correlates strongly with hash rate. Several works determine the optimal hash rate allocation between blockchains for *individual* miners or mining pools; e.g. [62–64].

Spiegelman et al. [8] apply the theory of Potential Games [65] to the problem of miner hash rate allocation across multiple blockchains. They prove that multiple stable equilibria can exist, and that they can be achieved without the use of a sophisticated utility function. However, they provide no means to explicitly identify equilibria, nor is it clear from their work how a single equilibrium is achieved among the multiple possibilities. Altman et al. [66] reach similar conclusions using a different model of hash rate allocation across cryptocurrencies and mining pools.

Han et al. [67] investigate doublespending on blockchains with relatively low hash rate instigated by either miners from a higher hash rate chain or attackers who purchase hash rate from a marketplace such as NiceHash [27]. Sapirshtein et al. [25] and Gervais et al. [68] apply MDPs to blockchains to analyze selfish mining [21] and double spend attacks.

There is much existing work in the finance literature related to *no arbitrage* (NA) conditions and the *law of one price* (LOOP) in the presence of short sale restrictions. Discrete-time models, like the one used in this document include: LeRoy et al. [15], Chichilnisky [69], He et al. [70], and Oleaga [71]. Kroeger and Sarkar [72] show that the LOOP does *not* hold in the Bitcoin / fiat exchange market. Yaish and Zohar [73] use the NA principle to price ASIC mining hardware.

The work of Kwon et al. [9] is most closely related to ours. They show that there exist multiple Nash equilibria for miners who allocate their hash rate among two blockchains sharing the same PoW algorithm. One of their equilibria coincides with $\boldsymbol{w}_{\text{eq}}$, the equilibrium we study, which they demonstrate is observed in practice. However, their model is limited compared to ours (see Table 2 in Appendix A). It assumes that the DAA of the blockchain with majority hash rate updates slowly; our model merely assumes that DAAs eventually come to rest (we characterize the conditions for the validity of this assumption in Corollary 2). The model of Kwon et al. also assumes that both blockchains share the same PoW Algorithm, whereas ours allows for them to differ. Among the top 100 blockchains by market capitalization at the time of writing, these two assumptions effectively limit their model to equilibria between Bitcoin and Bitcoin Cash or BitcoinSV. Finally, it is difficult to determine which equilibrium of Kwon et al. should apply at any given time because the equilibria generally depend on an unobservable quantity of *stick* miners.

The approach of Kwon et al. is also problematic in several ways. Their game mixes dynamic (loyal) with static (fickle) strategies, which causes confusion over when a given strategy is valid and how it should be applied. The fickle mining behavior is implausible; such miners are assumed to update their allocation only when the DAA changes, ignoring coin price swings in between. And all theoretical results concern fickle and loyal behaviors exclusively, but the main conclusions — the inevitable loss of miners on minority hash rate chains in the presence of automatic (arbitrage seeking) miners and injuring rival coins — are stated informally in terms of automatic mining.

11 CONCLUSION

We have presented a novel theory of the fiat value of security allocated among PoW blockchains, which is supported with empirical evidence and novel applications. Our principle finding is that, for any pair of cryptocurrencies, not necessarily sharing the same PoW algorithm, there exists a unique equilibrium allocation, based on market prices only, that is robust even to intentional manipulation of miner hash rate. We furthermore establish a strong Granger-causal link from market price change to change in security allocation, the opposite link is found to hold only under exceptional circumstances. We end with a trustless price ratio oracle that leverages the allocation equilibrium. The generality of our framework opens new doors for future work; our characterization of security in terms of opportunity cost can generalize to other consensus protocols, such as PoS.

REFERENCES

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” May 2009.

[2] J. Douceur, “The Sybil Attack,” in *Proc. Intl Wkshp on Peer-to-Peer Systems (IPTPS)*, Mar. 2002.

[3] S. King and S. Nadal, “Peercoin,” https://www.peercoin.net/whitepapers/peercoin-paper.pdf, August 2012.

[4] A. L. Goodkind, B. A. Jones, and R. P. Berrens, “Cryptodamages: Monetary value estimates of the air pollution and human health impacts of cryptocurrency mining,” *Energy Research & Social Science*, vol. 59, 2020.

[5] L. Badea and M. C. Mungiu-Pupazan, “The Economic and Environmental Impact of Bitcoin,” *IEEE Access*, vol. 9, pp. 48 091–48 104, 2021.

[6] M. Möser, R. Böhme, and D. Breuker, “An Inquiry into Money Laundering Tools in the Bitcoin Ecosystem,” in *Proc. APWG eCrime Researchers Summit*, 2013.

[7] S. Kethineni, Y. Cao, and C. Dodge, “Use of Bitcoin in Darknet Markets: Examining Facilitative Factors on Bitcoin-Related Crimes,” *American Journal of Criminal Justice*, vol. 43, no. 2, pp. 141–157, 2018.

[8] A. Spiegelman, I. Keidar, and M. Tennenholtz, “Game of Coins,” https://arxiv.org/abs/1805.08979, May 2018.

[9] Y. Kwon, H. Kim, J. Shin, and Y. Kim, “Bitcoin vs. Bitcoin Cash: Coexistence or Downfall of Bitcoin Cash?” in *IEEE Symposium on Security and Privacy*, February 2019, pp. 935–951.

[10] J. Zhuoer, “Infrastructure Funding Plan for Bitcoin Cash,” https://medium.com/@jiangzhuoer/infrastructure-funding-plan-for-bitcoin-cash-131fdcd2412e, January 2020.

[11] B. Laurie and R. Clayton, “Proof-of-Work Proves Not to Work,” in *Workshop on Economics and Information, Security (WEIS)*, 2004.

[12] D. L. Liu and J. Camp, “Proof of Work can Work,” in *Workshop on Economics and Information, Security (WEIS)*, 2006.

[13] J. Becker, D. Breuker, T. Heide, J. Holler, H. P. Rauer, and R. Böhme, “Can We Afford Integrity by Proof-of-Work? Scenarios Inspired by the Bitcoin Currency,” in *Workshop on Economics and Information, Security (WEIS)*, 2012.

[14] C. Granger, “Investigating causal relations by econometric models and cross-spectral methods,” *Econometrica*, vol. 37, pp. 424–438, 1969.

[15] S. F. LeRoy and J. Werner, *Principles of Financial Economics.* Cambridge University Press, 2014.

[16] A. Back, “Hashcash - Amortizable Publicly Auditable Cost-Functions,” http://www.hashcash.org/papers/amortizable.pdf, 2002.

[17] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A Survey on the Security of Blockchain Systems,” *Future Generation Computer Systems*, vol. 107, 2020.

[18] A. Biryukov, D. Khovratovich, and I. Pustogarov, “Deanonymisation of Clients in Bitcoin P2P Network,” in *ACM SIGSAC Conference on Computer and Communications Security*, 2014.

[19] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, “Eclipse Attacks on Bitcoin’s Peer-to-peer Network,” in *USENIX Security*, 2015.

[20] M. Porta, “Timewarp Attack: how to reduce the mining difficulty,” https://en.cryptonomist.ch/2019/05/20/timewarp-attack-mining-difficulty, 2019.

[21] I. Eyal and E. G. Sirer, “Majority is not enough: Bitcoin mining is vulnerable,” in *International conference on financial cryptography and data security.* Springer, 2014, pp. 436–454.

[22] M. del Castillo, “The DAO Attacked: Code Issue Leads to $60 Million Ether Theft,” http://www.coindesk.com/dao-attacked-code-issue-leads-60-million-ether-theft, June 2016.

[23] R. Böhme, L. Eckey, T. Moore, N. Narula, T. Ruffing, and A. Zohar, “Responsible vulnerability disclosure in cryptocurrencies,” *Communications of the ACM*, vol. 63, no. 10, pp. 62–71, 2020.

[24] M. Rosenfeld, “Analysis of hashrate-based double-spending,” https://bitcoil.co.il/Doublespend.pdf, December 2012.

[25] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, “Optimal Selfish Mining Strategies in Bitcoin,” https://arxiv.org/pdf/1507.06183.pdf, July 2015.

[26] H. McCook, “The Cost & Sustainability of Bitcoin,” https://www.academia.edu/37178295/The_Cost_and_Sustainability_of_Bitcoin_August_2018_, August 2018.

[27] “NiceHash,” https://www.nicehash.com.

[28] “Bitfinex,” https://www.bitfinex.com.

[29] “Coinbase,” http://pro.coinbase.com.

[30] “Bitstamp,” https://www.bitstamp.net.

[31] “Bigquery,” https://cloud.google.com/bigquery.

[32] M. Nesbitt, “Deep Chain Reorganization Detected on Ethereum Classic,” https://blog.coinbase.com/ethereum-classic-etc-is-currently-being-51-attacked-33be13ce32de, January 2019.

[33] A. Judmayer, A. Zamyatin, N. Stifter, A. Voyiatzis, and E. Weippl, “Merged Mining: Curse or Cure?” in *Proc. Cryptocurrencies and Blockchain Technology (CBT)*, 2017.

[34] J. Sargan and A. Bhargava, “Maximum Likelihood Estimation of Regression Models with First Order Moving Average Errors when the Root Lies on the Unit Circle,” *Econometrica: Journal of the Econometric Society*, pp. 799–820, 1983.

[35] C. Harper, “Ethereum’s Unannounced Hard Fork Was Trying to Prevent the Very Disruption It Caused,” https://www.coindesk.com/ethereums-hard-fork-disruption, November 2020.

[36] “Gnosis,” https://gnosis.io.

[37] “Augur,” https://www.augur.net.

[38] “Numerai,” https://numer.ai.

[39] “dYdX,” https://dydx.exchange.

[40] “Compound Finance,” https://compound.finance.

[41] “Dharma,” https://blog.dharma.io.

[42] “MakerDAO,” https://makerdao.com.

[43] “Provable,” http://provable.xyz.

[44] “Uniswap,” https://uniswap.org.

[45] “Bancor,” https://www.bancor.network.

[46] whgeorge, “Decentralized price oracle,” https://ethresear.ch/t/decentralized-price-oracle/1941, May 2018.

[47] “Dutchx,” https://fairdex.net.

[48] P. Shield, “bZx Hack Full Disclosure,” https://medium.com/@peckshield/bzx-hack-full-disclosure-with-detailed-profit-analysis-e6b1fa9b18fc.

[49] J. Prat and B. Walter, “An Equilibrium Model of the Market for Bitcoin Mining,” https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3143410, February 2018.

[50] J. Ma, J. S. Gans, and R. Tourky, “Market Structure in Bitcoin Mining,” https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3103104, June 2019.

[51] L. Kristoufek, “Bitcoin and its mining on the equilibrium path,” *Energy Economics*, vol. 85, 2020.

[52] G. Huberman, J. D. Leshno, and C. Moallemi, “An Economic Analysis of the Bitcoin Payment System,” http://fetch.econ.cam.ac.uk/papers/SSRN-id3025604-Huberman.pdf, March 2019.

[53] S. Noda, K. Okumura, and Y. Hashimoto, “An Economic Analysis of Difficulty Adjustment Algorithms in Proof-of-Work Blockchain Systems,” https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3410460, June 2019.

[54] B. Biais, C. Bisière, M. Bouvard, and C. Casamatta, “The Blockchain Folk Theorem,” *The Review of Financial Studies*, vol. 32, 2019.

[55] D. Y. Huang, K. Levchenko, and A. C. Snoeren, “Estimating Profitability of Alternative Cryptocurrencies,” in *Proc. Financial Cryptography and Data Security*, 2018.

[56] T. V. H. Nguyen, B. T. Nguyen, T. C. Nguyen, and Q. Q. Nguyen, “Price manipulation in the Bitcoin ecosystem,” *Research in International Business and Finance*, vol. 48, 2019.

[57] N. Gandal, J. Hamrick, T. Moore, and T. Oberman, “Price manipulation in the Bitcoin ecosystem,” *Journal of Monetary Economics*, vol. 95, 2018.

[58] D. Meshkov, A. Chepurnoy, and M. Jansen, “Revisiting Difficulty Control for Blockchain Systems,” in *Proc. Cryptocurrencies and Blockchain Technology (CBT)*, 2017.

[59] T. Király and L. Lomoschitz, "Profitability of the coin-hopping strategy," http://web.cs.elte.hu/egres/www/qp-18-03.html, March 2018.

[60] N. Tovanich, N. Soulie, N. Heulot, and P. Isenberg, "An Empirical Analysis of Pool Hopping Behavior in the Bitcoin Blockchain," in *IEEE International Conference on Blockchain and Cryptocurrency*, 2021.

[61] Y. Shibuya, G. Yamamoto, F. Kojima, E. Shi, S. Matsuo, and A. Laszka, "Selfish Mining Attacks Exacerbated by Elastic Hash Supply," in *Intl. Conf. on Financial Cryptography and Data Security*, 2021.

[62] G. Bissias, B. Levine, and D. Thibodeau, "Using Economic Risk to Model Miner Hash Rate Allocation in Cryptocurrencies," in *Proc. Cryptocurrencies and Blockchain Technology (CBT)*, 2018.

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

[64] P. Chatzigiannis, F. Baldimtsi, I. Griva, and J. Li, "Diversification Across Mining Pools: Optimal Mining Strategies under PoW," in *Workshop on the Economics of Information Security (WEIS)*, 2019.

[65] D. Monderer and L. S. Shapley, "Potential Games," in *Games and Economic Behavior*, vol. 14, no. 1, 1996, pp. 124–143.

[66] E. Altman, A. Reiffers, D. S. Menasche, M. Datar, S. Dhamal, and C. Touati, "Mining competition in a multi-cryptocurrency ecosystem at the network edge: A congestion game approach," *SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 114–117, Jan. 2019.

[67] R. Han, Z. Sui, J. Yu, J. Liu, and S. Chen, "Sucker punch makes you richer: Rethinking Proof-of-Work security model," https://eprint.iacr.org/2019/752, June 2019.

[68] A. Gervais, G. O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the Security and Performance of Proof of Work Blockchains," https://eprint.iacr.org/2016/555, 2016.

[69] G. Chichilnisky, "Limited arbitrage is necessary and sufficient for the existence of a competitive equilibrium with or without short sales," *Economic Theory*, vol. 5, no. 1, 1995.

[70] H. He and N. D. Pearson, "Consumption and portfolio policies with incomplete markets and short-sale constraints: The infinite dimensional case," *Journal of Economic Theory*, vol. 54, no. 2, pp. 259–304, 1991.

[71] G. E. Oleaga, "Arbitrage conditions with no short selling," *Boletín de Matemáticas*, 2012.

[72] A. Kroeger and A. Sarkar, "The Law of One Bitcoin Price," Federal Reserve Bank of Philadelphia, 2017.

[73] A. Yaish and A. Zohar, "Pricing ASICs for Cryptocurrency Mining," https://arxiv.org/pdf/2002.11064.pdf, February 2020.

[74] A. P. Ozisik, G. Bissias, and B. N. Levine, "Estimation of Miner Hash Rates and Consensus on Blockchains," University of Massachusetts, Amherst, MA, Tech. Rep. arXiv:1707.00082, July 2017.

A LIST OF SYMBOLS AND SUPPLEMENTAL FIGURES

Symbol	Description
A, B	Either an arbitrary blockchain (chain) or its native coin
X	Variable identifying a chain such that $X \in \{A, B\}$
$\mathtt{ALG}_X$	The PoW algorithm for chain X
M_X	The set of miners capable of performing PoW W_X
M	The union of miners M_A and M_B
H_X	The number of W_X hashes per second on chain X
T_X	Protocol targeted block inter-arrival time for chain X
t_X	Actual block inter-arrival time for chain X
τ	Time since epoch
σ_X	Fiat value of a single hash using W_X
S_X	Target security investment on chain X
s_X	Actual security investment on chain X
$\boldsymbol{w}$	Security allocation vector among chains A and B
$\boldsymbol{w}_i$	Security allocation for m_i among chains A and B
$\boldsymbol{u}_i$	Share of reward on chains A and B for miner m_i
V_X	Fiat value of coinbase reward plus fees
k_X	Number of coins in coinbase reward plus average fees
P_X	Fiat value of a single coin from chain X
D_X	Difficulty, expected hashes required to mine a block on chain X
R	Relative reward for mining on chain A
N	Total number of miners in the Security Allocation Game
m_i, m_{-i}	Miner i and all other miners, respectively
$\boldsymbol{\pi}$	Expected fiat payoff
e, e_i	Initial fiat endowment in aggregate and for miner m_i
$\boldsymbol{c}, \boldsymbol{c}_i$	Claim vector (of payoff) in aggregate and for m_i
$\boldsymbol{p}$	Portfolio pricing vector
$\Delta \boldsymbol{w}$	Allocation vector rebalancing

	Kwon et al.	Present Work
DAA behavior	Majority chain is slow	Both at rest
Price behavior	Ratio of coin prices stable	Ratio of coin prices stable
PoW algorithms	Both the same	Can differ
Endogenous strategies	Fickle, loyal, and automatic	Automatic (exploit arbitrage)
Miner knowledge	Number of stick miners, current prices, and difficulties	Current prices and difficulties

Table 2. Assumptions of Kwon et al. vs. the present work.

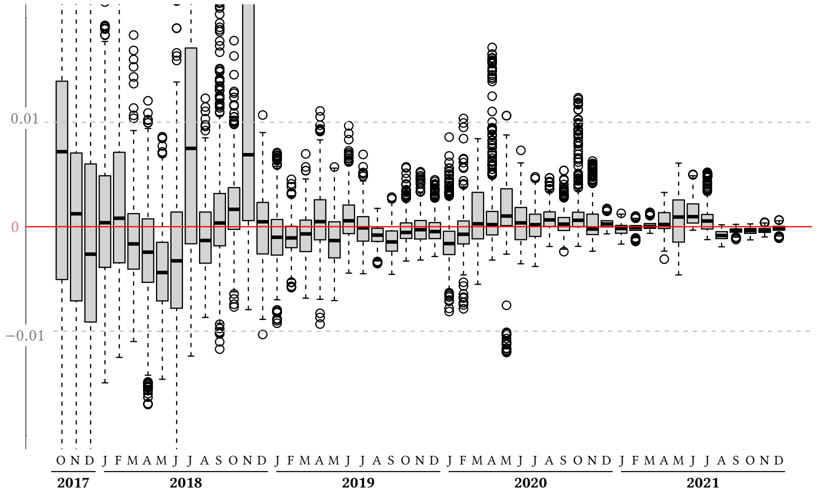


Fig. 7. *Evolution of the BTC / BCH prediction error over time. Monthly distributions of hourly differences between equilibrium and actual hash rate allocation (in allocation units). Positive values mean that, from market prices, our theory predicts a higher allocation of hash power to BTC than the actual value. The data ranges from October 1, 2017 until December 31, 2021.*

	Raw ratios		1st differences	
Time series	Statistic	p	Statistic	p
BTC/BCH actual	−3.44	0.05	−39.9	< 0.01
BTC/BCH equilibrium	−2.39	0.41	−32.5	< 0.01
ETH/ETC actual	−1.86	0.64	−32.5	< 0.01
ETH/ETC equilibrium	−1.95	0.60	−31.0	< 0.01

Table 3. *Check of preconditions. Augmented Dickey–Fuller tests for unit-roots in the hourly time series used for Granger causality. Series with p-values rejecting the null hypothesis fulfill the conditions that make the asymptotic theory valid.*

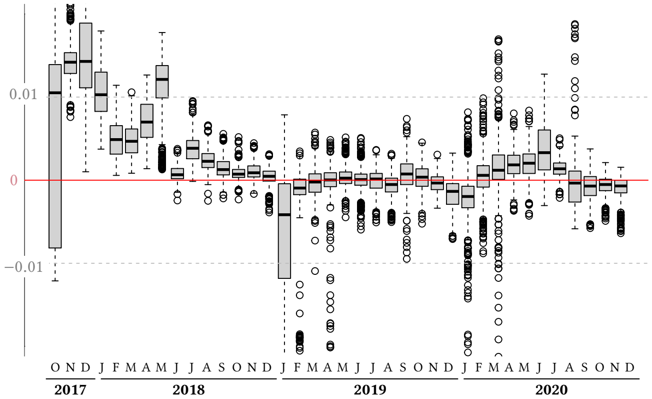


Fig. 8. *Evolution of the ETH/ETC prediction error over time. Monthly distributions of hourly differences between equilibrium and actual hash rate allocation (in allocation units). Positive values mean that for the observed market prices, our theory predicts a higher allocation of hash power to ETH than estimated from block times. The data ranges from October 1, 2017 until November 23, 2020 (the end of ETC block data on our source).*

B MARKOV DECISION PROCESS (MDP) DETAILS

At a high level, our MDP is comprised of states corresponding to the current difficulty on each chain, actions correspond to the allocation of available hash rate between the two chains, and transitions occur every second. Details are given below.

States. Each state is a tuple of the form $(D_A, D_B, \beta_A, \beta_B)$, where D_A and D_B are the difficulties on chains A and B, respectively, given in terms of the expected number of hashes per block. Each chain is additionally given a single bit β_A or β_B, which indicates that a bock is mined on the given chain when the bit is flipped from one state to the next.

Actions. Each action a corresponds to the amount of hash rate allocated to chain A for the next second. The remaining hash rate b is allocated to chain B so that $a + b = 6$, i.e. the total hash rate is equal to 6 hashes per second.

Transitions. A transition corresponds to an update in difficulty and block mining status, it occurs any time one of those values change on either blockchain. The only valid change in difficulty is to move from D_X to $D'_X = 2x$, for $X \in \{A, B\}$, which indicates that when a block is mined the difficulty updates to twice the hash rate applied to the chain (to enforce 2 second block times). For $X \in \{A, B\}$, let $P[D_X, x]$ be the probability that a block is mined on chain X after 1 second, for the given difficulty and hash rate. Similarly, define $C[D_X, x]$ to be the quantity of blocks mined on chain X during a 1 second interval, given that at least one block will be mined. We have the following transition probabilities and rewards where *success* on a given chain is defined as the event of mining at least one block.

- **Both success:** $(D_A, D_B, \beta_A, \beta_B) \rightarrow (D'_A, D'_B, 1 - \beta_A, 1 - \beta_B)$
  Probability: $P[D_A, a]P[D_B, b]$
  Reward: $2C[D_A, a] + C[D_B, b]$
- A **success:** $(D_A, D_B, \beta_A, \beta_B) \rightarrow (D'_A, D'_B, 1 - \beta_A, \beta_B)$
  Probability: $P[D_A, a](1 - P[D_B, b])$
  Reward: $2C[D_A, a]$
- B **success:** $(D_A, D_B, \beta_A, \beta_B) \rightarrow (D'_A, D'_B, \beta_A, 1 - \beta_B)$
  Probability: $(1 - P[D_A, a])P[D_B, b]$
  Reward: $C[D_B, b]$
- **None success:** $(D_A, D_B, \beta_A, \beta_B) \rightarrow (D'_A, D'_B, \beta_A, \beta_B)$
  Probability: $(1 - P[D_A, a])(1 - P[D_B, b])$
  Reward: 0

For PoW functions where the best known solving algorithm is trial and error, and negligible network latency, the block inter-arrival time is exponentially distributed. Specifically, Ozisik et al. [74] showed that inter-arrival time T_i for block i has distribution $T \sim \mathsf{Expon}(TD_i/x)$, where T is the target block time, D_i is the difficulty (expected number of hashes per block), and x is the actual number of hashes performed every T seconds. For $X \in \{A, B\}$, the success probability on chain X is therefore given by

$$P[D_X, x] = \int_0^1 f(t)dt,$$

where $f(t)$ is the PDF of the distribution $\mathsf{Expon}(D_X/x)$. That is to say, $P[D_X, x]$ it is given by the cumulative distribution for the exponential from time 0 until 1 second. To derive the expression for block quantity (in excess of 1 block during a 1 second interval) we must first contemplate the time that the first block is mined, t, and then the number of additional blocks that will be mined in the $1 - t$ remaining seconds. The former is governed by an exponential distribution, while the latter follows a Poisson distribution. We have

$$C[D_X, x] \approx 1 + \frac{\int_0^1 f(t) \sum_{i=0}^{10} if'(1-t)dt}{\int_0^1 f(t)dt,},$$

where again $f(t)$ is the PDF of the distribution `Expon`(D_X/x) and $f'(t)$ is the distribution `Poisson`(t). This expression first adds a reward for the first block mined. Next, for each time $t \in [0, 1]$, it calculates the expected quantity of blocks from 0 to 10 total (stopping at 10 because larger values are unlikely), weighted by the probability of mining the first block in time t. Finally, to condition on the event that a first block is found, the expected block count beyond 1 is normalized by the probability of mining a block in the first second. Notice that we use unit difficulty for all blocks other than the first since the DAA is assumed to be adjust perfectly at that point.

C FUTURES LEVERAGING `ORACLE`

EXAMPLE 1: Suppose that we wish to introduce fully decentralized *futures contracts* to blockchain A intended to be negotiated between two parties: guarantor $\mathcal{G}$ and beneficiary $\mathcal{B}$. To do so, a smart contract can be developed that leverages `Oracle`. Each futures contract, or *future* transfers from guarantor to beneficiary a quantity of coins A equivalent to the value of a quantity of coin B at a future date. Specifically, a future issued at the time when chains A and B are at block heights b_A and b_B, allows the beneficiary to trade the contract to the guarantor for a quantity of coins A equivalent to 1 coin B on the *expiry date*. We define expiry as the latter of block heights b'_A and b'_B, anticipated to be some time in the future (for example 90 days). Contract `Future` implements four methods: `Deposit`(a), `Recover`(a), `Issue`$(b_A, b_B, b'_A, b'_B, a)$, and `Redeem`$(b'_A, b'_B)$. `Deposit` is signed by $\mathcal{G}$; it deposits quantity a of coin A into `Future`. This will be used to pay $\mathcal{B}$ at expiry. Prior to calling `Issue`, the funds can be redeemed by $\mathcal{G}$ if he signs `Recover`. The call to `Issue` must be signed by both $\mathcal{G}$ and $\mathcal{B}$; signifying that they agree to the initial and expiry block times and fee of a coins, which is paid by $\mathcal{B}$ and immediately transferred to an account owned by $\mathcal{G}$. Once headers h'_A and h'_B at height b'_A and b'_B have been generated, $\mathcal{B}$ first calls `Update`(h'_B) on `Oracle` and then signs `Redeem`. In response to this method, contract `Future` deposits into an account controlled by $\mathcal{B}$ a quantity of A coins that are equivalent to the value of 1 coin B as determined by calling `Query`(b'_A, b'_B) on contract `Oracle`.

D PRICE ORACLE ALGORITHMS

Algorithm 1: `Oracle.Update`(h_B)

```
if $h_B[P] \neq \mathcal{H}(h'_B)$ then
    return;
end
if $\mathcal{H}(h_B) > h'_B[g]$ then
    return;
end
Headers$_B$.append($h_B$);
```

Algorithm 2: `Oracle.Query`$(b_A, b_B, \sigma_\Delta)$

```
if length(Headers_A) < b_A then
    throw error;
end
if length(Headers_B) < b_B then
    throw error;
end
return $\sigma_\Delta \frac{k_A}{k_B} \frac{\texttt{Headers}_B[b_B][D]}{\texttt{Headers}_A[b_A][D]}$;
```

E PROOFS

THEOREM 1: *The following allocation is a symmetric equilibrium for the Security Allocation Game:*

$$[\boldsymbol{w}_i^*, \boldsymbol{w}_{\text{-}i}^*] = \left[\frac{1}{N}(c, 1-c), \frac{n}{N}(c, 1-c)\right],$$

where $n = N - 1$ *and* $c = \frac{T_B R}{T_B R - T_A R + T_A}$. *When* $T_A = T_B$ *the equilibrium simplifies to* $c = R$.

PROOF: Allocation $[\boldsymbol{w}_i^*, \boldsymbol{w}_{\text{-}i}^*]$ constitutes a Nash equilibrium if every miner's best response at that point is to maintain the same allocation. Because miner resources are assumed to be homogenous, it will suffice to show that $w_{iA} = \frac{c}{N}$ is the best response when $w_{\text{-}iA}^* = \frac{cn}{N}$.

When the allocation for miners $m_{\text{-}i}$ is $w_{\text{-}iA}$, the best response for miner m_i is given by $\boldsymbol{\pi}^T \boldsymbol{w}_i$, which we denote in this proof simply by y_i. Thus, our task is to show that $w_{iA}^* = \frac{c}{N}$ is the global optimum of y_i when $w_{\text{-}iA} = \frac{cn}{N}$. To that end, we proceed by identifying and testing the critical points of function y_i, beginning with its local optima.

Solving $\frac{\partial y_i}{\partial w_{iA}} = 0$ gives all local optima. We have

$$\frac{w_{\text{-}iA}}{(w_{iA} + w_{\text{-}iA})^2} \frac{V_A}{T_A} + \frac{w_{\text{-}iA} - \frac{n}{N}}{(1 - w_{\text{-}iA} - w_{iA})^2} \frac{V_B}{T_B} = 0,$$

which implies

$$\begin{aligned} & \frac{w_{\text{-}iA}}{(w_{iA}+w_{\text{-}iA})^2} \frac{V_A}{T_A} = \frac{\frac{n}{N} - w_{\text{-}iA}}{(1-w_{\text{-}iA}-w_{iA})^2} \frac{V_B}{T_B} \\ \Rightarrow \quad & \frac{\sqrt{w_{\text{-}iA}}}{w_{iA}+w_{\text{-}iA}} \sqrt{\frac{V_A}{T_A}} = \pm \frac{\sqrt{\frac{n}{N} - w_{\text{-}iA}}}{1-w_{\text{-}iA}-w_{iA}} \sqrt{\frac{V_B}{T_B}}. \end{aligned}$$

The quantity on the left is always positive and because $w_{\text{-}iA} < \frac{n}{N}$ and $w_{iA} + w_{\text{-}iA} < 1$, the absolute value of the quantity on the right is also positive. Therefore, only the positive branch of the square root leads to a valid solution. It follows that

$$w_{iA} = \frac{(1 - w_{\text{-}iA})\sqrt{w_{\text{-}iA} V_A T_B} - w_{\text{-}iA}\sqrt{(\frac{n}{N} - w_{\text{-}iA}) V_B T_A}}{\sqrt{w_{\text{-}iA} V_A T_B} + \sqrt{(\frac{n}{N} - w_{\text{-}iA}) V_B T_A}} \tag{14}$$

is the only local optimum. Thus, we proceed by performing the substitution $w_{\text{-}iA}^* = \frac{cn}{N}$ in Eq. 14 and showing that its value is equal to $\frac{c}{N}$. Using $\frac{1-c}{c} = \frac{V_B}{V_A} \frac{T_A}{T_B}$ and $c = \frac{T_B V_A}{T_B V_A + T_A V_B}$, we have (algebra not shown)

$$\begin{aligned} w_{iA} \quad &= \quad \frac{(1-\frac{cn}{N})\sqrt{\frac{cn}{N} V_A T_B} - \frac{cn}{N}\sqrt{(\frac{n}{N} - \frac{cn}{N}) V_B T_A}}{\sqrt{\frac{cn}{N} V_A T_B} + \sqrt{(\frac{n}{N} - \frac{cn}{N}) V_B T_A}} \\ &= \quad \frac{c}{N}. \end{aligned}$$

The payoff to miner i for allocation $c\left(\frac{1}{N}, \frac{n}{N}\right)$ is

$$\begin{aligned} y_{ic} &= \left(\frac{\frac{c}{N}}{\frac{c}{N}+\frac{cn}{N}} \frac{V_A}{T_A} + \frac{\frac{(1-c)}{N}}{\frac{(1-c)}{N}+\frac{(1-c)n}{N}} \frac{V_B}{T_B} \right) \\ &= \frac{1}{N}\left(\frac{V_A}{T_A} + \frac{V_B}{T_B}\right). \end{aligned}$$

Next, we turn our attention to proving that $[\boldsymbol{w}_i^*, \boldsymbol{w}_{\text{-}i}^*]$ is actually a global optimum by showing its payoff, y_{ic}, exceeds that of other critical points of the payoff function. Endpoints $[(0, \frac{c}{N}), (\frac{cn}{N}, 0)]$ and $[(\frac{c}{N}, 0), (\frac{cn}{N}, 0)]$ constitute the remaining critical points. Their payoffs are, respectively,

$$y_{i0} = \frac{1}{N - cn}\frac{V_B}{T_B} \text{ and } y_{i\frac{1}{N}} = \frac{1}{1 + cn}\frac{V_A}{T_A}.$$

It can be shown that $y_{ic} \geq y_{i0}$ and $y_{ic} \geq y_{i\frac{1}{N}}$ for all choices of $\frac{V_A}{T_A}$ and $\frac{V_B}{T_B}$. Therefore, allocation $w_{iA} = \frac{c}{N}$ maximizes payoff when $w_{\text{-}iA} = \frac{cn}{N}$, so allocation $[\boldsymbol{w}_i^*, \boldsymbol{w}_{\text{-}i}^*]$ is a Nash equilibrium.

□

PROPOSITION 1: *For any fixed allocation $\boldsymbol{w}$, after the SAAs on chains A and B come to rest, the portfolio pricing vector will be $\boldsymbol{p} = e\boldsymbol{w}$.*

PROOF: Consider a blockchain X with aggregate claim $c_X(\tau)$ and prevailing price $p_X(\tau) = S_X(\tau)$. Together these two quantities entirely determine the actual security investment applied to the chain:

$$s_X(\tau) = c_X(\tau)p_X(\tau) = c_X(\tau)S_X(\tau). \tag{15}$$

Thus in order for the SAA to be at rest, it must be the case that $c_X(\tau) = 1$. The same reasoning can be applied to any blockchain, so that if both SAAs are at rest at time τ, then $\boldsymbol{c}(\tau)$ is a vector of all ones. Finally, from Eq. 6 we have that

$$\boldsymbol{p}(\tau) = e\boldsymbol{w}(\tau) \oslash \boldsymbol{c}(\tau) = e\boldsymbol{w}(\tau).$$

□

THEOREM 2: *Assume any choice of SAA for chains A and B (not necessarily the same). When both SAAs are at rest and the relative reward R is stable, there exists no arbitrage at the following allocation*

$$\boldsymbol{w}_{\text{eq}} = \left(\frac{T_B R}{T_B R - T_A R + T_A}, \frac{T_A(1-R)}{T_B R - T_A R + T_A}\right),$$

which simplifies to

$$\boldsymbol{w}_{\text{eq}} = (R, 1 - R),$$

if $T_A = T_B$.

PROOF: Suppose that the current allocation is $\boldsymbol{w}_{\text{eq}}$ and both SAAs are at rest. From Proposition 1, we know that the price of claims is given by

$$\boldsymbol{p}_{\text{eq}} = \frac{e}{T_B R - T_A R + T_A}(T_B R, T_A(1 - R)).$$

Now consider the payoff and price associated with the change in claim that manifests the following change in allocation:

$$\Delta\boldsymbol{w} = \frac{1}{T_B R - T_A R + T_A}(\delta_1, -\delta_2),$$

for arbitrary $\delta_1, \delta_2 > 0$. That is to say, the allocation to chain A is boosted proportional to δ_1 while the allocation to chain B is sold short proportional to δ_2. According to Eq. 6, in the moments before either SAA responds to this allocation change, the claim associated with $\Delta \boldsymbol{w}$ becomes $\Delta \boldsymbol{c} = e \Delta \boldsymbol{w} \oslash \boldsymbol{p}_{\text{eq}}$, or

$$\Delta \boldsymbol{c} = \left(\frac{\delta_1}{T_B R}, \frac{-\delta_2}{T_A(1-R)} \right).$$

Therefore, the payoff associated with this change is

$$\Delta \boldsymbol{c}^T \boldsymbol{\pi} = \frac{\delta_1 V_A}{T_A T_B R} - \frac{\delta_2 V_B}{T_A T_B (1-R)} = (\delta_1 - \delta_2) \frac{(V_A + V_B)}{T_A T_B}.$$

And the corresponding price is

$$\Delta \boldsymbol{c}^T \boldsymbol{p} = \frac{e(\delta_1 - \delta_2)}{T_B R - T_A R + T_A} = (\delta_1 - \delta_2) \frac{e(V_A + V_B)}{T_B V_A + T_A V_B}.$$

To prove the theorem, it will suffice to show that *(i)* when $\Delta \boldsymbol{c}^T \boldsymbol{\pi} > 0$, $\Delta \boldsymbol{c}^T \boldsymbol{p} \geq 0$ and *(ii)* when $\Delta \boldsymbol{c}^T \boldsymbol{p} < 0$, $\Delta \boldsymbol{c}^T \boldsymbol{\pi} \leq 0$. To that end, note that in order for $\Delta \boldsymbol{c}^T \boldsymbol{\pi} > 0$, it must be the case that $\delta_1 > \delta_2$. Therefore, $\Delta \boldsymbol{c}^T \boldsymbol{p} > 0$. Conversely, if $\Delta \boldsymbol{c}^T \boldsymbol{p} < 0$, then $\delta_2 > \delta_1$, which implies that $\Delta \boldsymbol{c}^T \boldsymbol{\pi} < 0$.

□

LEMMA 1: *For initial allocation $\boldsymbol{w}$ and price $\boldsymbol{p}$, with SAAs at rest, the claims associated with a symmetric rebalancing $\Delta \boldsymbol{w}$ are given by $\Delta \boldsymbol{c} = \Delta \boldsymbol{w} \oslash \boldsymbol{w}$ and it is always the case that $\Delta \boldsymbol{c}^T \boldsymbol{p} = 0$.*

PROOF: According to Eq. 6, prior to either SAA responding to the allocation rebalancing, the claim associated with $\Delta \boldsymbol{w}$ is given by

$$\Delta \boldsymbol{c} = e \Delta \boldsymbol{w} \oslash \boldsymbol{p}. \tag{16}$$

Meanwhile, Proposition 1 establishes that $\boldsymbol{p} = e\boldsymbol{w}$. Thus, it follows that $\Delta \boldsymbol{c} = \Delta \boldsymbol{w} \oslash \boldsymbol{w}$. Returning to Eq. 16, it is also apparent that $\Delta \boldsymbol{c}^T \boldsymbol{p} = e(\Delta w_A + \Delta w_B)$, which is always zero provided that $\Delta \boldsymbol{w}$ is symmetric.

□

THEOREM 3: *For any allocation $\boldsymbol{w} \neq \boldsymbol{w}_{\text{eq}}$, with SAAs at rest and relative reward R stable, there exists a symmetric allocation rebalancing $\Delta \boldsymbol{w}$, such that $|(\boldsymbol{w} + \Delta \boldsymbol{w}) - \boldsymbol{w}_{\text{eq}}| \leq |\boldsymbol{w} - \boldsymbol{w}_{\text{eq}}|$, which has price zero and strictly positive payoff.*

PROOF: Without loss of generality we may assume that $w_A < w_{\text{eq}A}$, which implies that $\boldsymbol{w} = \boldsymbol{w}_{\text{eq}} - (\delta_1, -\delta_2)$ for δ_1 and δ_2 such that $0 < \delta_1, \delta_2 < 1$. Let $\Delta \boldsymbol{w} = (\epsilon, -\epsilon)$ for some $\epsilon < \min\{\delta_1, \delta_2\}$. Note that, by construction, $|(\boldsymbol{w} + \Delta \boldsymbol{w}) - \boldsymbol{w}_{\text{eq}}| \leq |\boldsymbol{w} - \boldsymbol{w}_{\text{eq}}|$. According to Lemma 1, we have

$$\Delta \boldsymbol{c} = \Delta \boldsymbol{w} \oslash (\boldsymbol{w}_{\text{eq}} - (\delta_1, \delta_2)).$$

Substituting values for $\Delta \boldsymbol{w}$ and $\boldsymbol{w}_{\text{eq}}$ yields

$$\Delta \boldsymbol{c} = \left(\frac{\epsilon(T_B V_A + T_A V_B)}{(1-\delta_1) T_B V_A - \delta_1 T_A V_B}, \frac{-\epsilon(T_B V_A + T_A V_B)}{(1+\delta_2) T_A V_B + \delta_2 T_B V_A} \right).$$

It follows that payoff is given by

$$\Delta \boldsymbol{c}^T \boldsymbol{\pi} = \alpha \left(\frac{V_A}{(1-\delta_1) T_A T_B V_A - \delta_1 T_A^2 V_B} - \frac{V_B}{(1+\delta_2) T_A T_B V_B + \delta_2 T_B^2 V_A} \right),$$

where $\alpha = \epsilon(T_B V_A + T_A V_B)$. Notice that $(1-\delta_1) T_B V_A - \delta_1 T_A V_B = w_{\text{eq}A} - \delta_1 > 0$. Therefore, both terms in the difference above are positive. It follows that payoff $\Delta \boldsymbol{c}^T \boldsymbol{p}$ will be greater than zero provided that

$$-\delta_1 T_A^2 V_B^2 - \delta_2 T_B^2 V_A^2 < (\delta_1 + \delta_2) T_A T_B V_A V_B,$$

which is true for all valid δ_1 and δ_2.

Next, consider the price of rebalancing: $\Delta \boldsymbol{c}^T \boldsymbol{p}$. Since $\Delta \boldsymbol{c} = e \Delta \boldsymbol{w} \oslash \boldsymbol{p}$, it follows that

$$\Delta \boldsymbol{c}^T \boldsymbol{p} = e(\Delta w_A + \Delta w_B) = e(\epsilon - \epsilon) = 0,$$

which implies that the price associated with rebalancing is zero. □

COROLLARY 1: *For allocation $\boldsymbol{w} \neq \boldsymbol{w}_{\mathsf{eq}}$, with SAAs at rest and relative reward R stable, any symmetric rebalancing allocation $\Delta\boldsymbol{w}$ such that $|(\boldsymbol{w} + \Delta\boldsymbol{w}) - \boldsymbol{w}_{\mathsf{eq}}| > |\boldsymbol{w} - \boldsymbol{w}_{\mathsf{eq}}|$ has price zero will result in strictly negative payoff.*

PROOF: Again, without loss of generality, we may assume that $w_A < w_{\mathsf{eq}A}$ and that $\boldsymbol{w} = \boldsymbol{w}_{\mathsf{eq}} - (\delta_1, -\delta_2)$. To ensure that $|(\boldsymbol{w} + \Delta\boldsymbol{w}) - \boldsymbol{w}_{\mathsf{eq}}| > |\boldsymbol{w} - \boldsymbol{w}_{\mathsf{eq}}|$, it must be the case that $\Delta\boldsymbol{w} = (-\epsilon, \epsilon)$ for some $\epsilon > 0$. Since $\Delta\boldsymbol{w}$ is symmetric, Lemma 1 ensures that it achieves a portfolio price of zero. Following closely to the derivation in Theorem 3, the payoff is given by

$$\Delta\boldsymbol{c}^T\boldsymbol{\pi} = \left(\frac{-\epsilon V_A(T_B V_A + T_A V_B)}{(1-\delta_1)T_A T_B V_A - \delta_1 T_A^2 V_B} + \frac{\epsilon V_B(T_B V_A + T_A V_B)}{(1+\delta_2)T_A T_B V_B + \delta_2 T_B^2 V_A}\right),$$

which can never be positive. □

COROLLARY 2: *Let relative reward ratio R be given. If the security allocation is equal to $\boldsymbol{w}_{\mathsf{eq}}$ and SAAs come to rest, then the SAAs will remain at rest so long as R remains fixed and rogue miners do not change their allocation.*

PROOF: According to Theorem 2, for a given reward ratio R, there exists no arbitrage opportunity at allocation $\boldsymbol{w}_{\mathsf{eq}}$ provided that SAAs are at rest. Now suppose that the security investment allocation is equal to $\boldsymbol{w}_{\mathsf{eq}}$ when the reward ratio is R and SAAs are at rest. The equilibrium allocation itself will not change so long as the reward ratio remains fixed. By assumption, all rogue miners will maintain their allocation. The remaining miners all seek to exploit arbitrage. And because the allocation is at equilibrium, they have no incentive to change their allocation either. Therefore, SAAs will remain at rest so long as the reward ratio remains stable. □

THEOREM 4: *If $w_X > 0$ for $X \in \{A, B\}$ and the fraction of security investment devoted to exploiting arbitrage exceeds* $\max\{w_{\mathsf{eq}A}, w_{\mathsf{eq}B}\}$*, then allocation vector $\boldsymbol{w}$ will tend toward $\boldsymbol{w}_{\mathsf{eq}}$ whenever both SAAs are at rest and relative reward R is stable.*

PROOF: Collectively, the assumptions that both SAAs are at rest and relative reward R is stable, along with Theorems 2 and 3, and Corollary 1, imply that: *(i)* there exists a single security allocation $\boldsymbol{w}_{\mathsf{eq}}$ that achieves no arbitrage; *(ii)* at every other allocation, it is possible to exploit arbitrage by rebalancing in the direction of $\boldsymbol{w}_{\mathsf{eq}}$; and *(iii)* as long as miners maintain constant security (i.e., hash rate) across chains, *every* rebalancing that exploits arbitrage will move the allocation in the direction of $\boldsymbol{w}_{\mathsf{eq}}$. Thus, at any allocation $\boldsymbol{w} \neq \boldsymbol{w}_{\mathsf{eq}}$, a miner seeking to exploit arbitrage will move the allocation close to equilibrium. And because we assume that fraction $\max\{w_{\mathsf{eq}A}, w_{\mathsf{eq}B}\}$ of security investment seeks arbitrage, there cannot exist sufficient rogue security investment on either chain to keep the overall allocation out of equilibrium. □