Trust at Scale: The Economic Limits of Cryptocurrencies and Blockchains*†

Eric Budish‡

September 24, 2024

Abstract

Satoshi Nakamoto (2008) invented a new kind of economic system that does not need the support of government or rule of law. Trust and security instead arise from a combination of cryptography and economic incentives, all in a completely anonymous and decentralized system. This paper uses a simple three-equation argument to show that this new form of trust, while ingenious, is deeply economically limited—at least in its pure form without implicit support from rule of law. A zero-profits condition on the amount of "trust support" devoted to blockchain-based trust (work, stake, etc.) and an incentive compatibility condition on the system's security against majority attack (the Achilles' heel of all forms of permissionless consensus), together imply an equilibrium constraint that the recurring, "flow" cost of trust must be large relative to the value that can be gained in a majority attack. This is very expensive in absolute terms relative to the value of attack and moreover scales linearly with the value of attack. In some scenarios the cost exceeds global GDP. The economics of this form of trust would become much more attractive if an attacker were to lose the stock value of their capital in addition to paying the flow cost of attack, but this requires either external punishment or risk that the cryptocurrency collapses. The key contrast between Nakamoto trust and traditional trust grounded in rule of law is the economies of scale that arise from credible deterrence, as in Hayek (1960) and Becker (1968): Society or a firm pays a fixed cost to enjoy trust over a large quantity of economic activity at low or zero marginal cost.

*This paper originally circulated in June 2018 in a shorter form as Budish (2018).

†Acknowledgments: I am grateful to the editor Andrei Shleifer, the co-editor Stefanie Stantcheva and six anonymous referees for their valuable advice. Thanks are also due to Susan Athey, Vitalik Buterin, Glenn Ellison, Gene Fama, Alex Frankel, Joshua Gans, Matt Gentzkow, Edward Glaeser, Austan Goolsbee, Hanna Halaburda, Zhiguo He, Joi Ito, Steve Kaplan, Anil Kashyap, Judd Kessler, Scott Kominers, Randy Kroszner, Robin Lee, Jacob Leshno, Andrew Lewis-Pye, Shengwu Li, Jens Ludwig, Neale Mahoney, Gregor Matvos, Paul Milgrom, Sendhil Mullainathan, Vipin Narang, Neha Narula, Ariel Pakes, David Parkes, Al Roth, Tim Roughgarden, John Shim, Scott Stornetta, Adi Sunderam, Chad Syverson, Alex Tabarrok, Nusret Tas, David Tse, Rakesh Vohra and numerous seminar audiences. Ethan Che, Natalia Drozdoff, Matthew O'Keefe, Anand Shah, Peyman Shahidi, Jia Wan and Tianyi Zhang provided excellent research assistance.

‡University of Chicago Booth School of Business, eric.budish@chicagobooth.edu

1 Introduction

Economists have long widely agreed that the market system requires some form of government and rule of law for support. This is uncontroversial among even the most free-market oriented thinkers. Adam Smith (1776) mostly argues for reducing government interference in markets, but he does not go all the way to zero, writing that "commerce and manufactures can seldom flourish long in any state" without a legal system, property rights and contract enforcement, as well as certain public goods. Hayek (1960) grapples at length with the paradox that to maximize freedom—which he defines as the absence of coercion—it is necessary to have a government that has the power to coerce. Friedman (1962) famously described the government's role establishing the "rules of the game" for the market system and acting as its "umpire." There is significant debate within modern economics about what else government should do beyond these basic supports for the market system (e.g., social insurance, correcting externalities), but that there is some role for government and rule of law is essentially taken for granted.

Satoshi Nakamoto (2008) invented a new kind of economic system that does not need the support of government or laws.[1] Trust and security instead arise from a combination of cryptography and economic incentives, all in a completely anonymous and decentralized system. Computer scientists call the innovation "permissionless consensus": a large, anonymous, freely-entering and -exiting set of participants around the world is incentivized to collectively maintain a common data set, enabling trust in this data set without the need for rule of law or any specific trusted party. This invention enabled cryptocurrencies, including Nakamoto's own creation Bitcoin. The data structure maintained by the large set of participants is called a blockchain.[2]

It is no understatement to say that Nakamoto's invention captured the world's attention. One oft-cited figure is the $3 trillion of market capitalization of Bitcoin and other crypto assets at their peak, but even this figure seems to understate the amount of cultural, political, and commercial attention that has been paid to blockchains and cryptocurrencies. Yet at the same time, the economic usefulness of Nakamoto's invention remains an open question. To date, the majority of cryptocurrency volume appears to be speculative, with the other most widely documented use case being

[1] The anti-government views of Nakamoto and other early Bitcoin enthusiasts have been widely documented. See Popper (2015) for one early account and satoshi.nakamotoinstitute.org as a primary source. A few example quotes from these sources give a sense: "It's completely decentralized with no server or central authority" (Nakamoto email, 1/8/2009); "a new territory of freedom" (Nakamoto email, 11/6/2008); "outside the reach of any government" (Popper, pg 48).

[2] Not widely appreciated is that the blockchain data structure, *without* the novel method of trust, significantly predates Nakamoto, at least in terms of the core scientific ingredients if not popular and commercial appreciation of its potential usefulness (Haber and Stornetta, 1991; Bayer, Haber and Stornetta, 1993). This data structure is sometimes called a permissioned blockchain, and is in essence a well-architected database that is append-only, has clear rules about what parties can add what data, and uses cryptography to prove that past data has not been deleted or tampered with. See Section 2.5.1 and Budish and Sunderam (2023) for further discussion.

black-market transactions (Makarov and Schoar, 2021; Gensler, 2021; Buterin, 2022).[3] Moreover, the majority of this speculative volume has been through cryptocurrency exchanges—which are, at least in principle, centralized, trusted financial intermediaries.

This paper studies the economics of Nakamoto's novel form of trust. Is it economically viable as an alternative to the traditional market system supported by rule of law? What are the fundamental economics of the fundamental computer science innovation in Nakamoto (2008)?

I find that—at least in its pure form, without any implicit protection from rule of law—Nakamoto's novel form of trust faces serious economic limits. It is unusually expensive in absolute terms relative to the stakes involved, and its expense scales linearly with the stakes involved. These results have an if-then implication: if permissionless consensus in its pure form were to become a more important part of the global economic and financial system than it has been to date, then the costs of securing the trust would become preposterous—more than all of global GDP in some scenarios. The analysis may also sharpen our conceptual understanding of what is special about traditional forms of trust that are grounded in rule of law and other complementary sources such as reputations, relationships and collateral. The key distinction will prove to be economies of scale in the production of trust.

The core of the argument is three simple equations. The first equation is a zero-profits condition that describes the quantity of "trust support" devoted to maintaining permissionless consensus as a function of the compensation paid by the protocol for trust support, assuming that all participants behave honestly. Trust support can take the form of providing computational work as in Bitcoin ("proof of work"), providing other kinds of computational resources as in proof-of-storage or proof-of-memory protocols, or posting cryptocurrency coins in a proof-of-stake protocol such as modern Ethereum.[4] For a sense of magnitudes, in recent years the compensation to Bitcoin trust support

[3]Makarov and Schoar (2021) find that about 75% of Bitcoin transaction volume since 2015 involves cryptocurrency exchanges or exchange-like entitites, once the data are cleaned to account for spurious volume (such as a user moving their own funds from one address to another). They conclude that "the vast majority of Bitcoin transactions between real entities are for trading and speculative purposes." In a dataset from an earlier time period and using a different data cleaning and classification methodology, Foley, Karlsen and Putniņš (2019) find that 46% of Bitcoin transactions that do not involve cryptocurrency exchanges relate to illegal activity. Many prominent public officials, such as Treasury Secretary Janet Yellen and SEC Chair Gary Gensler, have also described cryptocurrency activity to date as mostly speculative or black-market (Cox, 2021; Gensler, 2021). Ethereum founder Vitalik Buterin wrote in a Dec 2022 essay that he is most excited about applications still to come in the future, not the ones that already exist which he describes as "hyperfinancialized" (Buterin, 2022).

[4]See Lewis-Pye and Roughgarden (2024) for an overview of the computer science literature on permissionless consensus including description of all of the key protocols and forms of trust support. Throughout the paper, when I use the phrase "Nakamoto trust" or "blockchain-based trust" I am referring to Nakamoto's overall idea of blockchain-based permissionless consensus, and I try to be clear when I am referring more specifically to Nakamoto (2008)'s specific implementation of permissionless consensus using longest-chain proof-of-work. In particular, the quantification analysis in Section 4 is specific to Nakamoto's longest-chain proof-of-work protocol, and the key economic security difference between proof-of-work and proof-of-stake protocols is analyzed in detail in Sections 5.1-5.2 and discussed later in the introduction.

(known as "miners") has averaged about $250,000 per block of data, or about $36M per day, and Bitcoin miners performed an average of about 200 million trillion calculations per second as an equilibrium response to this compensation. The computer science details behind this process are complicated (see Section 2), and vary to some extent by blockchain protocol, but the economics is standard free-entry logic. Variations of this first equation have appeared in numerous other prior papers.

The second equation is an incentive compatibility condition: how much security does a given level of trust support produce? The Achilles' heel of permissionless consensus is that it is vulnerable to "majority attack": Nakamoto (2008) and subsequent methods for creating an anonymous, decentralized consensus about the state of a dataset rely on a majority of the resources devoted to maintaining the dataset to behave honestly. This is not an obscure point; it is in the abstract of the famous Nakamoto (2008) paper and has been understood to be an issue more generally with distributed consensus systems since famous computer science research in the 1980s. Intuitively, permissionless consensus, whether in the form invented by Nakamoto or other subsequent variations such as proof-of-stake, always relies on some implicit version of majority or super-majority voting to adjudicate what the state is in case there is a dispute. The second equation captures that it must not be economically profitable for a potential attacker to provide a majority of the total trust support and manipulate the state.

It is worth briefly contrasting the approach to security taken in my equation (2) with the approach taken in the prior computer science literature. Famous early papers on distributed consensus, such as Pease et al. (1980), Lamport et al. (1982) and Dwork et al. (1988), stated results of the following form: as long as less than proportion ρ of the servers maintaining consensus are faulty ("Byzantine"), then the consensus protocol has good properties of safety (transactions, once confirmed, will not get reverted) and liveness (transactions can get confirmed within a reasonable amount of time). Nakamoto (2008) gives results of the same form—indeed, with an improvement from $\rho = \frac{1}{3}$ to $\rho = \frac{1}{2}$. However, the early computer science literature was studying what are now called *permissioned* consensus systems, e.g., a company keeping its database servers in synch with each other. Treating security as a 0-1 property that holds if and only if an attacker is less than some bound ρ seems reasonable for permissioned consensus—e.g., it would tell a company to use enough redundant hardware and enough overlapping security measures that the risk of proportion ρ of its servers failing at the same time is sufficiently small. In contrast, Nakamoto (2008) and subsequent blockchain researchers are studying *permissionless* consensus protocols, in which anybody can freely enter and exit the system at any time, anonymously, without any protection from rule of law. As soon as the system is permissionless, one has to ask what is an *attacker's incentive* to acquire proportion ρ of the total system resources. This rethink of the *economic* security of a

permissionless consensus protocol that I introduce in this paper applies to proof-of-work, proof-of-stake, or any other approach to permissionless consensus. They are all vulnerable to some form of majority attack, so they all need to be studied with both a free-entry condition like my equation (1) and an incentive-compatibility condition like my equation (2). In effect, my paper argues that Nakamoto (2008) made an error conceptualizing security in the same way as the 1980's consensus literature and not as an economic incentive compatibility constraint.[5]

My third equation is an equilibrium constraint that connects equations (1) and (2), i.e., connects the zero-profits condition to the incentive compatibility condition. The reason these two equations can be linked is that the amount of honest trust support appears in both. In equation (1), the amount of honest trust support reflects the recurring payments to this trust support. In equation (2), the amount of honest trust support determines the cost of majority attack. Equation (3) then tells us that the recurring payments to the honest trust support in the zero-profit equilibrium must be large relative to the value of attacking the system.

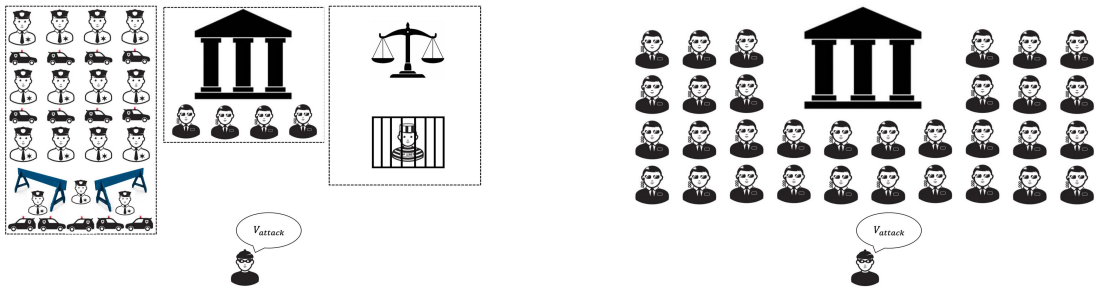
This is a very expensive form of trust! The recurring payments to trust support are a "flow", so equation (3) tells us that the flow costs of maintaining the trust must exceed the value of attack at all times.[6] Moreover, this required flow cost scales linearly with the value of attack. An intuition is that Nakamoto trust is memoryless—it is as secure at a moment in time as the amount of trust support at that moment in time. Under idealized attack circumstances (assuming away several reasonable frictions) I obtain an even stronger result, which is that the net cost of attack is zero—because the attacker also earns the trust-support compensation that would have gone to the honest participants.

The essential difference between the Nakamoto trust model and the traditional model grounded in rule of law is depicted in Figure 1. A criminal is thinking of robbing a bank. In the traditional model, the criminal must first consider how many security guards he will need to overcome. Then he will have to take into account that the bank will call in reinforcements from police, and that, if caught, he will go to jail. Similarly, consider a country thinking of whether to invade another country. They will have to consider the soldiers at the border (analog of security guards), but also that the invaded country will call in military reinforcements (analog of police), and that the invaded country may launch a counter-attack in retaliation (analog of Beckerian deterrence from

[5]Some permissionless consensus protocols are also vulnerable to forms of dishonest play by small participants, typically to obtain a disproportionate share of block rewards as opposed to stealing large sums of money. See Eyal and Sirer (2014), Carlsten et al. (2016), Biais et al. (2019) and Saleh (2021) for well known studies of this issue and of the conditions under which honest play is a Nash equilibrium for small participants.

[6]A 2016 blog post by Ethereum founder Vitalik Buterin contains an informal statement of this insight: "The size of the mining network has to be so large that attacks are inconceivable. Attackers of size less than X are discouraged from appearing by having the network constantly spend X every single day. I reject this logic because (i) it kills trees, and (ii) it fails to realize the cypherpunk spirit." See Section 3.3 for the full quotation and how it can be translated into this paper's model.

Figure 1: Comparison of Trust Models: Nakamoto versus Traditional



Traditional Trust	Nakamoto Trust
cost of overcoming guards + cost of overcoming police reinforcements + risk × punishment if caught $> V_{attack}$	cost of overcoming guards $> V_{attack}$

courts).[7] In contrast, the Nakamoto model is just to have a very large number of security guards at the bank, or soldiers at the border. This works, but it is very expensive and scales terribly with the stakes.

Notice two sources of scale economies in the traditional model. First, the police do not have to be present at the particular bank to provide security—they can provide security to a large number of locations at once as long as they are not all attacked simultaneously. Second, the courts can deter crime with just the credible threat to prosecute and imprison—a fixed cost investment in court capacity can deter a large quantity of potential criminal activity. This is the essence of Becker's (1968) model of optimal deterrence, and is central to Hayek's (1960) resolution of the paradox noted above that freedom requires a government with the power to coerce.[8]

Notice as well a subtle additional source of inefficiency in the Nakamoto trust model—the full scale of the trust support is present for all transactions, whether for large sums or small. This is like having the same number of security guards outside the local bank branch as outside Fort Knox or the Federal Reserve.

[7] I thank Edward Glaeser for drawing this connection to military strategy and Vipin Narang for a helpful discussion about the topic.

[8] Hayek's resolution is that a government's credible threat of coercion, in response to violations of clear, predictable, and symmetrically enforced laws, is both (i) not a violation of freedom, and (ii) sufficient to secure freedom. "The *threat* of coercion has a very different effect from that of *actual and unavoidable* coercion ... The great majority of the threats to coercion that a free society must employ are of this avoidable kind ... The sanctions of the law are designed only to prevent a person from doing certain things or to make him perform obligations that he has voluntarily incurred. ... Provided that I know beforehand ... I need never be coerced." (Pgs 209-210, emphasis added). "It is the cases that never come before the courts, not those that do, that are the measure of the certainty of the rule of law." (Pg. 316). "There is little difference between the knowledge that if he builds a bonfire on the floor of his living room his house will burn down, and the knowledge that if he sets his neighbor's house on fire he will find himself in jail. Like the laws of nature the laws of the state provide fixed features in the environment in which he has to move." (Pg. 221)

There are many other sources of trust that are familiar to economists besides rule of law, such as reputations, brands, relationships, collateral, and cultural or organizational norms.[9] The contrast versus Nakamoto trust is similar: in each of these cases the trust is more secure than the flow level of investment (e.g., a brand is more trustworthy than its advertising expenditure in the last 24 hours). It also bears emphasis that these sources of trust often work in conjunction with rule of law, sometimes implicitly. For example, a customer trusts Starbucks to provide good coffee because of its brand, but also because it is illegal for a different entity to impersonate Starbucks' name and imagery. In a Levin (2003) relational contract, the employee trusts that if they put in high effort they will get paid a performance bonus, but in the background, it is also the case that the employee knows the employer will pay at least the promised minimum because of rule of law, and the employer trusts the employee not to rob the company because of rule of law.[10]

A blockchain would be significantly more economically secure than described so far if both (i) the capital used to maintain the permissionless consensus is specialized, and (ii) an attack causes the attacker's specialized capital to lose its value or be confiscated. The first condition is satisfied for many major blockchains (e.g., Bitcoin ASICs or stake for proof-of-stake protocols). If the second condition obtains as well, then the attacker's cost is not just the flow cost of conducting the attack, but the stock value of their specialized capital—analogously to how a firm that cheats its customers loses the stock value of its reputation or brand. A modified version of the incentive compatibility condition (2) and some simple calculations suggest that the potential economic security improvement is 3-4 orders of magnitude. This difference is large enough to plausibly explain why major blockchains such as Bitcoin and Ethereum have not been majority attacked to date.

However, how exactly does the attacker lose their capital? One possibility is that the value of the cryptocurrency collapses because of the attack, and as a result the value of specialized capital collapses too. But, vulnerability to collapse is itself a serious problem—hardly a reassuring foundation for a novel economic system—and also raises the possibility of an attack motivated by this collapse per se. A second possibility is legal punishment, which certainly would work but which concedes that Nakamoto trust fails without support from rule of law. A third possibility is that

[9]Foundational work on trust from rule of law includes Schelling (1960), Becker (1968), Hart (1995), Shleifer and Vishny (1997), La Porta et al. (1998). Important work on other sources of trust includes Nelson (1974), Kreps et al. (1982), Fudenberg, Levine and Maskin (1994), Tadelis (1999) on brands and reputations; Baker, Gibbons and Murphy (2002), Levin (2003) on relationships; Kandori (1992), Holmstrom and Milgrom (1994), La Porta et al. (1997), Guiso, Sapienza and Zingales (2006) on norms.

[10]Formally, in Levin (2003)'s model, the employer pays the employee at least the fixed salary w_t no matter what, and the employee's lowest action, denoted $e_t = 0$, harms the firm only through poor effort, not theft. If the firm could pay the worker nothing or the worker could rob the firm, the scope for cooperation in the relational contract would be far worse (formally, each party's "renege" option would be much more attractive, undermining the relational contract's ability to be self-enforcing.)

the protocol itself could algorithmically confiscate or block the attacker's capital in a targeted punishment, while leaving honest participants' capital untouched. In a companion computer science paper (Budish, Lewis-Pye and Roughgarden, 2024) we show that this approach (i) is impossible for a broad class of permissionless consensus protocols that includes Bitcoin's; (ii) is possible for a class of permissionless consensus protocols that includes proof-of-stake Ethereum's, but only under a strong assumption about network reliability and only if the attacker is less than proportion $\rho = \frac{2}{3}$ of the total. Hence, the current paper's framework is a lens through which we can understand the economic goals of Ethereum's recent adoption of proof-of-stake and also its economic limits—for attackers of size less than $\rho = \frac{2}{3}$, an attack costs a stock not a flow, but if an attacker is incentivized to exceed $\rho = \frac{2}{3}$ then some external source of trust is required for security, such as rule of law.

The remainder of this paper is organized as follows. Section 2 provides a description of Nakamoto (2008) and related concepts. Section 3 presents the heart of the economic critique of Nakamoto's novel form of trust, equations (1)-(3). Section 4 uses the model to quantify the cost of keeping Nakamoto (2008)'s specific blockchain design secure against double-spending attacks. Section 5 analyzes the case of specialized capital that can lose its value because of the attack. Section 6 contrasts Nakamoto trust with traditional trust grounded in rule of law. Section 7 concludes. Appendix A discusses responses to this paper's argument since it first circulated in 2018. Appendix B provides technical results in support of the double-spending attack analysis. Appendix C compiles lists of cryptocurrency majority attacks, thefts of cryptocurrency financial institutions, and collapses of cryptocurrency financial instutions.

2 Overview of Nakamoto (2008) and Related Concepts

Sections 2.1-2.4 provide an overview of Nakamoto (2008). Some of the detail is specific to Bitcoin and Nakamoto's specific form of permissionless consensus, longest-chain proof-of-work, while also trying to describe the key ideas at a more conceptual level. Section 2.5 discusses three related concepts: permissioned blockchains, smart contracts, and proof-of-stake consensus. The overall goal of this section is to provide an overview of the relevant computer science concepts that is self-contained and at a sufficient level of detail to justify the economics analysis in the rest of the paper.[11] Readers already familiar with this material may skip this section without much loss.

[11]Readers interested in additional computer science detail should consult sources such as the original Nakamoto (2008) paper, Lewis-Pye and Roughgarden (2024)'s recent survey of the computer science literature on permissionless consensus, and Roughgarden's (2023) online course. There are several surveys aimed specifically at economists including Halaburda et al. (2022) and Böhme et al. (2015).

2.1 Transactions

The first step in describing Nakamoto (2008) is to describe transactions, and the limitations of other methods of keeping track of transactions.

Elements of a Transaction. The key elements of a cryptocurrency transaction are the sender of funds, the receiver of funds, the transaction amount, and a cryptographic signature. The sender and receiver are represented as alphanumeric strings called addresses; addresses are somewhat analogous to account numbers. The cryptographic signature uses standard ideas from public-key cryptography to prove that the transaction was initiated by the sender; that is, the signature could only be created by someone who knows the sender's private key for that address. The cryptographic signature also encodes the other transaction details, including the receiver and the transaction amount; it is like not only signing a check but also signing the seal of the envelope that contains the check, so the recipient and amount cannot be subsequently altered.

Limitations of a Shared Public Spreadsheet of Transactions. Imagine keeping track of such transactions on a shared public spreadsheet, such as a Google Doc. The cryptographic signature provides a certain level of trust in the data, in that only Alice, or someone in possession of Alice's private key, can add correctly-signed transactions in which Alice is the sender of funds. However, there are three vulnerabilities:

1. Alice could add a transaction in which she sends money she does not have.
2. Alice could add multiple transactions at the same or similar time, in which she sends money she does have but to multiple parties at the same time.
3. Alice could delete previous transactions from the shared public spreadsheet; either her own or others'.

Thus, while a shared public spreadsheet of transactions could be utilized among parties that trust each other—e.g., a modern version of the babysitting co-op parable in Krugman (1998)—this system is not suitable for tracking transactions among parties that do not have such a level of trust.

Limitations of a Trusted Party. Imagine keeping track of transactions through a widely trusted party that keeps track of balances, such as a central bank (e.g., a Central Bank Digital Currency or CBDC). This approach addresses the three vulnerabilities described above with respect to the shared public spreadsheet: the trusted party can ensure that only valid transactions are added to the ledger and that previous transactions are not deleted. However, the limitation

is that it requires such a trusted party. The central goal of Nakamoto (2008) is to have a trusted ledger of transactions that does not require any specific trusted party.

2.2 What is the Nakamoto Blockchain?

This section describes the Nakamoto (2008) blockchain (i.e., proof-of-work longest-chain consensus) in four steps.

I: Pending Transactions List. Users submit transactions to a pending transactions list, called the mempool. One can think of the mempool as in essence the shared public spreadsheet discussed above. However, transactions in the mempool are not considered official yet.

II: Valid Blocks. Any computer around the world can compete for the right to add transactions from the mempool to a data structure called the *blockchain*. The computational competition will be described in the next step.

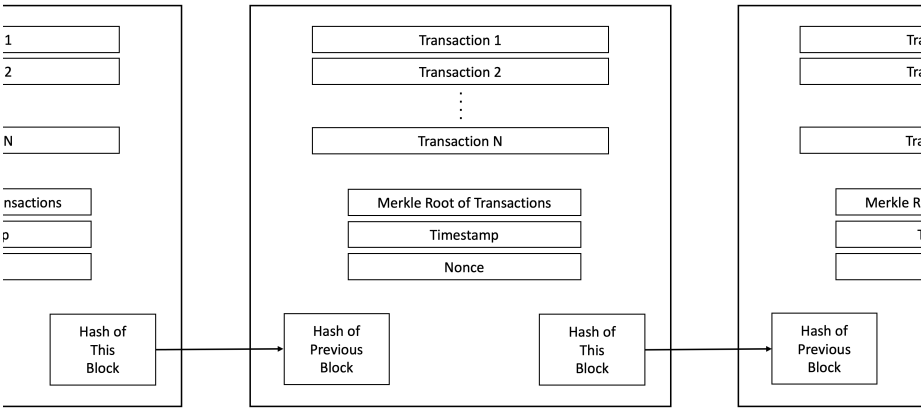
The phrase blockchain references that transactions are added in blocks (for Bitcoin, consisting of about 1000-2000 transactions), and each block of transactions "chains" to the previous block by including a hash of the data in the previous block. See Figure 2. This use of hashes to chain together a sequence of blocks of data was invented by Haber and Stornetta (1991) and Bayer, Haber and Stornetta (1993). Since the hash of the current block depends on the data in the previous block, which in turn includes its hash of the block before that, etc., any change to any element in the history of transactions affects the value of the hash of the current block.

For a block of transactions to be valid, the following three criteria must all be true:

1. Each individual transaction must be properly signed: the cryptographic signature could only be generated by a user in possession of the sender's private key.
2. Each individual transaction must be properly funded: given all transactions in previous blocks in the chain, the sender must be in possession of the cryptocurrency she or he is sending.
3. The transactions in a block must not contradict each other: there cannot be two or more transactions in a block in which a common sender sends the same cryptocurrency to multiple receivers.

III: Bitcoin Mining Computational Tournament. In Nakamoto (2008), the competition to add new blocks boils down to a massive, brute-force search for a lucky random alphanumeric string. More precisely, Bitcoin miners—where "miners" is just the terminology for computational power that attempts to add new blocks of transactions to the Bitcoin blockchain—choose a valid

Figure 2: Illustration of the Blockchain Data Structure



Notes: See the text of Section 2.1 for a description of transactions and the text of Section 2.2 for a description of the overall blockchain data structure and the other elements in the diagram.

block of Bitcoin transactions from the mempool that they wish to chain to the previous block of transactions, and search for an alphanumeric string (called a nonce) such that when that alphanumeric string, in combination with all of the other data in the block they are trying to add, is all hashed together using the hash function SHA-256, the result has a very large number of leading zeros.

For readers unfamiliar with hash functions, it is highly recommended to go to a website like https://www.movable-type.co.uk/scripts/sha256.html to get a feel for how they work. For example, the hash of the title of this paper is 09b23bf1eb4b7cda... which has one leading zero. A block added to the Bitcoin blockchain in January 2024, block 824601, has the hash

0000000000000000000237dcb2a4e6ebb21b499cd81a1ec94b49053c8636be34

which has 19 leading zeros. Since each digit in the hash can take on values 0-9 and a-f, and the SHA-256 hash function is pseudorandom, the likelihood of finding an alphanumeric string that produces a hash with 19 leading zeros is 1 out of 16^{19}, which is about 1 out of 75 billion trillion. The number of leading zeros required is calibrated by the Bitcoin system every roughly two weeks, based on the current amount of computational power devoted to Bitcoin mining, to ensure that blocks are successfully mined on average every 10 minutes. (This calibration can be finer than is possible using just zeros, e.g., 19 leading zeros and a 20th digit weakly less than 7.)

When a miner finds a lucky alphanumeric string, they publicly broadcast their block to all of the other Bitcoin miners. Other Bitcoin miners can quickly check whether the block is valid; that is, does the set of transactions in the block meet the criteria listed above in Step II, and does the alphanumeric string indeed produce a valid hash with enough leading zeros. Critically, while

finding a lucky alphanumeric string is extremely computationally intensive, checking the validity of a given block is computationally trivial. For this reason, a valid block is "proof-of-work"—proof that the miner who found the block did a large amount of computational work in expectation.

The lucky miner who broadcast the valid block gets compensated in two ways. First, the miner is compensated with new Bitcoins. This is called the "block reward", which was originally 50 Bitcoins per block, and halves every roughly four years, most recently in April 2024 to 3.125 Bitcoins per block. Second, the miner earns transactions fees associated with the transactions they included in their block. The economics of these transactions fees are considered in depth in Huberman, Leshno and Moallemi (2021); users who place a high value on getting their transaction added to the blockchain quickly can ensure faster service by offering a larger transaction fee, so there is an auction-theoretic flavor to the fees, as well as queueing and congestion issues.

IV: Longest-Chain Rule. Once a valid block is broadcast and the other miners have checked its validity, miners are supposed to move on to mining the next block. To induce this behavior, Nakamoto (2008) proposed the *longest-chain rule*—the rule that, if there are multiple chains of blocks, the longest chain, as measured by the amount of computational work, is the official consensus record of transactions.

Intuitively, Nakamoto's longest-chain rule provides a decentralized way to coordinate miners' efforts. If miners focus their attention on the current longest chain, and they find a lucky alphanumeric string and mine a block, then their new block will be part of the new longest chain, and hence new official record, and the miner will earn the block reward. Nakamoto (2008) shows formally that as long as a majority of computational power follows the longest-chain rule, then the longest chain will outpace attackers with probability that converges to one exponentially in the honest-majority's share and the deficit the attacker must overcome.

A related intuition is that the longest-chain rule provides a decentralized way to adjudicate disputes—computational power "votes" on the true state, and the majority rules.

The game-theoretic validity of longest-chain consensus has received considerable academic attention. The most general treatment to date is Biais et al. (2019), who show that honest mining on the longest chain is indeed a Nash equilibrium, though there can be other equilibria as well. Carlsten et al. (2016) show that longest-chain consensus is an equilibrium only if the block reward component of miner compensation is large enough. Kroll, Davey and Felten (2013) provide credible intuition for why longest-chain consensus is a Nash equilibrium, though without a formal game-theoretic model.

However, these analyses explicitly assume that all miners are "small"—that is, they assume away the possibility of majority attack. Majority attack, discussed next, will be at the heart of

this paper's analysis.

2.3 Vulnerability to Majority Attack

Nakamoto's blockchain is vulnerable to attack by an adversary with 51% or more of the computational power. This is because the adversary, whenever they like, can create an alternative chain of blocks that will outpace the honest chain of blocks with probability one, and hence become the new consensus. This vulnerability is widely understood—it is even in the abstract of the Nakamoto (2008) paper (excerpted below). Moreover, that Nakamoto's blockchain is vulnerable to attack is not surprising to computer scientists in the sense that previous approaches to distributed consensus, based on the Byzantine Fault Tolerance (BFT) paradigm, were also vulnerable to attack by a too-large adversary except under very restrictive assumptions (Pease et al., 1980; Lamport et al., 1982; Dolev and Strong, 1983; Fischer et al., 1985; Dwork et al., 1988).[12]

The canonical attack Nakamoto (2008) worried about is called a double-spend: the attacker sends Bitcoins in transactions on the original honest chain, and then deletes those transactions from the consensus record with their alternative chain, allowing them to spend the same currency twice. Section 4 will describe double-spending attacks in detail and analyze their economic implications.

Eyal and Sirer (2014) show that Bitcoin is also vulnerable to a form of minority attack, in which a large-enough miner can sometimes profit, in expectation, from holding back a solved block so that they can work on extending it in private, while other miners therefore focus their attention on what is probabilistically not the longest chain. However, the purpose of the Eyal and Sirer (2014) minority attack is more circumscribed in that its goal is to obtain a disproportionate share of mining rewards, not to manipulate the blockchain to double spend. A somewhat analogous issue arises with longest-chain proof-of-stake consensus (Saleh, 2021).

2.4 Nakamoto (2008): Summary

The abstract of Nakamoto (2008) succinctly summarizes the accomplishment and its vulnerability:

> "A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital

[12]The exception in which communication among honest parties is secure even in the presence of an unbounded adversary requires that the honest parties have access to a communication network that never experiences delays longer than a known bound (the "synchronous communications model") and have access in advance of communication to all honest parties' cryptographic public keys (the "public key infrastructure" assumption). These assumptions are frequently satisfied in practical applications with pre-existing trust (e.g., secure military communications) but are widely viewed to be incompatible with the kind of communication Nakamoto (2008) is trying to facilitate over the internet among parties without pre-existing trust. See Roughgarden (2023) for an accessible treatment of key results on BFT consensus.

> signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain serves not only as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. *As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers.* The network itself requires minimal structure. Messages are broadcast on a best effort basis and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone." (Emphasis added)

The accomplishment is a "purely peer-to-peer version of electronic cash" without the use of a "trusted third party." Trust in the integrity of the data emerges from the hash-based proof-of-work, conducted by an unstructured network with free entry and exit. The longest chain is the official record of "what happened"—i.e., is the (permissionless) consensus.

The vulnerability is majority attack—the construction relies on the assumption that "a majority of CPU power is controlled by nodes that are not cooperating to attack the network."

2.5 Clarifications and Discussion

2.5.1 Permissioned Blockchains

As interest in Bitcoin and Nakamoto's blockchain surged, many started to use the phrase "blockchain" to describe similarly-architected databases maintained by *known, trusted parties*—that is, *without* the central scientific innovation of Nakamoto (2008). This concept is sometimes known as a permissioned or private blockchain, or sometimes as distributed ledger technology (see, e.g., Bakos and Halaburda, 2023). An IBM marketing campaign called it "Blockchain for Business." Goldman Sachs called such blockchains "The New Technology of Trust." (Goldman Sachs, 2018)

Many researchers and observers view this use of the phrase "blockchain" as hype for what is in essence just an append-only distributed database with well-defined permissions, and with cryptography to protect against data tampering as in Haber and Stornetta (1991). The financial columnist Matt Levine memorably wrote:

> "If you announce that you are updating the database software used by a consortium of banks to track derivatives trades, the New York Times will not write an article

> about it. If you say that you are blockchaining the blockchain software used by a blockchain of blockchains to blockchain blockchain blockchains, the New York Times will blockchain a blockchain about it." (Levine, 2017)

As should be clear, this paper's critique is of blockchains and permissionless consensus in the sense of Nakamoto (2008), not of distributed databases with trust grounded in traditional sources. It should be uncontroversial that well-architected databases are economically useful, even if there is debate about what to call them. See Budish and Sunderam (2023) for discussion of the potential value of the blockchain data structure in the context of traditional finance.

2.5.2 Smart Contracts

Notice that Nakamoto's novel form of trust is not specific to currency transactions. One can replace "Alice sends Bob 10 Bitcoins, signed by Alice" with any executable computer instruction signed by Alice. This idea is often called a "smart contract" (see Buterin, 2014*a*).

The analysis framework of this paper applies analogously to blockchains that allow smart contracts though the attack possibilities may differ.

2.5.3 Proof-of-Stake

The computational work Bitcoin miners must perform to add a new block serves the role of sybil resistance, i.e., making it expensive to add new identities to the permissionless system. Without sybil resistance an attacker could create infinitely many identities.

Since Nakamoto (2008) there have been several other approaches taken to sybil resistance for permissionless consensus, the most prominent of which is proof-of-stake. Roughly, instead of voting for the correct chain with computational work, participants vote for the correct chain with stake in the cryptocurrency. Ethereum, the second-most valuable cryptocurrency project after Bitcoin, switched from proof-of-work to proof-of-stake in Fall 2022, and its founder has been discussing the potential benefits of proof-of-stake since as early as 2014 (Buterin, 2014*b*, 2016).

One motivation for proof-of-stake over proof-of-work is to reduce environmental externalities. The computational work that powers Bitcoin consumes on the order of 0.3-0.8% of all global electricity, which is a fairly astonishing figure.[13] As will become clear, however, the environmental issue is orthogonal to the economic security concerns raised in this paper about Nakamoto (2008)—whether the cost of trust support is the cost of computational power or the opportunity cost of holding stake the arguments in Section 3 go through unchanged.

[13]De Vries (2018); Digiconomist (2022). The 0.8% figure is Digiconomist (2022)'s main estimate, whereas the 0.3% figure is based on its best-case analysis under the assumption that all Bitcoin mining equipment is maximally energy efficient.

The more important aspect of proof-of-stake for the purpose of this paper's argument is that stakes are not as memoryless as computational work: stakes can be locked up on chain for a period of time, like collateral, and observably persist for the time they are locked, like reputation. This opens up the possibility of punishing attackers by confiscating their locked stakes ("slashing"), which makes attacks more expensive and hence the blockchain more secure. Intuitively, this is an attempt to algorithmically mimic the traditional trust that is created by law in combination with financial collateral (Buterin, 2014*b*, 2016; Buterin and Griffith, 2019). The analysis in Section 5 will show that if slashing works it can improve the cost of security by several orders of magnitude.

However, recent research suggests that this approach to security, while intuitively compelling, faces its own limitations. Tas et al. (2023) show as their Theorem 1 that it is impossible to guarantee that a large-enough attacker can be successfully slashed by any positive amount before the attacker is able to withdraw their stake. Budish, Lewis-Pye and Roughgarden (2024) derive a possibility result for slashing the attacker's stake without punishing honest participants, but the result requires both a strong assumption about the nature of the networking environment and the assumption that the attacker's majority is smaller than $\frac{2}{3}$ of the total locked-up stake. If the attacker is too large then the attacker can circumvent the punishment; a rough intuition is that a large-enough attacker controls the protocol's legal system.[14] An interpretation of these results is that a proof-of-stake blockchain can successfully mimic traditional trust to deter sub-$\frac{2}{3}$ attackers, but that rule of law must step in in the case of a large-enough attacker. See further discussion in Section 5.2.2.

Proof-of-stake also raises a wide variety of implementation complexities which in turn can create protocol-specific opportunities for dishonest play other than the majority attacks that are central to this paper (and universal across all forms of permissionless consensus). See Roughgarden (2023) Lectures 12.1-12.24 for detailed discussion of the potential issues and how they are addressed in various protocols. Of special note, the earliest versions of proof-of-stake used a version of Nakamoto longest-chain consensus and faced a problem called "nothing-at-stake," under which forks might persist indefinitely in Nash equilibrium even with only small participants; see Saleh (2021) for a careful analysis.

[14]The network reliability assumption in Budish, Lewis-Pye and Roughgarden (2024) Theorem 5.1 is that it is possible to impose a delay period between when a user (including honest users) asks to unlock their stake and when they can use it in a transaction, such that the delay period is longer than any feasible attack. Otherwise the attacker could withdraw their stake and spend it before their attack is detected and punished, as in Tas et al. (2023) and the negative results Theorems 4.1-4.2 in Budish, Lewis-Pye and Roughgarden (2024). An implication is that, for a proof-of-stake cryptocurrency to have significant real-world economic utility, only a fraction of the total stake can be locked up for trust support with the rest unlocked for potential use.

3 Nakamoto Trust: A Critique in Three Equations

Sections 3.1-3.3 present the three equation critique of Nakamoto's novel form of trust. Section 3.4 presents a result that shows that the net cost of attack of Nakamoto trust may be *zero* under strong assumptions. Section 3.5 presents a one-shot game version of the analysis that captures the essence of the critique of Nakamoto trust while abstracting from many details. Section 3.6 compares Nakamoto trust to trust from repeated interaction.

3.1 Zero-Profit Condition (Honest Play)

Our conceptual question here is: how much "trust support" will maintain the permissionless consensus if we restrict all participants to behave honestly?

Let there be a large finite number I of honest participants who follow a given permissionless consensus protocol automatically. We may think of I as representing all people who could potentially provide part of the decentralized support for Nakamoto trust. For example, I is the number of people connected to the internet around the world.

Each player i chooses a quantity of "trust support" $x_i \in \mathbb{R}^+$, which we may think of as their quantity of computational work in a proof-of-work blockchain, or their quantity of some other costly action in another blockchain such as stake, storage, memory, etc. Let $N = \sum_{i=1}^{I} x_i$ denote the total quantity of trust support. A player can choose a quantity of zero if they like, which is how we can think about people not participating in the decentralized trust. Our equilibrium concept for N will be a zero-profit condition. This is meant to capture the permissionless, free-entry / free-exit nature of Nakamoto trust. Nash equilibrium is studied in the one-shot game analysis of Section 3.5 and is very similar.

Let c denote the cost per unit time to supply one unit of trust support. For example, for proof-of-work, this is the cost per unit time to run one unit of computational power, including variable costs such as electricity and a rental cost of capital for capital equipment. We will sometimes use the notation $c = rC + \eta$, where rC is the rental cost of capital and η is the variable cost of electricity. For proof-of-stake, c is the opportunity cost per unit time to supply one unit of stake.

Let p_{block} denote the economic reward paid to a participant who successfully adds a new block of transactions, i.e., wins a computational tournament in the case of proof-of-work. We will treat p_{block} as exogenous and derive constraints on it below. Participants' probability of winning the next reward p_{block} is equal to their share of trust support. Specifically, player i wins the next reward with probability $\frac{x_i}{N}$. For the purpose of this paper, we will consider the compensation to providers of trust support in aggregate, without distinguishing between whether this compensation is in the form of newly issued cryptocurrency (which are a form of seignorage tax on holders of

the currency) or transaction fees.

Let D denote the block difficulty level, defined as the number of units of trust-support-time needed, in expectation, to add one new block. For proof-of-work, we may assume that new blocks arrive Poisson. That is, if there are N units of trust support, blocks are solved according to a Poisson point process with mean $\frac{D}{N}$. For some proof-of-stake protocols blocks arrive deterministically at interval $\frac{D}{N}$.[15]

Note a potential source of confusion is that costs c are incurred per unit time whereas rewards p_{block} are earned per block. The next two concepts will help map between objects that are per unit time and objects that are per block. First, we can define profits per unit of trust support per unit time as

$$\frac{1}{N}\frac{D}{N}p_{block} - c,$$

because some unit of trust support solves a block every $\frac{D}{N}$ time in expectation and each of the N units is equally likely to be the winner.

Second, define honest equilibrium as follows:

Definition 1. A *zero-profit honest Nakamoto trust equilibrium* consists of quantities $\{x_i^*\}_{i=1}^I$ and a difficulty level D^* such that participants (i) solve one block per unit time (as a normalization), and (ii) earn zero economic profits in expectation.

Proposition 1. *Let $N^* = \sum_{i=1}^I x_i^*$. In any zero-profit honest Nakamoto trust equilibrium,*

$$N^* c = p_{block}. \tag{1}$$

and $D^* = N^*$.

Proof. For participants to solve one block per unit time (condition (i)) requires $D^* = N^*$. This in turn implies that profits per unit of trust support per unit time are $\frac{1}{N}\frac{D}{N}p_{block} - c = \frac{1}{N^*}p_{block} - c$. For these profits to be zero (condition (ii)) in turn implies $N^* c = p_{block}$. □

Proposition 1 is widely known and is the standard characterization of a rent-seeking tournament: the prize in the tournament, p_{block}, is dissipated by expenditures aimed at winning the prize, $N^* c$.[16] Prat and Walter (2021) provide empirical support that equation (1) describes actual equi-

[15] A difference between Bitcoin longest-chain proof-of-work and Ethereum BFT-style proof-of-stake is that in the latter the protocol directly observes the quantity of trust support N, because the trust support is provided by stake that is locked on chain. In the terminology of Lewis-Pye and Roughgarden (2024) this makes Ethereum a quasi-permissionless protocol.

[16] See, for example: Kroll, Davey and Felten (2013) pg. 8; Huberman, Leshno and Moallemi (2021) Theorem 1; Easley, O'Hara and Basu (2019) equation (2); Chiu and Koeppl (2022) Lemma 1; Ma, Gans and Tourky (2019) equation (7); and Halaburda et al. (2022) equation (4). It is also straightforward to allow for heterogeneous mining

librium behavior in the Bitcoin mining market, with some additional nuances related to capital adjustment costs.

3.2 Incentive Compatibilty Condition (Majority Attack)

Our conceptual question here is: how much security is generated by the amount of honest trust-support characterized in equation (1)? As discussed in Section 2.3, it is widely understood that an agent who provides a majority of the trust support could successfully attack Nakamoto (2008)'s blockchain, e.g., by double spending. This is an issue more generally with all forms of permissionless consensus: intuitively, the decentralized trust support "votes" on the true state of the blockchain and the vote can be manipulated by a majority or super-majority. In this sub-section, we focus on the direct costs of attacking a permissionless consensus protocol. In Section 5 we will consider the possibility that, as a post-attack consequence of the attack, the attacker's trust-support capital loses its value or is confiscated.

Consider an additional player, the attacker, not restricted to honest play. This player can attack by choosing AN^* units of trust support, $A > 1$, for an $\frac{A}{A+1}$ majority at cost AN^*c per unit time. Denote the expected duration of the attack by $t(A)$; the timing details of attacks will vary by protocol and we will derive a closed form for $t(A)$ for longest-chain proof-of-work in Section 4. Call $AN^*c \cdot t(A)$ the gross cost of attack. The attacker can choose A to minimize $A \cdot t(A)$. Call this optimum $A^* \cdot t(A^*)$; Appendix B studies this optimum numerically for longest-chain proof-of-work.

Let V_{attack} denote the value of attack. For now, let us think about this value of attack in the abstract, but have in mind that the value of attack grows as the blockchain's economic usefulness grows.

Definition 2. Nakamoto trust is *incentive compatible against an outsider attack, on a gross-cost basis*, if the gross cost of attack exceeds the benefits of attack:

$$A^*N^*c \cdot t(A^*) > V_{attack}. \tag{2}$$

Two brief remarks are required. First, condition (2) is the IC constraint for an outside attacker, but an attack could also come from an insider, i.e., part of the current honest trust support N^*. The outside attacker IC condition (2) seems more attactive conceptually, because it treats the honest participants as small and dispersed per the Nakamoto ideal and in equilibrium with the level of compensation p_{block}. That said, an inside attacker might be more realistic in practice, e.g.,

costs. Let $c(\cdot)$ denote a continuous weakly increasing function where $c(n)$ gives the per-block cost of the nth unit of computational power. Then (1) becomes $N^* c(N^*) = p_{block}$. The marginal unit of computational power earns zero economic profits.

for Bitcoin there is evidence that mining is concentrated (Makarov and Schoar, 2021; Cong, He and Li, 2021). Second, the left-hand-side of (2) is the gross cost of attack. However, the attacker would earn rewards for the blocks in their new chain, which subsidizes their attack. We will come back to the distinction between gross and net costs of attack in Section 3.4.

3.3 Equilibrium Constraint

In the hoped-for equilibrium in which participants are honest, the amount of trust support devoted to maintaining the blockchain is characterized by the zero-profit equilibrium (1).[17] The incentive-compatibility condition (2) then relates this amount of trust support to the level of security generated. Since N^*c appears in both the zero-profit condition (1) and the incentive-compatibility condition (2), we can combine the two equations:

Theorem 1. *(Equilibrium Constraint) The zero-profit honest Nakamoto trust equilibrium condition (1) and the incentive-compatibility against outsider attack condition (2) together imply the equilibrium constraint:*

$$p_{block} > \frac{V_{attack}}{A^* \cdot t(A^*)} \tag{3}$$

In words: the equilibrium per-block payment to participants for providing trust support on the blockchain must be large relative to the benefits of attacking the blockchain.

Proof. (3) follows directly from combining (1) and (2). □

Economically, this is a very expensive form of trust. Imagine if a brand were only as trustworthy as its flow investment in advertising, or users of the Visa network had to pay fees to Visa, every ten minutes, that were large relative to the value of a successful attack on the Visa network. Or, imagine that a country were only as secure as its flow expenditure on soldiers at the border.

Another contrast is trust that is supported by rule-of-law. In such cases, the cost of cheating to the cheating party is related not to the direct costs of conducting the crime, but to the costs of potentially getting caught and punished (Becker, 1968). A government able to credibly impose large punishments (the parameter f in Becker's model) can deter large attacks or crimes at comparatively low cost. As emphasized, the ingenious aspect of the Nakamoto (2008) form of trust is that it is completely anonymous and decentralized, without any reliance on rule-of-law, relationships or other traditional sources. But, this aspect also makes the Nakamoto (2008) form of trust much more expensive.

[17] If trust-support providers earn positive economic profits in equilibrium then the quantity N^* characterized by the zero-profit condition in (1) is an upper bound, and (3) obtains as is.

From a computer security perspective, the key thing to note about (3) is that the security of the blockchain is *linear* in the amount of expenditure on trust support. For instance, if attacking the system grows 1000 times more attractive, then the cost of securing the system must grow 1000 times as well for the system to remain secure. In contrast, in many other contexts investments in computer security yield convex returns, such as traditional uses of cryptography—analogously to how a lock on a door increases the security of a house by more than the cost of the lock. It is much more expensive to break modern cryptography than it is to implement it![18] Imagine if the cost of attacking Visa was that you had to have as much computational power as Visa for a few hours.

A blog post by Ethereum founder Vitalik Buterin (2016) deserves credit for an early informal statement of this equilibrium constraint:

> "Because proof of work security can only come from block rewards, and incentives to miners can only come from the risk of them losing their future block rewards, proof of work necessarily operates on a logic of massive power incentivized into existence by massive rewards. Recovery from attacks in PoW is very hard: the first time it happens, you can hard fork to change the PoW and thereby render the attacker's ASICs useless, but the second time you no longer have that option, and so the attacker can attack again and again. Hence, the size of the mining network has to be so large that attacks are inconceivable. *Attackers of size less than X are discouraged from appearing by having the network constantly spend X every single day.* I reject this logic because (i) it kills trees, and (ii) it fails to realize the cypherpunk spirit – cost of attack and cost of defense are at a 1:1 ratio, so there is no defender's advantage." (Emphasis added).

Buterin's language can be translated into the formal analysis of this paper by dollar-izing both uses of X: "Attackers of size less than X" can be interpreted as attackers with an attack opportunity worth less than V_{attack} (not as the size of the attacker's computational power, else the two X's are not comparable). "Having the network constantly spend X every single day" can be interpreted as assuming that an attack takes one day, i.e., $A^* \cdot t(A^*)$ is one day worth of block-compute costs, and then requiring for security that $A^* \cdot t(A^*) \cdot p_{block} > V_{attack}$, i.e., equation (3).

3.4 Zero Net Attack Cost Theorem

What we may call the *net* cost of attack can differ from the gross cost of attack, modeled above, for three potential reasons: block rewards, attacker cost frictions, and effects of the attack on the

[18] For example, to break an SHA-256 encrypted data set through brute force would require $2^{256} \approx 10^{77}$ calculations. I estimate that if you had a trillion Amazon Web Services' worth of compute power (about $65 billion trillion of capital), running for the age of the universe (14 billion years), that would get you to about 10^{45} hashes.

value of the cryptocurrency itself.

First, the attacker earns block rewards from their attack. That is, after the attacker's alternative chain replaces the honest chain, the attacker earns the block rewards associated with the blocks in the new longest chain. These block rewards in effect subsidize the attack. An A attacker who attacks for t time performs $At \cdot N^*$ units of trust support. If the difficulty stays constant at $D' = D^* = N^*$, then this corresponds to At block rewards in expectation. If the difficulty on the attacker chain adjusts upwards, i.e., $D' > D^*$, then the attacker will earn $At \cdot \frac{N^*}{D'} < At$ block rewards.

Second, the attacker may face frictions relative to the cost of honest trust support. For example, in proof-of-work, the attacker's compute power might be less energy efficient than the honest miners' compute power. Or, in either proof-of-work or proof-of-stake, there might be frictions associated with starting and stopping the attack. Let $\kappa \geq 0$ parameterize the attacker's cost inefficiency relative to honest mining, such that their total cost of attack is $(1 + \kappa)At \cdot N^*c$.

Third, the attack may harm the value of the cryptocurrency. This reduces the value of the attacker's block rewards and reduces the value of the cryptocurrency the attacker is left with after double spending. If we let $\Delta_{attack} \geq 0$ parameterize this decline, this reduces the value of the attacker's block rewards by $\Delta_{attack}At \cdot N^*c$ and reduces the benefit of a double spending attack originally worth V_{attack} by $\Delta_{attack}V_{attack}$. If the attacker's capital is specific to the attacked cryptocurrency (e.g., ASICs, stake), then the attack would reduce the value of the attacker's capital as well; we will return to this issue in Section 5.

In the ideal case for the attacker with respect to these three sources of cost difference, we have the following remarkable conclusion:

Theorem 2. *(Zero Net Attack Cost) If the attacker does not face any cost frictions relative to the costs of honest participants ($\kappa = 0$), the attack concludes without any difficulty adjustment ($D' = D^*$), and the attack does not cause the value of the cryptocurrency to fall ($\Delta_{attack} = 0$), then the net cost of attack is zero.*

Proof. The attacker's trust-support cost of attack is $(1+\kappa)At \cdot N^*c$. The net value of the attacker's block rewards is $At \cdot \frac{N^*}{D'}p_{block}(1 - \Delta_{attack})$. The reduction in the value of the cryptocurrency the attacker is left with after double spending is $\Delta_{attack}V_{attack}$. If $\kappa = \Delta_{attack} = 0$ and $D' = D^*$, then substituting $N^*c = p_{block}$ and $D^* = N^*$ from Proposition 1 yields a net cost of $At \cdot N^*c - At \cdot p_{block} = 0$. □

The intuition behind this result is that the attacker is fully compensated for their trust-support costs for the same reason that honest participants are fully compensated for their trust-support

costs under honest play. In effect, permissionless consensus treats the attacker "as if" they are an honest participant, because the majority determines the truth.

Moroz et al. (2020), Auer (2019), Tabarrok (2019) and Jacob Leshno (in a communication with the author) derive similar results to Theorem 2 building off of Budish (2018).[19] Bonneau (2016)'s analysis of "bribery" attacks deserves early credit for the intuition that the net cost of attacking Bitcoin might be very small because of the block rewards subsidy, as does a 2017 blog post of Buterin (2017) who stated the idea in a footnote. Recent work of Gans and Halaburda (2023) generalizes the zero net attack cost result and, by incorporating transaction fees into their model, shows that it is possible for an inside attacker to have a slighty negative net attack cost.

To be clear, zero attack frictions seems unrealistic. But, zero friction is often useful as a benchmark case, and the result does reinforce that Nakamoto trust is economically implausible when taken literally.

3.5 One-Shot Game Analysis

The analysis above uses a price-theoretic zero-profit equilibrium concept for honest play, and contains several details that are helpful for mapping to the specifics of Nakamoto (2008) but are complex. As a complement to that approach consider the following stylized one-shot game, which yields a Nash equilibrium solution and abstracts from some of the complexities above.[20]

There are I players. Each player i chooses a quantity x_i of trust support (work, stake, etc.) and a "posture" $a_i \in \{Honest, Attack\}$. Costs are c per unit of trust support. Define $N = \sum_{i=1}^{I} x_i$. Payoffs are as follows. If there is a player i with $x_i > \frac{N}{2}$ and $a_i = Attack$, this player gets a payoff of V_{attack}, gross of their costs. All other players get zero. Else, each player gets a payoff of $\frac{x_i}{N}p$.

Our question is: under what conditions does there exist a Nash equilibrium, denoted $\{(a_i^*, x_i^*)\}_{i=I}^{I}$, in which all players choose $a_i^* = Honest$? Call such a profile, if one exists, an honest equilibrium.

Lemma 1. *If there is an honest equilibrium, then $N^*c \leq p$. (Analog of (1))*

Proof. Towards a contradiction, assume there is an honest equilibrium with $N^*c > p$. Choose any player i with $x_i^* > 0$. Player i's net payoff is $\frac{x_i^*}{N^*}p - x_i^*c < x_i^*c - x_i^*c = 0$. So the player has a profitable deviation by choosing $x_i' = 0$ instead. Contradiction. □

In words, this lemma tells us that the amount spent on trust support N^*c will be no greater than the compensation paid for this trust support p, analogously to equation (1) above.

[19] The analysis in the June 2018 draft artificially constrained the attacker to earn at most t block rewards. The June 2018 draft also did not have explicit cost frictions. Rather, the assumption that an attacker earns at most t block rewards is like an implicit cost friction, related to starting and stopping the attack, of $(A-1)t \cdot N^*c$. As a result, the June 2018 draft had slightly different simulated net costs than here and did not have Theorem 2.

[20] I am grateful to Rakesh Vohra for suggesting this one-shot game approach.

Lemma 2. *If there is an honest equilibrium with $x_i^* = 0$ for some player i, then $N^*c \geq V_{attack}$. (Analog of (2))*

Proof. Assume there is an honest equilibrium with $x_i^* = 0$. Lemma 1 implies $N^*c \leq p$. Consider the possible deviation by player i to choose quantity $x_i' = N^* + \epsilon$ for $\epsilon > 0$ and posture $a_i = Attack$. Since $x_i' > N^*$ player $i's$ attack succeeds. Player i's net payoff from the deviation x_i' is thus $V_{attack} - (N^*c + \epsilon c)$. Player i's net payoff from the conjectured honest equilibrium is 0. Hence to avoid contradiction we need $V_{attack} - (N^*c + \epsilon c) < 0 \rightarrow N^*c > V_{attack} + \epsilon c$. Taking the limit as $\epsilon \rightarrow 0$ yields the desired result. □

Proposition 2. *A necessary condition for an honest equilibrium is $p \geq \frac{V_{attack}}{1+\frac{1}{I}}$. In the limit as $I \rightarrow \infty$ this is $p \geq V_{attack}$. (Analog of (3))*

Proof. Conjecture an honest equilibrium. Player i's payoff in honest equilibrium is $\frac{x_i^*}{N^*}p - x_i^*c$. Consider a deviation by i in which they attack by choosing $a_i' = Attack$ and $x_i' = N_{j\neq i}^* + \epsilon = \sum_{j\neq i} x_j^* + \epsilon$ for some $\epsilon > 0$. For this to be worse for player i requires $V_{attack} - N^*{}_{j\neq i}c - \epsilon \leq \frac{x_i^*}{N^*}p - x_i^*c$. Rearranging, using Lemma 1, and noting that $\min(x_i^*) \leq \frac{1}{I}N^*$ yields $V_{attack} \leq p(1 + \frac{1}{I})$. □

An interpretation of the timing of this game is that p and c now represent, respectively, blockchain compensation and trust-support-costs for an amount of time commensurate with the duration of an attack, i.e., the analog of $A^* \cdot t(A^*)$ in (3). The result says that the cost of running the blockchain for an attack-duration amount of time must exceed the value of attacking it.

Note that this simple model purposefully restricts attention to honest play or majority attack. Many prior game-theoretic analyses of blockchains assume that all players are small, eliminating the possibility of majority attack, but allow for a richer set of strategies for such small players than just honestly following the protocol (e.g., Eyal and Sirer, 2014; Carlsten et al., 2016; Biais et al., 2019; Saleh, 2021). Since Proposition 2 is a *necessary* condition for the existence of an honest equilibrium, the result still holds even with an enlarged strategy space including behaviors such as selfish mining in Eyal and Sirer (2014) or nothing-at-stake in Saleh (2021).

3.6 Comparison to Trust from Repeated Interaction

Since Schelling (1956) and Aumann (1959), economists have studied trust that is facilitated by repeated mutually-beneficial interaction. Suppose two agents play a repeated prisoner's dilemma game where each participant's per-period payoff from cooperation is τ (for trust), a participant who defects while the other cooperates earns $b > \tau$ (for betray), if both players defect they earn 0, and the discount factor is δ. In the well-known grim-trigger equilibrium, cooperation is possible if

$$b - \tau \leq \frac{\delta}{1-\delta}\tau, \tag{4}$$

that is, if the one-shot benefit of cheating is smaller than the net-present-value of the trusting relationship.

To compare this form of trust from repeated interaction, "Schelling trust", to Nakamoto's permissonless consensus, let us rewrite (4) using $V_{attack} \equiv b$ and $V_{trust} \equiv \frac{1}{1-\delta}\tau$, and consider repeated play of the one-shot game we just analyzed in Section 3.5. We thus have the comparison of incentive compatibility conditions:

$$\begin{aligned} Schelling\ Trust &: V_{trust} \geq V_{attack} \\ Nakamoto\ Trust &: p \geq V_{attack} \end{aligned} \tag{5}$$

The reader will observe from (5) that the first essential difference between Nakamoto (2008) trust and trust from mutually-beneficial repeated interaction is on the cost-of-attack side. In Schelling (1956), the cost of attack is the loss of the net present value of the mutually-beneficial relationship.[21] The more valuable is the relationship, the more value is secured against attack. In Nakamoto trust, in contrast, the repeated interaction among providers of trust support (e.g., miners) is zero profit, as if $\tau = 0$. Instead, the cost of attack comes from a different source, which is that for a short period of time (a single play of the game in Section 3.5) an attacker has to be larger than the honest trust-support providers who are themselves playing a zero profit game (in the limit as $I \to \infty$). Said differently, in Schelling (1956) the cost of attack is a stock (the net present value of the relationship) whereas in Nakamoto (2008) the cost of attack is a flow (the recurring costs of honest trust support for a short period of time).

The second essential difference between these two forms of trust is the cost of honest play. In Nakamoto trust, honest play costs the same p per period as it costs to attack—that is the equilibrium cost to induce the zero-profit trust support on a continual basis. In Schelling (1956), conditional on an existing trust relationship, the recurring cost of honest play is *zero*. Of course, in many contexts the development of a trust relationship in the first place may take investment (e.g., investment in a brand), so one can think of the per-period cost as the interest on this fixed cost investment.

Table 1 summarizes this comparison. It is also worth noting that Schelling trust on its own likely is not enough for high-value financial applications. In financial applications, one can think of τ as the gains from trade in a transaction, and think of b as like the *nominal value* of the transaction (Budish and Sunderam, 2023). So it takes a lot of repeated interaction to sustain

[21] From Schelling (1956): "What makes many agreements enforceable is only the recognition of future opportunities for agreement that will be eliminated if mutual trust is not created and maintained, and whose value outweighs the momentary gain from cheating in the present instance." See Wolitzky (2022) for a recent wide-ranging survey of models of trust from repeated interaction.

Table 1: Comparison of Nakamoto Trust and Schelling Trust

	Cost of Attack	Cost of Honest Play
Nakamoto Trust	p, a flow cost. Attacker pays cost of honest trust support for a short period of time.	p per period. Cost to induce the zero-profit honest trust support on a recurring basis.
Schelling Trust	$V_{trust} = \frac{1}{1-\delta}\tau$, a stock cost. Attacker loses the net present value of the trust relationship.	0 per period, given pre-existing trust relationship.

Notes: See discussion in the text of Section 3.6. Entries for Nakamoto trust are based on repeated-play of the one-shot game analyzed in Section 3.5 in the limit as the number of players grows large.

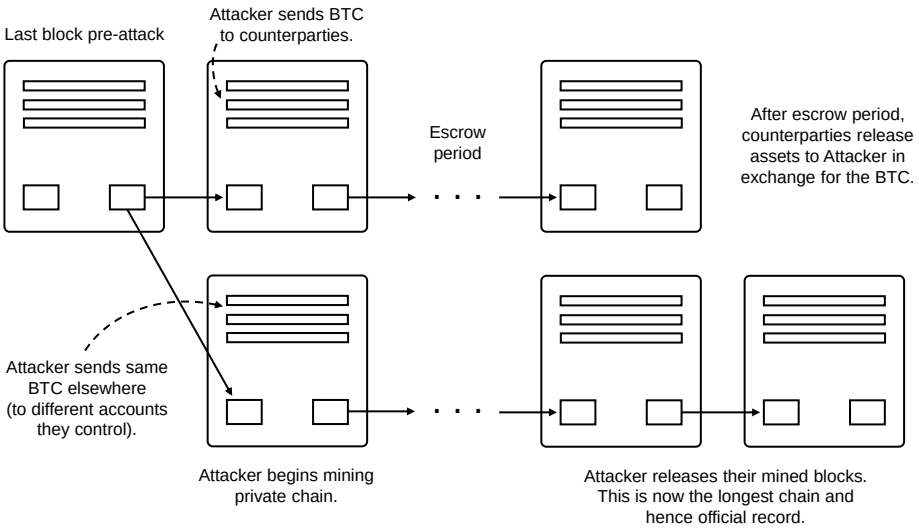
trust for high stakes—it is helpful to have some support from rule of law and collateral. We will return to this topic in Section 6.

4 Analysis of Double-Spending Attacks

Equation (3) tells us that the possibility of a double-spending attack places equilibrium constraints on Nakamoto's novel form of trust. This section is an effort to understand these constraints quantitatively. For this quantification exercise, I focus specifically on Nakamoto (2008)'s proof-of-work longest-chain protocol. Section 4.1 briefly describes the mechanics of a double-spend attack. Section 4.2 analyzes the expected duration and cost of double-spend attacks. Section 4.3 analyzes the implied equilibrium cost of Nakamoto trust using two related thought experiments for the value of attack. First, given a potential dollar value of V_{attack}, what is the equilibrium cost of security in dollars and as a percentage of V_{attack}? Second, given a potential dollar volume of honest transactions using Nakamoto trust, denoted V_{honest}, and an assumption that an attacker can double-spend this honest volume for a period of time, what is the equilibrium cost of security as a percentage of honest transaction volume?

A caveat for this section is that there is no perfect way to model V_{attack}. These two thought experiments reflect my best attempt to analyze the equilibrium implications of the analysis in Section 3 in as transparent a manner as possible. Future research may improve on these efforts.

Figure 3: Illustration of Double-Spending Attack



Notes: See the text for description.

4.1 Mechanics of a Double-Spend Attack

The canonical attack Nakamoto (2008) worries about is called a double spend, in which an attacker is able to spend the same currency multiple times by effectively deleting some of their transactions from the record. Such attacks are also called safety or consistency violations in the distributed consensus literature. In a double-spending attack on a longest-chain consensus protocol, the attacker engages in the following actions in sequence:

(i) The attacker sends cryptocurrency in exchange for goods or assets, potentially in many transactions over many blocks.

(ii) The attacker allows those transactions to be added to the blockchain in the usual way as described in Section 2.2.

(iii) The attacker works in secret to create an alternative longest chain that does not include the transactions in (i). Instead they send the cryptocurrency to other accounts they control.

(iv) The attacker waits for any escrow periods to elapse, so they receive the goods or assets they transacted for in (i).

(v) The attacker releases their alternative longest chain. The attacker now has both the goods or assets they transacted for and their cryptocurrency.

See Figure 3 for an illustration.

4.2 Duration and Gross Cost of Attack

The denominator of the right-hand-side of equation (3), $A^* \cdot t(A^*)$, is the cost of a double-spend attack in units of equilibrium per-block trust-support costs N^*c. It is possible to obtain a closed-form expression for the expected duration t of a double-spending attack as a function of the attacker majority A and other parameters that affect duration. Specifically, let e denote the escrow period, and let k denote the number of blocks in which the attacker places transactions that they will subsequently revert (i.e., the number of blocks in step (i) of the attack as described above). Assume that honest participants produce new blocks as a Poisson process with arrival rate 1 and the attacker produces new blocks as a Poisson process with arrival rate A.

Proposition 3. *For Nakamoto (2008) proof-of-work longest-chain consensus, the expected duration t of the double-spending attack as a function of the attacker majority A, escrow period e, and number of blocks in which the attacker places transactions k, is given by:*

$$t(A,e,k) = (k+e) + \left[\sum_{i=0}^{k+e} \left(\frac{i+1}{A-1}\right) \cdot \frac{(2(k+e)-1-i)!}{(k+e-i)!(k+e-1)!} \left(\frac{A}{1+A}\right)^{k+e-i} \left(\frac{1}{1+A}\right)^{k+e}\right]. \quad (6)$$

As the attacker majority grows large ($A \to \infty$), $t(A,e,k)$ converges to $k+e$. In the limit as $A \to_+ 1$, we have $t(A,e,k) \to \infty$.

Proof. See Appendix B. □

Prior work by Grunspan and Pérez-Marco (2018, 2022) derives closely related mathematical results for the analysis of an attacker with less than 50% of the total hash power. Expression (6) can be understood as follows. In the attacker's best case their attack takes $k+e$ time. This best case occurs if, as soon as the honest participants have produced $k+e$ blocks and all of the assets the attacker has transacted for have been released from escrow, the attacker is ready with their alternative longest chain of at least $k+e+1$ blocks. Suppose, on the other hand, that the attacker is behind the honest chain by $i \geq 0$ blocks at the time the honest participants produce their $k+e$ block. Given the Poisson arrival processes, it will take the attacker $\frac{i+1}{A-1}$ of time in expectation to strictly surpass the honest chain. The last part of the expression gives the probability that the attacker's deficit is i blocks.

Table 2 provides example calculations of duration t and the gross trust-support-cost term At. For example, if $k+e = 12$ blocks (2 hours), then attacker majorities of $A = 1.2 - 1.5$ generate average attack durations of 13.4-19.7 blocks, or about 2-3.5 hours, and average gross costs of 20.1-23.6 times N^*c. Notably, smaller majorities lead to significantly longer attack durations and

Table 2: Expected Duration and Gross Cost of Attack

A. Expected Duration of Attack in Units of Honest Block Time t

Attacker Majority	# Blocks of Double Spending + Escrow Period $(k+e)$					
	1	6	12	36	144	1,008
$A = 1.05$ (51%)	25.51	42.43	56.59	97.91	229.61	1,074.55
$A = 1.10$ (52%)	13.02	22.88	31.73	60.14	167.31	1,010.90
$A = 1.20$ (55%)	6.79	13.24	19.69	43.10	146.79	1,008.00
$A = 1.33$ (57%)	4.34	9.60	15.35	38.04	144.18	1,008.00
$A = 1.50$ (60%)	3.08	7.84	13.43	36.47	144.00	1,008.00
$A = 2.00$ (67%)	1.89	6.45	12.20	36.01	144.00	1,008.00
$A = 5.00$ (83%)	1.12	6.00	12.00	36.00	144.00	1,008.00

B. Expected Gross Cost of Attack in Units of Per-Block Honest Trust-Support N^*c

Attacker Majority	# Blocks of Double Spending + Escrow Period $(k+e)$					
	1	6	12	36	144	1,008
$A = 1.05$ (51%)	26.78	44.56	59.42	102.81	241.09	1,128.28
$A = 1.10$ (52%)	14.32	25.16	34.90	66.16	184.04	1,111.99
$A = 1.20$ (55%)	8.14	15.89	23.63	51.72	176.15	1,209.60
$A = 1.33$ (57%)	5.78	12.77	20.41	50.59	191.76	1,340.64
$A = 1.50$ (60%)	4.62	11.76	20.14	54.71	216.01	1,512.00
$A = 2.00$ (67%)	3.78	12.89	24.39	72.02	288.00	2,016.00
$A = 5.00$ (83%)	5.59	30.02	60.00	180.00	720.00	5,040.00

Notes: Expected duration t as a function of the attacker majority A, escrow period e and the number of blocks in which the attacker places transactions k is computed using formula (6) in the text and double-checked using a computational simulation. Each block corresponds to 10 minutes in expectation. In clock time, the columns correspond to 10 minutes, 1 hour, 2 hours, 6 hours, 1 day, 1 week.

higher costs. For example, if $A = 1.05$, which corresponds to a 51.2% attacker majority, the average duration is 56.6 blocks (9.5 hours) and the average gross cost is 59.4 times N^*c.

As $k+e$ grows larger, the average attack duration t gets proportionally closer to the best-case duration $k+e$ for all values of A. The intuition for this is simple law-of-large numbers. However, even for very large values of $k+e$, such as $k+e = 1008$ which corresponds to a full week, the gross-cost-minimizing attacker majority is larger than 51%. It is true that a 51% majority is enough to ensure statistically that the attack will eventually succeed, but a cost-minimizing attacker will choose a somewhat larger majority.

Appendix B provides numerical analysis of the cost-minimizing attacker majority A^* and min-

imum gross costs $A^* \cdot t(A^*)$ as a function of $k + e$.

4.3 Implied Equilibrium Cost of Nakamoto Trust

4.3.1 Thought Experiment 1: What is the Equilibrium Cost of Security for a Given Value of V_{attack}?

A majority attacker will not use their majority to double spend for a cappucino at Starbucks. They will use their majority to conduct transactions that are as large as possible given the current uses of cryptocurrencies, possibly using many different addresses over many blocks of transactions. V_{attack}, therefore, can be understood as a statistic on the economic usefulness of Nakamoto trust. The higher is the economic throughput of the blockchain for honest participants, the more value an attacker can double spend.[22]

In this first thought experiment, I simply consider a wide range of dollar values for V_{attack}. I use \$1,000 as the low-end of this range, representing Bitcoin's early days when even buying a pizza was remarkable. I use \$100 billion to represent the high-end of this range. While arbitrary, this seems a reasonable order of magnitude for a large-scale attack on the global financial system—e.g., the U.S. financial system has daily transaction volume that conservatively exceeds \$4 trillion (Budish and Sunderam, 2023). This figure also represents about 7% of Bitcoin's peak market capitalization.

Table 3 presents results for a base case scenario in which $k + e = 12$. This corresponds to an escrow period of one hour ($e = 6$), as is standard practice for Bitcoin, and an assumption that the attacker spreads their double-spend transactions over one hour's worth of blocks ($k = 6$).[23] To keep the Nakamoto blockchain secure in this base case requires a per-block cost that is 5.00% of the value secured against a double-spending attack. This follows directly from equation (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{A^* \cdot t(A^*)}$, with $t(A^*) = 13.99$ at the attacker-optimal choice of $A^* = 1.43$ (or 59%). Per transaction, assuming 2000 transactions per block, the cost is 0.0025% of the value secured against attack.

These costs likely sound economically plausible. But consider how they scale with time and with the amount of value secured. A cost of 5% per block amounts to 720% of the value secured

[22] This point likely seems obvious, but it was missed in past academic literature on double-spending attacks. The computer science literature did not explicitly model the economic benefits of attack, and therefore missed that the value of attack might scale with Bitcoin's usefulness (Rosenfeld, 2014; Eyal and Sirer, 2014; Bonneau, 2016). Within economics, a model of Chiu and Koeppl (2022) assumes that an attack involves just a single transaction and holds this transaction size fixed. The authors conclude that the system becomes more secure as its economic value grows relative to this fixed transaction size. This is like noting that it is less attractive to engage in a double-spending attack for a cappucino in 2024 than it was in 2009.

[23] A prior draft did not have the k parameter and instead assumed that the attacker placed all their double-spend transactions in a single block. This is is equivalent to assuming $k = 1$, making $k + e = 7$ the base case in that version with Bitcoin's standard escrow period of $e = 6$. This accounts for the difference in the numbers in Table 3.

Table 3: Cost to Secure Against Attack: Base Case Analysis

	Per-Block	Per-Day	Per-Year	Per-Transaction
Security Costs as % of Value Secured	5.00%	720%	262,801%	0.0025%
To Secure:				
$1 thousand	$50.0 dollars	$7.2 thousand	$2.6 million	2.5 cents
$1 million	$50.0 thousand	$7.2 million	$2.6 billion	$25.0 dollars
$1 billion	$50.0 million	$7.2 billion	$2.6 trillion	$25.0 thousand
$10 billion	$500.0 million	$72.0 billion	$26.3 trillion	$250.0 thousand
$100 billion	$5.0 billion	$720.0 billion	$262.8 trillion	$2.5 million

Notes: See equation (3) and the text of Section 4.3.1 for description. The Base Case scenario assumes $k + e = 12$ blocks (2 hours).

against attack per day, and about 263,000% of the value secured against attack per year. For example, to secure the system against a $1 billion attack requires $2.63 trillion of annual security expense. To secure the system against a $40 billion attack requires $105 trillion per year, or all of 2023 global GDP per the World Bank.

The per-transaction fee of 0.0025% likely sounds very small, but this is a percentage of the value secured against attack, not the size of the transaction. For example, if an attack could be worth $1bn, then each transaction must pay 0.0025% of $1bn which is $25,000 of security costs. The intuition is that every transaction has to implicitly pay for the costs of the large standing army—as if a transaction for a cappucino has the security required by Fort Knox.

Table 4 presents a sensitivity analysis. As the escrow period e grows, and the number of blocks k it takes the attacker to execute their double-spend transactions grows, the cost of security declines. Whereas the cost of security is 5% of V_{attack} per block in the base case, the cost declines to 0.09% of V_{attack} if $e + k$ amounts to a week ($e + k = 1008$). However, even in the scenario where $e + k$ equals a week, the cost to secure against large attacks remains high. To secure against a $10 billion attack requires an annual security cost of $478 billion, which is more than the United States' annual spending on prisons, courts and police (see Section 6). To secure against a $100 billion attack requires an annual security cost of 4.5% of global GDP.

4.3.2 Thought Experiment 2: What is the Equilibrium Cost of Security as a % of Honest Transaction Volume?

As a second thought experiment, assume that honest users of a cryptocurrency transact an average of V_{honest} of volume per block, and that a majority attacker can double spend exactly this average

Table 4: Cost to Secure Against Attack: Sensitivity Analysis

	# Blocks of Double Spending + Escrow Period ($k+e$)					
	1	6	12	36	144	1,008
Per-Block Security Costs as % of V_{attack}	26.74%	8.55%	5.00%	1.99%	0.57%	0.09%
Annual Security Costs if V_{attack} Equals:						
\$1 thousand	\$14.1 M	\$4.5 M	\$2.6 M	\$1.0 M	\$301.3 K	\$47.8 K
\$1 million	\$14.1 B	\$4.5 B	\$2.6 B	\$1.0 B	\$301.3 M	\$47.8 M
\$1 billion	\$14.1 T	\$4.5 T	\$2.6 T	\$1.0 T	\$301.3 B	\$47.8 B
\$10 billion	\$140.5 T	\$44.9 T	\$26.3 T	\$10.5 T	\$3.0 T	\$478.2 B
\$100 billion	\$1.4 Q	\$449.1 T	\$262.8 T	\$104.7 T	\$30.1 T	\$4.8 T

Notes: Each block corresponds to 10 minutes in expectation. In clock time, the columns correspond to 10 minutes, 1 hour, 2 hours, 6 hours, 1 day, 1 week. The abbreviations K = Thousand; M = Million; B = Billion; T = Trillion; Q = Quadrillion.

Table 5: Costs to Secure Against Attack as % of Honest Volume

		Escrow Period (e)					
		1	6	12	36	144	1,008
V_{attack} # Blocks of Honest Volume (k)	1	18.07 %	7.61 %	4.69 %	1.94 %	0.57 %	0.09 %
	6	45.68 %	30.00 %	21.55 %	10.45 %	3.31 %	0.54 %
	12	56.26 %	43.10 %	33.85 %	18.58 %	6.39 %	1.08 %
	36	70.01 %	62.67 %	55.73 %	38.82 %	16.78 %	3.17 %
	144	82.02 %	79.50 %	76.66 %	67.14 %	43.26 %	11.52 %
	1,008	91.63 %	91.20 %	90.68 %	88.67 %	80.63 %	46.87 %

Notes: Each block corresponds to 10 minutes in expectation. In clock time, the columns correspond to 10 minutes, 1 hour, 2 hours, 6 hours, 1 day, 1 week.

honest volume per block, for k blocks total. That is, $V_{attack} = kV_{honest}$. This is a simple structural model in which the value of a majority attack grows with economic usefulness, as measured by the amount of honest transaction volume. I caution that this thought experiment has likely biases in both directions. It is conservative in that a majority attacker would seek to transact the largest possible amounts, not just average amounts.[24] It is aggressive in that a majority attacker would likely share block space with honest users whose transactions happen to be in the mempool at the

time the attacker starts double spending.

Table 5 presents results for this thought experiment. In the base case in which the escrow period is 1 hour ($e = 6$) and the attacker can double spend for 1 hour worth of blocks ($k = 6$), i.e., $V_{attack} = kV_{honest}$ corresponds to one hour of average honest transaction volume, the equilibrium cost of security is 30% of V_{honest} per block.[25] This percentage cost increases with the number of blocks the attacker can double spend for and decreases with the escrow period. For example, if the attacker can double spend one hour of average honest transaction volume ($k = 6$), but the escrow period is one day ($e = 144$), then the required security cost is just 3.31% of honest transaction volume— roughly in line with the costs of credit card transactions in the traditional financial system. In the other direction, if the escrow period is the standard one hour ($e = 6$) but the attacker can double spend one day's worth of average honest transaction volume ($k = 144$), then the required security cost is 79% of V_{honest} per block. For any fixed escrow period, the cost converges to 100% of V_{honest} as $k \to \infty$.

4.4 Discussion

The double-spending analysis is consistent with the modest early use cases of Bitcoin, in which Bitcoin was primarily used by hobbyists and for small-scale black market activity (e.g., online gambling, Silk Road). In these early days, the amount that could be gained in a double-spending attack was not very high, because there were not high-value transaction opportunities. If a double-spending attack could gain at most $1,000, then the implicit cost per transaction in the base case necessary to secure the trust is just $0.025.

The double-spending analysis is also consistent with larger-scale black-market uses of cryptocurrencies, especially as black-market users may be most willing to pay the high implicit costs. For example, if a double-spending attack could gain at most $10 million, then the implicit cost per transaction in the base case needs to be about $250. This is modest relative to the costs of transporting large amounts of cash (Rogoff, 2017).

[24] Let honest transaction volume be drawn from a distribution with support on $[\underline{V}, \overline{V}]$ and mean V_{honest}. For the system to facilitate honest transaction volume for any quantity drawn from the distribution, it must be the case that it is not worthwhile to double-spend even for the maximum honest transaction value $\overline{V}$. We can use data on the real-world distribution of blockchain transaction volume to understand the empirical relationship between observed $\overline{V}$ and V_{honest}. If we look at Bitcoin volume in BTC in 2023, the ratio of maximum-to-mean volume $\frac{\overline{V}}{V_{honest}}$ is about 7.5 at the 1-hour level, 5.2 at the 2-hour level, and 1.85 at the 1-day level. Makarov and Schoar (2021) find that about 90% of Bitcoin volume is spurious. Under the strong assumption that the attacker double-spends the maximum observed transaction volume and it is all non-spurious, then these ratios of $\frac{\overline{V}}{V_{honest}}$ could be multiplied by a factor of 10.

[25] Table C.1 in the appendix compiles data on double-spend attacks on forks of Bitcoin and Ethereum. The Bitcoin Gold attacks had a length of largest reorganization of 16-22 blocks (roughly 2.5-4 hours). The Ethereum Classic attacks had lengths of 140 blocks to 7000 blocks (30 minutes to 1 day). The details of these attacks are thinly reported but may give a sense of the range of realistic values of k and e.

Where the analysis suggests greater skepticism is the use of cryptocurrencies and Nakamoto trust as a major component of the mainstream global financial system (again, considering its pure form without support from rule of law). If cryptocurrencies and Nakamoto trust were to become more integrated with the mainstream global financial system, then it would be possible to move amounts of value that are ordinary in the scheme of global finance, and hence it would be possible to double spend for amounts of value that are ordinary in the scheme of global finance. The analysis suggests that this scenario is unrealistic because of the way the trust model scales. To secure the system against attacks of $1 billion—which is less than 0.2% of daily trading volume in the U.S. Treasury market alone—requires a per-transaction security cost of $25,000 and an annual security cost of $2.6 trillion. To secure against attacks of $40 billion requires an annual security cost of all of global GDP. While market power and fees in traditional finance are clearly an important economic issue (Greenwood and Scharfstein, 2013; Philippon, 2015), and Huberman, Leshno and Moallemi (2021) are careful to remind us to compare the costs of the Nakamoto trust model against the costs of market power in traditional finance, it is clear from these calculations that Nakamoto trust is very expensive relative to traditional trust. We will return to this comparison between Nakamoto trust and traditional trust in Section 6.

A conceptual insight that is reinforced by the double-spending analysis is that blockchain security should be thought of not as a 0-1 variable that breaks at a threshold ρ, as in the distributed consensus literature, but as more like a (high) percentage tax.

5 Specific Capital and Collapse

Nakamoto (2008) envisioned that ordinary computers would be used to maintain the Bitcoin blockchain, famously using the phrase "one-CPU-one-vote". Since 2013, however, Bitcoin mining has been dominated by specialized computer chips called ASICs (application-specific integrated circuits). ASICs have the SHA-256 hash function etched directly onto hardware, which makes them extremely efficient at Bitcoin mining and useless for any other application that does not involve performing a large number of SHA-256 hashes.[26] Many of the other largest cryptocurrencies use a proof-of-stake consensus protocol, most prominently Ethereum since fall of 2022. With proof-of-stake, the capital used to maintain the blockchain is intrinsically specialized to the blockchain, as the capital is literally units of the cryptocurrency.

[26]These specialized chips are so much more efficient than general-purpose chips that execute SHA-256 in software that, I estimate, even if one controlled *all* of Amazon Web Services that would amount to about 0.05% of Bitcoin's hash rate. This calculation is based on AWS owning $65 billion of technology capital per its 2021 10-K filing, the calculations below that the Bitcoin capital stock is about $12.5 billion, and an assumption that specialized ASIC chips are at least 10,000 times more economically efficient at SHA-256 hashing than general-purpose computers.

If the capital used to maintain the blockchain is specialized (as opposed to repurposable), and a majority attack causes the attacker's capital to lose its value or be confiscated, then the attacker cost model needs to be modified. In addition to charging the attacker the flow cost of the attack, we also need to charge the attacker the stock value of their loss of specialized capital. This makes an attack significantly more expensive.

Section 5.1 redoes the theoretical analysis of Section 3 under an alternative incentive constraint that includes the stock value of the attacker's capital. Section 5.2 considers three logical possibilities for how an attacker might lose their capital value: collapse of the cryptocurrency; internal-to-protocol punishment without collapse; and external-to-protocol responses to the attack. Section 5.3 discusses the analysis.

5.1 Analysis if Attacker Loses their Capital

Let $c = rC + \eta$ denote the cost per unit time to supply one unit of trust support, where C denotes the fixed cost of specialized capital (e.g., ASICs, stake), r denotes the rental cost of capital per unit time (risk-adjusted interest expense plus depreciation), and η denotes variable costs per unit time (e.g., electricity). The honest Nakamoto trust equilibrium condition (1) can be rewritten:

$$N^*(rC + \eta) = p_{block}. \tag{7}$$

An outside attacker would need at least N^*C worth of specialized capital to conduct the attack, while an inside attacker would need at least $\frac{N^*C}{2}$ of capital.

For this subsection, let us focus on an outside attacker who loses all of their capital as a result of the attack, e.g., because of a total collapse of the cryptocurrency—this is the case for which the cost of attack is highest. Given how small the flow costs of attack are, as analyzed in Section 4, let us ignore these and focus only on the stock cost of the specialized capital. This yields an approximate attack cost of N^*C and an approximate incentive compatibility condition of

$$N^*C > V_{attack}. \tag{8}$$

We can compute N^*C as a function of p_{block}. Let $\mu = \frac{rC}{rC+\eta}$ denote the capital share of trust support. The honest equilibrium (7) can be rewritten:

$$N^*C = \frac{\mu p_{block}}{r}. \tag{9}$$

Hence we can derive a modified version of equation (3):

$$p_{block} > \frac{r}{\mu} V_{attack}. \tag{10}$$

This is several orders of magnitude more secure than (3) because r is the interest rate per unit time. Here is an example calculation for Bitcoin. Assume the capital share of mining is $\mu = 0.4$ (De Vries, 2018; Digiconomist, 2022), and the annual discount rate for ASICs is 50% (ASICs depreciate quickly and mining is risky), which implies that the per-unit-time discount rate is $r \approx 0.0008\%$. This means $\frac{r}{\mu} = 0.002\%$. Now compare $\frac{r}{\mu}$ on the right-hand-side of (10) to the $\frac{1}{A^* \cdot t(A^*)}$ factor on the right-hand-side of (3). If we use the base case value of $\frac{1}{A^* \cdot t(A^*)} = 5.0\%$, we have a roughly 2500-fold improvement in the cost of security.

If we use these same values for μ and r and use p_{block} of \$250,000, then (9) implies a capital stock of \$12.5 billion, which about matches what is implied by current prices for state-of-the-art ASIC machines.[27] This suggests these magnitudes are reasonable.

5.2 Issue: How Does Attacker Lose their Capital?

5.2.1 Collapse of the Cryptocurrency

One way the attacker can lose their capital value is if the majority attack causes a significant decline in the value of the cryptocurrency. For example, a majority attack on a major cryptocurrency would be widely reported and might cause a market crash.

Mathematically, suppose the majority attack causes a proportional decline in the value of the cryptocurrency of Δ_{attack}. If the specialized capital is stake, and the attacker is unable to withdraw their stake before the decline occurs, then their stake declines in value by $\Delta_{attack} N^* C$. If the specialized capital is hardware, and the attacker is unable to liquidate their hardware before the decline occurs, then the decline can be faster than rate Δ_{attack}. Specifically, if we assume that the market is in the equilibrium (7) prior to the attack and we assume that the post-attack equilibrium value of specialized capital reflects a permanent decline in block rewards of Δ_{attack}, then the specialized capital declines in value by proportion $\max(1, \frac{\Delta_{attack}}{\mu})$.[28]

[27] A Bitmain Antminer S19j XP has a current retail price of \$4,983 (https://www.bitmain.com/, accessed August 31, 2023) and it would take about 2.3 million of these machines to match Bitcoin's current hash rate, for a total capital cost of about \$11.5 billion at retail prices. I do not have any information on how retail prices relate to the prices paid by large-scale miners.

[28] Starting with the equilibrium $N^*(rC + \eta) = p_{block}$, we wish to recompute the post-attack equilibrium value of capital C' given a decline in rewards to $(1 - \Delta_{attack})p_{block}$. This can be computed as $N^* rC' = (1 - \Delta_{attack})p_{block} - N^*\eta = (1 - \Delta_{attack})p_{block} - (1 - \mu)p_{block}$. So the ratio $\frac{C'}{C} = \frac{(1-\Delta_{attack})-(1-\mu)}{1-(1-\mu)} = \frac{\mu - \Delta_{attack}}{\mu} = 1 - \frac{\Delta_{attack}}{\mu}$. If Δ_{attack} exceeds μ then the post-attack capital value is zero and some capital will be mothballed given the decline in compensation. A richer model would reflect the stochastic nature of p_{block} and the associated option-value of capital, along the lines of Prat and Walter (2021), both before and after the attack.

Thus, if Δ_{attack} is large this significantly increases the attacker's costs. The Bitcoin Wiki classifies the majority attack into its "Probably Not a Problem" category for this reason (Bitcoin Wiki, 2020).[29] However, vulnerability to collapse is undesirable for two related reasons. First, collapse harms honest holders of the cryptocurrency and the specialized capital. Second, there is the possibility of an attack motivated by this harm per se, i.e., a sabotage attack. The possibility of a sabotage attack was first raised by Rosenfeld (2014).[30]

What is the value of a sabotage attack on a significant cryptocurrency such as Bitcoin or Ethereum? It is hard to say of course, but easy to imagine that the magnitudes are already large, and would be larger still if cryptocurrencies become more significantly integrated into the global financial system. Open interest in CME Bitcoin futures as of April 2024 is about 28,000 contracts, each tracking 5 Bitcoins, worth about $9 billion at current prices. According to data from The Block, open interest in Bitcoin futures aggregated across the major crypto exchanges is about $24 billion as of April 2024, and about $9.5 billion for Ethereum futures.[31] These figures give a sense of magnitudes for what could be made from a short-selling attack.

The market capitalization of cryptocurrencies gives another sense of magnitudes for the amount of economic harm an attacker could cause. Bitcoin's market capitalization has been as high as about $1.4 trillion and Ethereum's as high as about $500 billion. Across all crypto assets tracked by CoinMarketCap, market capitalization peaked at about $3 trillion. Paypal co-founder Peter Thiel (2022) recently predicted that Bitcoin will be worth more than $100 trillion.

Last, Ethereum founder Vitalik Buterin described a future in which it is "just considered normal for there to be *trillion dollar assets* that are managed on Ethereum." (Klein, 2022, emphasis added). If indeed assets of that magnitude are managed on Ethereum or other blockchains, without implicit or explicit protections from rule of law, then the value and risk of sabotage would be large.

[29] "A miner with more than 50% hash power is incentived [sic] to reduce their mining power and refrain from attacking in order for their mining equipment and bitcoin income to retain its value."

[30] "In this section we will assume $q < p$ [i.e., that the attacker does not have a majority]. Otherwise, all bets are off with the current Bitcoin protocol ... The honest miners, who no longer receive any rewards, would quit due to lack of incentive; this will make it even easier for the attacker to maintain his dominance. This will cause either the collapse of Bitcoin or a move to a modified protocol. As such, *this attack is best seen as an attempt to destroy Bitcoin*, motivated not by the desire to obtain Bitcoin value, but rather wishing to maintain entrenched economical systems or obtain speculative profits from holding a short position." (Emphasis Added)

[31] CME open interest data is available via its website. I found open interest data from crypto exchanges at https://www.theblock.co/data/crypto-markets/futures/. I believe this to be a credible source but am less confident in it than I am the CME figures. For what it's worth, when I wrote the June 2018 draft of this paper, CME + CBOE open interest was about $160 million, and crypto exchange futures did not, to my knowledge, exist at the time. That is, futures market open interest has grown by two orders of magnitude in the past few years.

5.2.2 Internal-to-Protocol Punishment without Collapse

It would be very attractive to get the security benefits of an attack costing a stock not a flow, that is an attack that costs $O(N^*C)$ yielding equilibrium constraint (10) rather than costing $O(N^*c)$ yielding equilibrium constraint (3), without needing the cryptocurrency to collapse. This is what Ethereum proof-of-stake consensus with "slashing" is trying to accomplish.

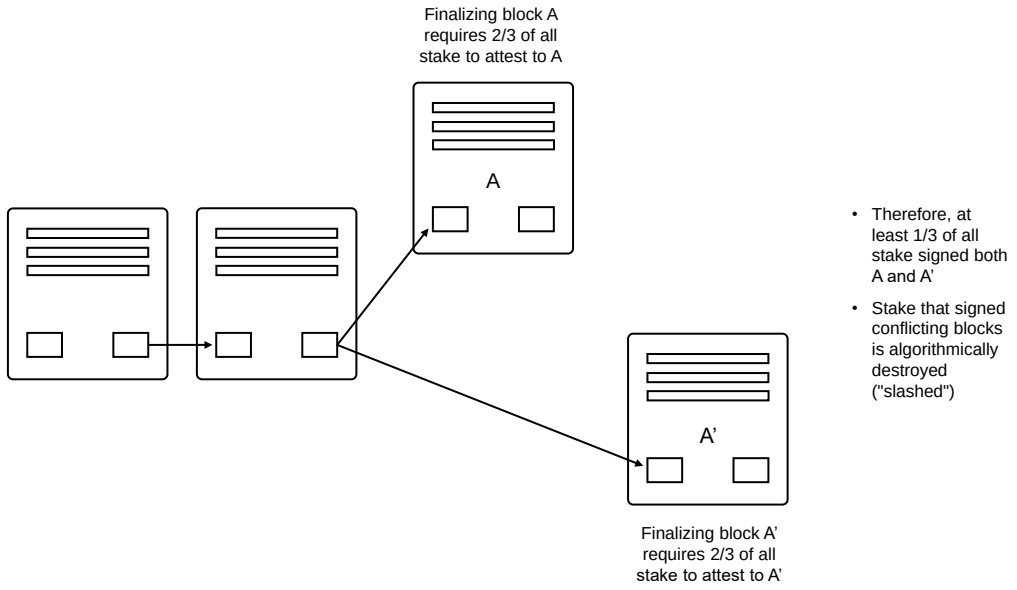
Slashing relies on two distinct departures from Nakamoto (2008)'s version of permissionless consensus. First, the trust-support capital is stake that exists on-chain, as opposed to computational equipment that exists in the physical world off-chain. Second, Ethereum's consensus protocol requires, roughly speaking, at least $\frac{2}{3}$ of all of the trust-support capital to sign a block before it is added to the record, in what is known as the Byzantine Fault Tolerance (BFT) paradigm. These two features together mean that, in the event of a double-spending attack, the attacker will leave cryptographic proof that they have violated the protocol. If blocks A and A' conflict each other (e.g., send the same currency to two different places), then for both to be confirmed, at least $\frac{2}{3}$ of all of the stake has to have signed A and $\frac{2}{3}$ has to have signed A', so at least $\frac{1}{3}$ of all of the stake has to have signed both conflicting transactions. The idea of slashing is simply to confiscate the stake that has signed conflicting transactions—to "slash" it from the record. See Figure 4.

Intuitively, slashing is trying to mimic the traditional trust combination of collateral plus rule-of-law. A counterparty posts collateral and if they cheat the legal system can seize the collateral. The difficulty is that a large-enough attacker can effectively control or stall the protocol's legal system. An impossibility result of Tas et al. (2023) (Theorem 1) shows that is is impossible for a proof-of-stake protocol to achieve any positive amount of what they call "slashable safety," meaning confiscating any positive amount of the attacker's stake, if the attacker can be large enough. The proof shows that a large-enough attacker (specifically, with at least $\rho = \frac{2}{3}$) can always withdraw all of the capital that signed conflicting transactions before the confirmation of any transactions that slash their stake.

Budish, Lewis-Pye and Roughgarden (2024) provide a possibility result for proof-of-stake with slashing to successfully confiscate the attacker's capital that signed conflicting transactions if both (i) the attacker is strictly less than $\rho = \frac{2}{3}$ of the total stake, and (ii) there is a known finite bound on the maximum possible network delay.[32] The protocol used to prove the result has a

[32]Assumption (ii) is known as the synchronous communications assumption and is considered a strong assumption in the distributed consensus literature (see Lewis-Pye and Roughgarden, 2024). Budish, Lewis-Pye and Roughgarden (2024) prove an impossibility result for slashing for the partially synchronous communications model of Dwork, Lynch and Stockmeyer (1988). Budish, Lewis-Pye and Roughgarden (2024) also prove additional impossibility results for economic security from targeted punishment for any permissionless consensus protocol in the class of "fully permissionless" or "dynamically available" protocols. The fully permissionless class includes Bitcoin proof-of-work and the dynamically available class includes some other forms of proof-of-stake based on longest-chain consensus rather than the BFT paradigm.

Figure 4: Proof-of-Stake with Slashing



Notes: Double-spending attacks (or more generally safety violations) leave on-chain proof of malfeasance in BFT-style proof-of-stake consensus. This proof can be the basis for the protocol's algorithmically confiscating the attacker's capital, unless the attacker is so large that they can prevent the punishment from happening before they remove their guilty stake from the system. See the text for description.

lock-up period for stake that is long relative to the maximum possible honest network delay, and constructs an in-protocol recovery-from-attack procedure that a sub-$\frac{2}{3}$ attacker cannot thwart. A rough intuition for how recovery is possible is that an attack requires at least $\frac{1}{3}$ of the total stake to sign conflicting transactions (per Figure 4), so a sub-$\frac{2}{3}$ attacker is thus left with $< \frac{2}{3} - \frac{1}{3} = \frac{1}{3}$ of stake that is not accused of attack, which is less than 50% of the total non-accused stake.

A separate issue with BFT-style consensus is what are known as liveness attacks. Because $\frac{2}{3}$ of all stake has to sign each block, an attacker with greater than $\rho = \frac{1}{3}$ of the total stake can effectively halt transactions for a significant period of time.[33] If Ethereum were to become more integrated into the traditional global financial system this would seem to be a significant source of attack risk. And, unlike for a safety violation as depicted in the figure, a liveness attack does not leave any cryptographic proof of the attacker's malfeasance. The silent stake could be having a legitimate network outage or computer failure. For these reasons, Ethereum does slash stake that is silent for a long period of time but very slowly. By my calculations, one could halt Ethereum

[33]It is possible to improve the safety of BFT-style consensus against double-spending attacks by requiring $1-\rho_l > \frac{2}{3}$ of stake to sign each block, but this lowers the threshold at which the consensus is vulnerable to liveness attacks to $\rho_l < \frac{1}{3}$. See further discussion in Budish, Lewis-Pye and Roughgarden (2024). This intrinsic tradeoff between safety and liveness is familiar in the distributed consensus literature more broadly (Lewis-Pye and Roughgarden, 2024).

for a day at a cost of 0.09% of the total honest stake, two days at a cost of 0.37% of the honest stake, a week at a cost of 4.68% of the honest stake, and two weeks at a cost of 20.53% of the honest stake.[34]

I conclude that Ethereum proof-of-stake with slashing achieves a security improvement over Bitcoin that can be understood through the lens of this paper's analysis: a sub-$\frac{2}{3}$ double-spend attacker incurs costs that are $O(N^*C)$ even if $\Delta_{attack} = 0$. However, to deter $> \frac{2}{3}$ attackers or to deter liveness attacks in the event Ethereum becomes more integrated with global financial markets likely requires a source of trust support external to the protocol, such as rule of law.[35]

5.2.3 External-to-Protocol Punishment

The third logical possibility for how an attacker could lose their capital value is by a punishment external to the blockchain protocol. For example, an attacker who engaged in the short-selling attacks discussed in Section 5.2.1 might face legal consequences, which could include prison and significant financial penalties. This would certainly work in the sense of getting the security benefits of equilibrium constraint (10) rather than (3), but the whole question of this paper is whether Nakamoto's novel form of trust works economically without government and rule of law.

5.3 Discussion

The theoretical analysis of Section 5.1 shows that if the attacker loses their capital as a result of the attack, permissionless consensus becomes significantly more economically attractive. A 2500-times improvement in the cost of trust is enough to transform Nakamoto's novel form of trust from being extremely expensive relative to traditional forms of trust to competitive with traditional trust. For example, referring to Table 3, the cost to secure against an attack of $100 billion goes from about 2.5 times global GDP to just over $100 billion per year, which is the same

[34]These calculations are based on the description of the Inactivity Penalty Mechanism provided by Edgington (2023), which is linked to from Ethereum.org. The latest version of this mechanism was implemented in Ethereum's major update in September 2022 (Bellatrix) and has remained unchanged as of its most recent major update in April 2024 (Deneb). Note that if Ethereum were to increase its security against double-spend attacks as discussed in the previous footnote, it would lower the costs of liveness attacks. For example, I calculate that if Ethereum required 90% of stake to sign each block (instead of $\frac{2}{3}$), then a liveness attack could halt Ethereum for a day at a cost of 0.02% of honest stake, two days at a cost of 0.08% of honest stake, a week at a cost of 1.04% of honest stake, and two weeks at a cost of 4.56% of honest stake.

[35]Ethereum's official developer documentation discusses the risk of $\frac{2}{3}$ attackers in some detail, writing "As the supermajority stakeholder, the attacker would always control the contents of the finalized blocks, with the power to spend, rewind and spend again, censor certain transactions and reorg the chain at will." The discussion concludes "The only real defenses here are the enormous cost of 66% of the total staked ether, and the option to fall back to the social layer to coordinate adoption of an alternative fork." (Ethereum.org, 2023) My understanding of the term "social layer" is that it is meant to encompass any form of trust support external to the protocol itself. My own speculation is that this would include rule of law if large sums of money are under dispute.

order of magnitude as spending on police, courts and prisons in the United States (see Section 6). Or, referring to Table 5, if we assume an attacker can double-spend for $k = 6$ blocks of honest volume and the escrow period is $e = 6$ blocks, then the cost to secure against attack goes from a 30% tax on honest transaction volume to just 0.012% of honest transaction volume, which is cheaper than most payment fees.

The difficulty is how the attacker loses their capital. Collapse risk (Section 5.2.1) does not seem like a plausible foundation for a novel economic system, especially when one considers the possibility of sabotage. In-protocol punishment without collapse (Section 5.2.2) would be very attractive but faces impossibility theorems which suggest it is only a partial solution. External-to-protocol punishment (Section 5.2.3) certainly could work but begs the question of what is accomplished if permissionless consensus is ultimately reliant on traditional rule of law.

6 Comparison of Nakamoto Trust and Traditional Trust

In this section we return to the contrast discussed in the introduction between Nakamoto trust and traditional trust supported by rule of law and complementary sources, such as reputations and relationships. The essential difference is economies of scale.

6.1 Beckerian Deterrence as an Economy of Scale

For concreteness, consider a financial transaction between two parties of size V, but where one of the parties has an opportunity to cheat and steal the other party's assets. Specifically, Party 1 chooses an action from the set $\{Engage,\ Don't\ Engage\}$; Party 2 chooses an action from the set $\{Honest,\ Cheat\}$; if the players choose $Engage$ and $Honest$ then both parties get a payoff of $\tau > 0$, representing the net benefit of an honest transaction; but if Party 1 chooses $Engage$ and Party 2 chooses $Cheat$, then Party 2 gets a payoff of V and Party 1 gets a payoff of $-V$, representing that Party 2 has stolen Party 1's assets. If Party 1 chooses $Don't\ Engage$ then both parties get a payoff of zero. Clearly, in the game as described so far, the only equilibrium is for Party 1 to choose $Don't\ Engage$, so the parties will forego the benefit of transacting.

Now add a legal system with the power to enforce contracts. Specifically, if Party 2 plays $Cheat$, then Party 1 can pay a cost c_l to adjudicate the transaction in court, and the court can perfectly observe whether Party 2 played $Cheat$ or $Honest$. If the court observes that Party 2 played $Cheat$ it can compel Party 2 to return Party 1's assets and punish Party 2 with a large fine f. In this scenario, Party 1's payoff is $-c_l$, the cost of bringing the matter to court, and Party 2's payoff is $-f$, the cost of the fine.

Clearly, the legal system makes it an equilibrium for the parties to transact honestly; the credible threat of a large fine deters Player 2 from cheating. And, since the players will transact honestly on the equilibrium path, the court need not even involve itself with most transactions in the first place. This is the point emphasized in the introduction about the economies of scale in traditional trust that is implicit in Hayek (1960) and Becker (1968). The key insight is: *A society that pays a fixed cost of operating a court system can facilitate honest transactions that have zero marginal cost of security because of the deterrence effect. This is a scale economy for traditional trust.*

We can translate this conceptual point about scale economies for traditional trust into the language of our earlier analysis. Consider the stripped-down model of Section 3.5 augmented in two ways. First, as in Section 4.3.2, add a parameter V_{honest} that represents the average volume transacted per period by honest users of the system if trust is secured. Second, model traditional trust as costing a fixed cost F plus a variable cost per unit transacted of c, such that society's cost of traditional trust is $F + cV_{honest}$ per period. In Figure 1, the c can be interpreted as the cost of the security guards outside the bank and the F can be interpreted as society's cost of police and courts.

The two trust models' cost per unit volume are thus:

$$Traditional\ Trust : \frac{F}{V_{honest}} + c \tag{11}$$

$$Nakamoto\ Trust : \frac{V_{attack}}{V_{honest}}$$

Traditional trust has scale economies to the extent that fixed costs F that support trust can scale over a large quantity of transaction volume V_{honest}. Beckerian deterrence of crime is a leading example. Similar scale economies of trust arise in the private sector from fixed-cost investments in brands, reputation, relationships or collateral. Often such investments work in conjunction with societal fixed-cost investments in rule of law. For example, a firm's brand, reputation or relationships can serve as a credible commitment to provide high quality on those dimensions of quality that are not contractible (Nelson, 1974; Fudenberg, Levine and Maskin, 1994; Tadelis, 1999; Baker, Gibbons and Murphy, 2002; Levin, 2003), while laws or contracts can cover the dimensions that are contractible.

Collateral is a particularly important example to discuss in our context. Imagine that a bank intermediates the transaction above between Party 1 and Party 2 and has general-purpose collateral on its balance sheet that exceeds the value of the transaction. The bank can be trusted not to abscond with the parties' assets, without any appeal to reputation or brand, if a court can

compel it to compensate the parties out of its general-purpose collateral if it cheats. Moreover, the cost of general-purpose collateral as a source of trust support is low because collateral earns a market rate of return. Under the assumptions of the Modigliani and Miller (1958) theorem the cost of collateral as a source of trust support is *zero.* Estimates from the empirical literature on the magnitudes of violations of the Modigliani-Miller theorem find that the cost of collateral is not literally zero, but is less than 1% per year of the collateral amount, which likely translates to less than 0.01% of transaction volume (see Budish and Sunderam, 2023). That is, $\frac{F}{V_{honest}} < 0.0001$ for collateral.

Nakamoto trust, in contrast, only enjoys economies of scale with transaction volume if the scope for attacking the system V_{attack} does not grow with the system's usefulness for honest participants V_{honest}, which seems unlikely without support from rule of law. In the stylized financial transaction above, the size of the attack opportunity equals the size of the honest transaction, i.e., $\frac{V_{attack}}{V_{honest}} = 1$. Indeed, the ratio $\frac{V_{attack}}{V_{honest}}$ could easily exceed 1, as a majority attacker will engage in large transactions whereas V_{honest} measures the size of average transactions.

Table 6 presents a summary comparison of these different forms of trust.

Table 6: Comparison of Trust Models

	Cost to Break Trust	Variable Cost of Honest Play	Fixed Cost Investments that Support Honest Play
Nakamoto Trust in its original form (Nakamoto, 2008)	Flow. Attacker pays cost of honest trust support $p_{block} = N^* c$ for a short period of time.	p_{block} per period.	–
Repeated Interaction (Schelling, 1956; Aumann, 1959)	Stock. Attacker loses the net present value of the trust relationship $V_{trust} = \frac{1}{1-\delta}\tau$.	0, given pre-existing trust relationship.	Investment in trust relationship, brand, reputation.
Credible Deterrence (Hayek, 1960; Becker, 1968)	Arbitrarily high up to value of life and assets. Attacker can be punished by the state.	"Security guards" in Figure 1. Variable costs of monitoring and enforcement associated with specific locations or transactions.	Investment in government and legal system.
Collateral	Stock + Punishment. Attacker loses the value of their collateral plus any additional punishment.	0, under conditions of Modigliani-Miller theorem. Requires legal enforcement mechanism which entails fixed costs.	Collateral value.
Nakamoto with Collapse	Stock. Attacker loses specialized capital worth N^*C.	p_{block} per period.	ASICs or other specialized computational equipment.
Proof-of-Stake with Slashing (Idealized)	Stock. Attacker loses specialized capital worth N^*C.	p_{block} per period.	Cryptocurrency stake.

Notes: Nakamoto trust in its original form: see analysis in Section 3.1-3.2. Repeated interaction: see analysis in Section 3.6. Credible deterrence: see discussion in Section 6.1. Collateral: see discussion in Section 6.1. Nakamoto with collapse: see analysis in Section 5.1 and discussion in Section 5.2.1. Proof-of-Stake with Slashing (Idealized): see analysis in Section 5.1 and discussion in Section 5.2.3; "idealized" means ignoring the issues that arise if the attacker is large enough to prevent the protocol from punishing them before they withdraw their guilty stake and ignoring the risk of liveness attacks.

6.2 Sense of Magnitudes for Finance

Total annual spending on police, prisons and courts, at the state, local and federal level, is about $300 billion per year in the United States (Urban Institute, 2023). Real value added in the U.S. financial industry is about $800 billion per year.[36] So we could use $1 trillion per year as a conservative upper bound for the cost of trust in the U.S. financial sector, since the former figure includes spending that is unrelated to finance and the latter figure includes spending that is unrelated to trust.[37] We can use $1 quadrillion per year as a conservative lower bound for transaction volume in the U.S. financial sector (Budish and Sunderam, 2023). We can thus upper bound the cost of trust support $\frac{F}{V_{honest}} + c$ in traditional finance by 0.1% of transaction volume. Clearly this is just a rough ballpark. Many fees in traditional finance, especially for large transactions, are on the order of 0.01% or less of transaction volume.[38]

7 Conclusion

Nakamoto (2008)'s novel form of trust—a completely anonymous and decentralized "permissionless consensus," without any support from government or trusted intermediaries—is *ingenious but expensive*. Equation (3) says that for the trust to be meaningful requires that the flow cost of maintaining the trust must be large relative to the one-shot value of attacking it. This is like a very large implicit tax. Moreover, the cost of Nakamoto trust scales linearly with the value of the attack—e.g., securing against a $1 billion attack is 1000 times more expensive than securing

[36]This figure is taken from the U.S. Bureau of Economic Analysis "Real Value Added by Industry" data, lines 56-57 ("Federal Reserve banks, credit intermediation, and related activities" and "Securities, commodity contracts, and investments.") Real value added measures payments to both capital and labor. See Philippon (2015) on why it is a useful measure for the cost of the financial sector.

[37]Gennaioli, Shleifer and Vishny (2015) distinguish between financial sector trust in the sense of security from expropriation or theft and trust in the sense of confidence to take risks. They argue that high fees and market power in the financial sector, especially as relates to investment management (see Greenwood and Scharfstein, 2013), can often be intrepreted as demand for the latter kind of trust, confidence to take risks. The former kind of trust, security from theft, is the aspect I have in mind here as relevant for the comparison to Nakamoto trust.

[38]For example, Hu, Pan and Wang (2021) report that the median fee charged in the treasury repo market is 2 basis points (0.02%) on an annual basis which translates to about 0.00005% per day. Interactive Brokers' fees for large foreign exchange transactions are about 0.001%. Budish, Lee and Shim (2024) find that the average exchange trading fee in the U.S. stock market is $0.0001 per-share-per-side, or about 0.0001% on a $100 share of stock. Fees are higher for retail transactions but still often relatively small. Visa's annual operating expenses are less than 0.1% of their annual transaction volume and their revenue is about 0.25% of volume (Visa Annual Report, 2022). Asset management fees for hundreds of Vanguard index funds are less than 0.10% (https://investor.vanguard.com/investment-products/list/all). There are of course numerous other fees in finance that are much higher, but these tend not to be fees for trust (in the sense of security from theft, see previous fn.) but fees that reflect market power over consumers (Campbell, 2006, Greenwood and Scharfstein, 2013) or consumers' willingness to pay for financial advice (Gennaioli, Shleifer and Vishny, 2015). A famous paper of Philippon (2015) estimates the cost of the financial sector as a percentage of the value of real intermediation (as opposed to transaction volume) and finds that this is about 1.5-2%.

against a $1 million attack. This equilibrium constraint suggests that if cryptocurrencies were to become a more significant part of the global financial system than they have been to date, then their costs would have to grow to absurd levels (absent implicit support from rule of law). In the base case analysis considered in Section 4, it would take all of global GDP to secure the system against a $40 billion attack. Traditional trust, whether from rule of law, reputations, relationships, collateral, etc., is by no means perfect, but is a bargain relative to Nakamoto (see Section 6).

A conceptual insight of this paper for computer science is that the economic security of a permissionless consensus protocol should be thought of not as a 0-1 variable that breaks at a threshold ρ, as in the classic distributed consensus literature (e.g., Lamport et al. (1982), Dwork et al. (1988)), but as an incentive compatibility constraint.

Nakamoto trust would be a lot more economically attractive if an attacker lost the stock value of their capital in addition to paying the flow cost of attack, as considered in Section 5. However, this requires either (i) that security rests on the possibility of outright collapse, or (ii) that the blockchain is ultimately reliant on an external source of trust support if there is a large-enough attacker, such as rule of law.

It bears emphasis that the paper's analysis is consistent with the continued use of cryptocurrencies and blockchains for black-market purposes, and more generally in use cases where users are willing to pay the high implicit costs of anonymous, decentralized trust.

This paper's analysis is also consistent with the usefulness of the blockchain data structure *without* Nakamoto (2008)'s novel form of trust. This is often called distributed ledger technology or a permissioned blockchain (see fn. 2 and Section 2.5.1). Indeed, what this paper highlights is that it is exactly the aspect of Nakamoto (2008) that is so innovative relative to these kinds of distributed databases—the anonymous, decentralized trust—that is the source of its economic limits. As one specific example, it is completely consistent with this paper's analysis that Central Bank Digital Currencies (CBDCs) could be of high economic value. CBDCs take some technical inspiration from cryptocurrencies but are anchored in traditional trust from rule of law and the reputation of central banks, and thus do not face the scaling problem of Nakamoto trust highlighted in this paper.

At a broader level, this paper builds on the view tracing all the way back to Adam Smith that government and laws are essential ingredients for the market system. A central point this paper has tried to emphasize is a fairly simple one implicit in Hayek (1960) and Becker (1968), though I have not seen it stated explicitly in this language, which is that traditional trust supported by rule of law enjoys economies of scale. Society pays the fixed cost of the apparatus of rule of law, or firms pay the fixed cost of building a brand or reputation or holding collateral (each of which works in conjunction with laws), and these fixed cost assets can provide trust over a large number

of economic activities at low or zero marginal cost.

Directions for Future Research. I highlight two directions for future research given the message of this paper. First, and most directly, is there a "solution" to this paper's critique of Nakamoto trust? Informally, is there a way to generate trust in a public dataset (or, specifically, a cryptocurrency) that has some of the anonymity and decentralization aspects of Nakamoto (2008) while being signficantly less economically constrained by the arguments in this paper?[39] Appendix A describes several of the responses this paper has received since it first circulated in 2018. The most promising responses combine blockchain-based trust with traditional trust in some way. For example, Ethereum's new protocol algorithmically mimics the combination of collateral plus rule of law to punish sub-$\frac{2}{3}$ attackers but must rely on some external source of trust support to punish large-enough attackers. If large sums of money were in dispute, it seems likely that this external source of trust support would involve rule of law. Another interesting response is to concede that blockchain trust is intrinsically very expensive, per this paper's argument, but to only use it for occasional large transactions with long escrow periods ("Layer 1"), while most transactions are conducted off chain ("Layer 2"), supported by external sources of trust. The idea of Bitcoin as a "digital gold" held by traditional institutional investors also fits this paradigm. Bitcoin Exchange Traded Funds (ETFs) have over $1bn of volume per trading day, traded on traditional regulated stock exchanges and managed by institutions like Blackrock and Fidelity. An open conceptual question about these responses is what the permissionless consensus part adds given the ultimate reliance on external sources of trust support. Are these combinations of permissionless consensus and traditional sources of trust superior to existing alternatives? Does permissionless consensus expand the production possibilities frontier for trust?

The second direction for research is a broader conceptual question: How should economists and computer scientists model trust that comes from a combination of technology and rule of law? More generally, how should researchers understand trust when it comes from multiple sources in the same transaction that work in complement with each other?[40] This is often the case in practice, with trust arising from some combination of rule of law, reputations, relationships, brands, collateral, norms, technology, etc., often implicitly and without drawing notice. Consider the completely ordinary transaction of buying a cup of coffee at the local coffee shop. The consumer trusts the

[39]Recent efforts to formalize this question include Leshno, Pass and Shi (2023) and Budish, Lewis-Pye and Roughgarden (2024).

[40]One simple preliminary exploration of this question is in Section 4.1 of Budish and Sunderam (2023) who conceptualize trust as getting to cooperate-cooperate in a prisoner's dilemma (as discussed in La Porta et al., 1997), and model (i) technology as eliminating some actions from the possibility set, (ii) law as changing the payoffs to some actions via punishment, and (iii) reputation as the differential incentive to cooperate if play is repeated versus one-shot (as in the traditional folk theorem arguments of Aumann, 1959, Fudenberg, Levine and Maskin, 1994 and others).

coffee shop to provide quality coffee because of reputational incentives, and perhaps implicitly food-safety laws. The coffee shop trusts the consumer's payment if cash because counterfeiting is technologically complex and illegal, and if electronic because of traditional cryptography and because the financial intermediary has reputational, relational and legal reason to follow through. The employee trusts their employer to follow through with promised compensation because of laws and the implicit relational contract. Both the customer and the employee trust the other not to rob them because of laws and social norms. All this trust for a cup of coffee! These multiple layers of trust that work together for even the most ordinary of economic transactions are likely part of what makes the traditional market system so robust and, dare I say, beautiful, but also part of what makes trust so hard to model satisfactorily.

References

Auer, Raphael. 2019. "Beyond the Doomsday Economics of 'Proof-of-Work' in Cryptocurrencies." *BIS Working Paper No. 765*.

Aumann, R.J. 1959. "Acceptable Points in General Cooperative n-Person Games." In *Contributions to the Theory of Games IV, Annals of Mathematics Study 40*. 287–324. Princeton University Press.

Bailey, Norman T.J. 1957. "Some Further Results in the Non-Equilibrium Theory of a Simple Queue." *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 19(2): 326–333.

Baker, George, Robert Gibbons, and Kevin J. Murphy. 2002. "Relational Contracts and the Theory of the Firm." *The Quarterly Journal of Economics*, 117(1): 39–84.

Bakos, Yannis, and Hanna Halaburda. 2023. "Tradeoffs in Permissioned vs Permissionless Blockchains: Trust and Performance." *NYU Stern School of Business Working Paper*. Available at SSRN: `https://ssrn.com/abstract=3789425`.

Bayer, Dave, Stuart Haber, and W. Scott Stornetta. 1993. "Improving the Efficiency and Reliability of Digital Time-Stamping." In *Sequences II: Methods in Communication, Security and Computer Science*. 329–334. Springer.

Becker, Gary S. 1968. "Crime and Punishment: An Economic Approach." *Journal of Political Economy*, 76(2): 169–217.

Biais, Bruno, Christophe Bisiere, Matthieu Bouvard, and Catherine Casamatta. 2019. "The Blockchain Folk Theorem." *The Review of Financial Studies*, 32(5): 1662–1715.

Bitcoin Magazine. 2022. "Peter Thiel - Bitcoin Keynote - Bitcoin 2022 Conference." Last modified April 7, 2022. Retrieved May 1, 2022 from `https://www.youtube.com/watch?v=ko6K82pXcPA`.

Bitcoin Wiki. 2020. "Weaknesses → Probably Not a Problem → Attacker Has A Lot of Computing Power." Last modified June 27, 2020. Retrieved April 22, 2022 from `https://en.bitcoin.it/wiki/Weaknesses#Attacker_has_a_lot_of_computing_power`.

Böhme, Rainer, Nicolas Christin, Benjamin Edelman, and Tyler Moore. 2015. "Bitcoin: Economics, Technology, and Governance." *Journal of Economic Perspectives*, 29(2): 213–238.

Bonneau, Joseph. 2016. "Why Buy When You Can Rent? Bribery Attacks on Bitcoin Consensus." In *Proceedings of the 20th International Conference on Financial Cryptography and Data Security (FC 2016)*. 19–26. Springer.

Budish, Eric. 2018. "The Economic Limits of Bitcoin and the Blockchain." NBER Working Paper No. 24717.

Budish, Eric, and Adi Sunderam. 2023. "Blockchain Technology and Stablecoins in Traditional Finance." In *Sveriges Riksbank 7th Annual Macroprudential Conference*.

Budish, Eric, Andrew Lewis-Pye, and Tim Roughgarden. 2024. "The Economic Limits of Permissionless Consensus." In *Proceedings of the 25th ACM Conference on Economics and Computation (EC '24)*. Forthcoming. Available at `https://arxiv.org/abs/2405.09173`.

Budish, Eric, Robin S. Lee, and John J. Shim. 2024. "A Theory of Stock Exchange Competition and Innovation: Will the Market Fix the Market?" *Journal of Political Economy*, 132(4): 1209–1246.

Buterin, Vitalik. 2014*a*. "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform." (White Paper).

Buterin, Vitalik. 2014*b*. "Slasher: A Punitive Proof-of-Stake Algorithm." Last modified January 15, 2014. Retrieved May 5, 2022 from `https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/`.

Buterin, Vitalik. 2016. "A Proof of Stake Design Philosophy." *Medium*, Last Modified December 30, 2016. Retrieved May 1, 2022 from `https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51`.

Buterin, Vitalik. 2017. "Minimal Slashing Conditions." Last Modified March 2, 2017. Retrieved Nov 7, 2023 from `https://medium.com/@VitalikButerin/minimal-slashing-conditions-20f0b500fc6c`.

Buterin, Vitalik. 2022. "What in the Ethereum Application Ecosystem Excites Me." Last Modified December 5, 2022. Retrieved Sep 26, 2023 from `https://vitalik.eth.limo/general/2022/12/05/excited.html`.

Buterin, Vitalik, and Virgil Griffith. 2019. "Casper the Friendly Finality Gadget." *arXiv preprint arXiv:1710.09437*.

Campbell, John Y. 2006. "Household Finance." *The Journal of Finance*, 61(4): 1553–1604.

Carlsten, Miles, Harry Kalodner, S. Matthew Weinberg, and Arvind Narayanan. 2016. "On the Instability of Bitcoin Without the Block Reward." In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 154–167.

Chiu, Jonathan, and Thorsten V. Koeppl. 2022. "The Economics of Cryptocurrency: Bitcoin and Beyond." *Canadian Journal of Economics*, 55(4): 1762–1798.

Cochrane, John H. 2013. "Finance: Function Matters, Not Size." *Journal of Economic Perspectives*, 27(2): 29–50.

Cong, Lin William, Zhiguo He, and Jiasun Li. 2021. "Decentralized Mining in Centralized Pools." *The Review of Financial Studies*, 34(3): 1191–1235.

Cox, Jeff. 2021. "Yellen Sounds Warning About 'Extremely Inefficient' Bitcoin." *CNBC*. Last modified February 22, 2021. Retrieved May 1, 2022 from `https://www.cnbc.com/2021/02/22/yellen-sounds-warning-about-extremely-inefficient-bitcoin.html`.

De Vries, Alex. 2018. "Bitcoin's Growing Energy Problem." *Joule*, 2(5): 801–805.

Digiconomist. 2022. "Bitcoin Energy Consumption Index." Retrieved May 19, 2022 from `https://digiconomist.net/bitcoin-energy-consumption`.

Dolev, D., and H. R. Strong. 1983. "Authenticated Algorithms for Byzantine Agreement." *SIAM Journal on Computing*, 12(4): 656–666.

Dwork, Cynthia, Nancy Lynch, and Larry Stockmeyer. 1988. "Consensus in the Presence of Partial Synchrony." *Journal of the ACM*, 35(2): 288–323.

Easley, David, Maureen O'Hara, and Soumya Basu. 2019. "From Mining to Markets: The Evolution of Bitcoin Transaction Fees." *Journal of Financial Economics*, 134(1): 91–109.

Edgington, Ben. 2023. "Inactivity Leak." Available at `https://eth2book.info/capella/part2/incentives/inactivity/`.

Ethereum.org. 2023. "Ethereum Proof-of-Stake Attack and Defense." Last Modified Aug 15, 2023. Retrieved July 3, 2024 from `https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/attack-and-defense/`.

Eyal, Ittay, and Emin Gun Sirer. 2014. "Majority is not Enough: Bitcoin Mining is Vulnerable." In *Proceedings of the 18th International Conference on Financial Cryptography and Data Security (FC 2014)*. 436–454. Springer.

Fischer, M., N. Lynch, and M. Paterson. 1985. "Impossibility of Distributed Consensus with One Faulty Process." *Journal of the ACM*, 32(2): 374–382.

Foley, Sean, Jonathan R. Karlsen, and Tālis J. Putniņš. 2019. "Sex, Drugs, and Bitcoin: How Much Illegal Activity is Financed through Cryptocurrencies?" *The Review of Financial Studies*, 32(5): 1798–1853.

Friedman, Milton. 1962. *Capitalism and Freedom.* The University of Chicago Press.

Fudenberg, Drew, David Levine, and Eric Maskin. 1994. "The Folk Theorem with Imperfect Public Information." *Econometrica*, 62(5).

Gans, Joshua S., and Hanna Halaburda. 2023. "'Zero Cost' Majority Attacks on Permissionless Blockchains." Available at `https://ssrn.com/abstract=4505460`.

Gennaioli, Nicola, Andrei Shleifer, and Robert Vishny. 2015. "Money Doctors." *The Journal of Finance*, 70(1): 91–114.

Gensler, Gary. 2021. "Remarks Before the Aspen Security Forum." Last modified August 3, 2021. Retrieved May 1, 2022 from `https://www.sec.gov/news/public-statement/gensler-aspen-security-forum-2021-08-03`.

Goldman Sachs. 2018. "Blockchain - The New Technology of Trust." Retrieved April 11, 2018, from `http://www.goldmansachs.com/our-thinking/pages/blockchain/`.

Greenwood, Robin, and David Scharfstein. 2013. "The Growth of Finance." *Journal of Economic Perspectives*, 27(2): 3–28.

Grunspan, Cyril, and Ricardo Pérez-Marco. 2018. "Double Spend Races." *International Journal of Theoretical and Applied Finance*, 21(08): 1850053.

Grunspan, Cyril, and Ricardo Pérez-Marco. 2022. "On Profitability of Nakamoto Double Spend." *Probability in the Engineering and Informational Sciences*, 36(3): 732–746.

Guiso, Luigi, Paola Sapienza, and Luigi Zingales. 2006. "Does Culture Affect Economic Outcomes?" *Journal of Economic Perspectives*, 20(2): 23–48.

Haber, Stuart, and W. Scott Stornetta. 1991. "How to Time-Stamp a Digital Document." *Journal of Cryptography*, 3(2): 99–111.

Halaburda, Hanna, Guillaume Haeringer, Joshua S. Gans, and Neil Gandal. 2022. "The Microeconomics of Cryptocurrencies." *Journal of Economic Literature*, 60(3): 971–1013.

Hart, Oliver. 1995. *Firms, Contracts, and Financial Structure.* Clarendon Press.

Hayek, Friedrich A. 1960. *The Constitution of Liberty.* The University of Chicago Press.

Holmstrom, Bengt, and Paul Milgrom. 1994. "The Firm as an Incentive System." *The American Economic Review*, 84(4): 972–991.

Huberman, Gur, Jacob D. Leshno, and Ciamac Moallemi. 2021. "Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System." *The Review of Economic Studies*, 88(6): 3011–3040.

Hu, Grace Xing, Jun Pan, and Jiang Wang. 2021. "Tri-Party Repo Pricing." *Journal of Financial and Quantitative Analysis*, 56(1): 337–371.

Kandori, Michihiro. 1992. "Social Norms and Community Enforcement." *The Review of Economic Studies*, 59(1): 63–80.

Klein, Ezra. 2022. "Ethereum's Founder on What Crypto Can – and Can't – Do." Audio podcast episode. Available at `https://www.nytimes.com/2022/09/30/podcasts/transcript-ezra-klein-interviews-vitalik-buterin.html`.

Kreps, David M., Paul Milgrom, John Roberts, and Robert Wilson. 1982. "Rational Cooperation in the Finitely Repeated Prisoners' Dilemma." *Journal of Economic Theory*, 27(2): 245–252.

Kroll, Joshua A., Ian C. Davey, and Edward W. Felten. 2013. "The Economics of Bitcoin Mining or, Bitcoin in the Presence of Adversaries." In *12th Workshop on the Economics of Information Security.*

Krugman, Paul. 1998. "Baby-Sitting the Economy." *Slate.* Last modified Aug 14, 1998. Retrieved May 1, 2022 from `https://slate.com/business/1998/08/baby-sitting-the-economy.html`.

Lamport, Leslie, Robert Shostak, and Marshall Pease. 1982. "The Byzantine Generals Problem." *ACM Transactions on Programming Languages and Systems*, 4(3): 382–401.

La Porta, Rafael, Florencio Lopez-de-Silanes, Andrei Shleifer, and Robert W. Vishny. 1997. "Trust in Large Organizations." *American Economic Review Papers and Proceedings*, 87(2): 333–338.

La Porta, Rafael, Florencio Lopez-de-Silanes, Andrei Shleifer, and Robert W. Vishny. 1998. "Law and Finance." *Journal of Political Economy*, 106(6): 1113–1155.

Leshno, Jacob, Rafael Pass, and Elaine Shi. 2023. "Can open decentralized ledgers be economically secure?" Cryptology ePrint Archive Working Paper 2023/1516, Available at https://eprint.iacr.org/2023/1516.

Levine, Matt. 2017. "Bank Blockchains and an Alibaba Box." *Bloomberg View*, Last modified January 10, 2017. Retrieved May 24, 2022 from https://www.bloomberg.com/view/articles/2017-01-10/bank-blockchains-and-an-alibaba-box.

Levin, Jonathan. 2003. "Relational Incentive Contracts." *American Economic Review*, 93(3): 835–857.

Lewis-Pye, Andrew, and Tim Roughgarden. 2024. "Permissionless Consensus." *arXiv preprint arXiv:2304.14701.*

Ma, June, Joshua S. Gans, and Rabee Tourky. 2019. "Market Structure in Bitcoin Mining." Rotman School of Management Working Paper No. 3103104, Available at SSRN: https://ssrn.com/abstract=3103104.

Makarov, Igor, and Antoinette Schoar. 2021. "Blockchain Analysis of the Bitcoin Market." NBER Working Paper No. 29396.

Modigliani, Franco, and Merton H. Miller. 1958. "The Cost of Capital, Corporation Finance and the Theory of Investment." *The American Economic Review*, 48(3): 261–297.

Moroz, Daniel J., Daniel J. Aronoff, Neha Narula, and David C. Parkes. 2020. "Double-Spend Counterattacks: Threat of Retaliation in Proof-of-Work Systems." *arXiv preprint arXiv:2002.10736.*

Nakamoto, Satoshi. 2008. "Bitcoin: A Peer-to-Peer Electronic Cash System." (White Paper).

Nelson, Phillip. 1974. "Advertising as Information." *Journal of Political Economy*, 82(4): 729–754.

Pease, M., R. Shostak, and L. Lamport. 1980. "Reaching Agreement in the Presence of Faults." *Journal of the ACM*, 27(2): 228–234.

Philippon, Thomas. 2015. "Has the US Finance Industry Become Less Efficient? On the Theory and Measurement of Financial Intermediation." *American Economic Review*, 105(4): 1408–1438.

Popper, Nathaniel. 2015. *Digital Gold: Bitcoin and the Inside Story of the Misfits and Millionaires Trying to Reinvent Money.* Harper Paperbacks.

Prat, Julien, and Benjamin Walter. 2021. "An Equilibrium Model of the Market for Bitcoin Mining." *Journal of Political Economy*, 129(8): 2415–2452.

Rogoff, Kenneth. 2017. *The Curse of Cash.* Princeton University Press.

Rosenfeld, Meni. 2014. "Analysis of Hashrate-Based Double-Spending." *arXiv preprint arXiv:1402.2009*.

Roughgarden, Tim. 2023. "Online Course: Foundations of Blockchains." Retrieved Sep 23, 2024 from `https://www.youtube.com/playlist?list=PLEGCF-WLh2RLOHv_xUGLqRts_9JxrckiA`.

Saleh, Fahad. 2021. "Blockchain Without Waste: Proof-of-Stake." *The Review of Financial Studies*, 34: 1156–1190.

Schelling, Thomas C. 1956. "An Essay on Bargaining." *American Economic Review*, 46(3): 281–306.

Schelling, Thomas C. 1960. *The Strategy of Conflict.* Harvard University Press.

Shleifer, Andrei, and Robert W. Vishny. 1997. "A Survey of Corporate Governance." *The Journal of Finance*, 52(2): 737–783.

Smith, Adam. 1776. *The Wealth of Nations.* Penguin Classics; First Edition (March 25, 1982).

Tabarrok, Alex. 2019. "Bitcoin is Less Secure than Most People Think." Last Modified January 7, 2019. Retrieved Sep 26, 2023 from `https://marginalrevolution.com/marginalrevolution/2019/01/bitcoin-much-less-secure-people-think.html`.

Tadelis, Steven. 1999. "What's in a Name? Reputation as a Tradeable Asset." *American Economic Review*, 89(3): 548–563.

Tas, Ertem Nusret, David Tse, Fangyu Gai, Sreeram Kannan, Mohammad Ali Maddah-Ali, and Fisher Yu. 2023. "Bitcoin-Enhanced Proof-of-Stake Security: Possibilities and Impossibilities." *arXiv preprint arXiv:2207.08392*.

Urban Institute. 2023. "Criminal Justice Expenditures: Police, Corrections, and Courts." Retrieved Sep 26, 2023 from `https://www.urban.org/policy-centers/cross-center-initiatives/state-and-local-finance-initiative/state-and-local-backgrounders/criminal-justice-police-corrections-courts-expenditures`.

Visa. 2021. "Visa Annual Report." Last modified Nov 18, 2021. Retrieved May 1, 2022 from https://s29.q4cdn.com/385744025/files/doc_downloads/Visa-Inc_-Fiscal-2021-Annual-Report.pdf.

Visa. 2022. "Visa Inc. Fiscal 2022 Annual Report." Retrieved Sep 22, 2023 from https://s29.q4cdn.com/385744025/files/doc_downloads/2022/Visa-Inc-Fiscal-2022-Annual-Report.pdf.

Wolitzky, Alexander. 2022. "Cooperation in Large Societies." In *Advances in Economics and Econometrics: Twelfth World Congress*. Forthcoming. Available at https://economics.mit.edu/sites/default/files/2023-04/ESWC%20chapter%20July%202022_1.pdf.

Appendix

A Discussion of Responses to this Paper's Argument

This paper first circulated in shorter form in June 2018. I received a lot of comments and counter-arguments in response to the paper's main line of argument.

I have tried to handle the central line of counter-argument throughout the main text of this updated draft. This is the point made by Huberman, Leshno and Moallemi (2021) and many practitioners that we should compare Nakamoto's costs to the costs of market power in traditional finance, which are also high.[41] I hope the present draft of the text makes more clear the conditional nature of the paper's argument: if Nakamoto trust becomes more economically useful, then it will also have to get more expensive, linearly, or it will be vulnerable to attack. I hope as well that the more explicit computational simulations, for varying levels of V_{attack} all the way up to $100 billion, as well as the analysis in Section 6, make clear that the way Nakamoto's security cost model scales is importantly different from how costs scale for traditional finance protected by rule-of-law.

I have also tried to address the main technical innovation that may seem to address my argument in the main text, which is the idea of proof-of-stake with slashing. This approach is economically attractive because it mimics the traditional trust model of collateral plus rule-of-law, so the cost of attack is not the flow cost of trust-support but the stock value of the attacker's collateral. This is not quite as cost-effective as the use of collateral in the traditional financial system backed by rule-of-law, which costs zero under the assumptions of the Modigliani-Miller theorem, but it is a 2500x improvment over the costs of Bitcoin's security model as shown in Section 5. Unfortunately, as discussed in the main text, impossibility theorems of Tas et al. (2023) and Budish, Lewis-Pye and Roughgarden (2024) show that proof-of-stake with slashing faces technical limitations if the attacker is large enough or if the communications network is not sufficiently reliable. The rough intuition is that a large enough attacker can thwart the protocol's legal system. Thus, the present paper's model is a lens through which we can understand the economic goals of Ethereum's adoption of proof-of-stake with slashing, while the related computer science research helps us understand the conditions under which it works and its technical limitations.

In the remainder of this appendix I discuss several of the other most common comments and counter-arguments I have received about this paper since it was first circulated.

[41] See Philippon (2015) and Greenwood and Scharfstein (2013) on high costs of traditional finance, and see Cochrane (2013) for a counterpoint.

A.1 Community Response

A majority attack on Bitcoin or any other major cryptocurrency would be widely noticed. A line of argument I heard frequently in response to the June 2018 draft is that the Bitcoin community would organize a response to the attack. For example, the community could organize a "hard fork" off of the state of the blockchain just prior to the attack, which would include all transactions perceived to be valid, void any perceived-as-invalid transactions, possibly confiscate or void the attacker's other Bitcoin holdings if these are traceable, and possibly change the hash function or find some other way to ignore or circumvent the attacker's majority of compute power.[42]

The community response argument seems valid as an argument that attacks might be more expensive or difficult to execute than is modeled in Sections 3-4, but it raises four important issues.

First, and most obviously, the argument contradicts the notion of anonymous, decentralized trust. It relies on a specific set of trusted individuals in the Bitcoin community.

Second, a hard fork harms honest holders of specialized capital too.

Third, after a hard fork, the blockchain goes from equilibrium constraint (10) to equilibrium constraint (3), so it is vulnerable to repeated attack. This point is made in Buterin's 2016 blog post quoted in the main text.

Fourth, consider the community response argument from the perspective of a traditional financial institution. In the event of a large-scale attack that involves billions of dollars, the traditional financial institution would, in this telling, be left in the hands of the Bitcoin community. At present, reliance on a tight-knit community of those most invested in Bitcoin (whether financially, intellectually, etc.), may sound reassuring—those with the most to lose would rally together to save it. But now imagine the hypothetical future in which Bitcoin becomes a more integral part of the global financial system, and imagine there is a fight over whether an entity like a Goldman Sachs is entitled to billions of dollars worth of Bitcoin that it believes was stolen—but the longest chain says otherwise. Will the "vampire squid" be made whole by the "Bitcoin community?" Quite possibly, but one can hopefully see the potential weakness of relying on an amorphous community as a source of trust for the global economic and financial system.

A.2 Rule of Law

A related line of argument I have heard frequently is that, in the event of a large-scale attack specifically on a financial institution such as a bank or exchange, rule of law would step in. For

[42]The phrase "hard fork" means that in addition to coordinating on a particular fork of a blockchain if there are multiple—in this case, the attacker's chain, which is the longest, and the chain the community is urging be coordinated on in response—the code used by miners is updated as well. This could include hard-coded state information such as the new chain or information about voided Bitcoins held by the attacker, code updates such as a new hash function, etc.

example, the financial institutions depicted as the victims of a double-spend in Figure 3, once they realize they no longer have the Bitcoins paid to them because of the attack, would obtain help from rule-of-law tracing down the attacker and recovering the stolen funds.

This response, too, seems internally valid while contradicting the idea of anonymous, decentralized trust. In this view, cryptocurrencies are mostly based on anonymous, decentralized trust — hence evading most forms of scrutiny by regulators and law enforcement — but, if there is a large attack, then rule of law will come to the rescue.

A.3 Counterattacks

Moroz et al. (2020) extend the analysis in Budish (2018) to enable the victim of a double-spending attack to attack back. They consider a game in which there is an Attacker and a Defender. If the Attacker double spends against the Defender for v dollars, the Defender can then retaliate, themselves organizing a 51% or more majority, to attack back so that the original honest chain becomes the longest chain again. This allows the Defender to recover their property.

For example, suppose the escrow period is 6, denote the initial double-spend transaction as taking place in block 1, and suppose the attacker chain replaces the honest chain as soon as the escrow period elapses, as in Figure 3. Notationally, suppose the honest chain consists of blocks {1, 2, ..., 7} at the time the honest chain is replaced, and the attacker chain that replaces it is {1', 2', ..., 7', 8'}. If the Defender can quickly organize a majority of their own, then they can build off of the {1, 2, ..., 7} chain, and eventually surpass the attacker chain, recovering their property. For example, maybe the honest chain reaches block 10 before the Attacker chain reaches block 10', so then {1, 2, ..., 10} is the new longest chain and the Defender has their property back from the correct transaction in block 1.

This argument is game theoretically valid, and indeed there are theoretical subtleties to the argument that the reader can appreciate for themselves in the paper. That said, it relies on every large-scale participant in the Bitcoin system being able and willing to conduct a 51% attack on a moment's notice.

A.4 Modification to Nakamoto I: Increase Throughput

Bitcoin processes about 2000 transactions per block, which is about 288,000 per day or 105 million per year. In contrast, Visa processes about 165 billion transactions per year (Visa, 2021).

The reader will notice that the logic in equations (1)-(3) does not depend directly on the number of transactions in a block. If the number of transactions in a Bitcoin block were to increase by 1000x (to roughly Visa's level), then the required p_{block} to keep Bitcoin secure against a given scale

of attack V_{attack}, per equation (3) would not change. Thus, the required cost *per transaction* to keep Bitcoin secure against a given scale of attack would decline by a factor of 1000.

In this scenario of a 1000x throughput increase, Bitcoin's security costs per transaction are still large, but less astonishingly so. In the base case, to secure Bitcoin against a $1 billion attack would require costs per transaction of $25 instead of $25,000. To secure against a $100 billion attack would require costs per transaction of $2,500 instead of $2.5 million.

A subtlety is that as the number of transactions per block grows, so too might the scope for attack. That is, V_{attack} might grow as well.

Still, this seems a promising response to the logic of this paper. A particularly interesting variation on this idea is the paradigm called "Layer 2." In this paradigm, the Bitcoin blockchain ("Layer 1") would be used for relatively large transactions, but smaller transactions would be conducted off-chain, possibly supported with traditional forms of trust, with just occasional netting on the main Bitcoin blockchain. In this paradigm, as well, the large transactions on chain could also have a long escrow period, making attacks more expensive.[43]

A.5 Modification to Nakamoto II: Tweak Longest-Chain Rule

The discussion above in A.1 expressed skepticism about the "community" response to the logic of this paper. However, what about modifying the longest-chain rule to try to encode what the community would *want* to do in the event of an attack.

The modification to the longest-chain rule could take advantage of two specific features of double-spending attacks:

1. The Attacker has to sign transactions both to the victim of the double-spending attack—call this the Bank—and to another account they control—call this the Cousin account. The fact that there are multiple-signed transactions for the same funds is an initial proof that something suspicious has happened.
2. The Attacker has to make the signed transaction to the Bank public significantly before—in "real-world clock time"—the signed transaction to their Cousin account.

The difficulty with just using facts #1 and #2 to void the transaction to the Cousin is alluded to with the phrase "real-world clock time." Part of what the Nakamoto (2008) blockchain innovation accomplishes is a sequencing of data that does not rely on an external, trusted, time-stamping device.

Relatedly, the difficulty with just using fact #1 and having the policy "if there are multiple correctly signed transactions sending the same funds, destroy the funds" is that the victim of

[43] I thank Neha Narula for several helpful conversations about this approach.

the double-spending attack, the Bank, will by now have sent real-world financial assets to the Attacker—and this transaction, in the real world (off the blockchain), cannot be voided no matter how we modify the blockchain protocol. A different way to put the concern is that such a policy would allow any party that sends funds on the blockchain in exchange for goods or financial assets off the blockchain, to then void the counterparty's received funds after the fact. This seems a recipe for sabotage of the traditional financial sector.

The open question, then, is whether the protocol can be modified so that in the event of fact #1, multiple signed transactions, there is some way to appeal to fact #2, grounded in the sequencing of events in real-world clock time, not adjudicated by the longest-chain rule's determination of the sequence of events.

One pursuit along these lines is Leshno, Pass and Shi (2023). Their approach, which they call "Stubborn Nakamoto", is fully secure against double-spending attacks but, instead, has to permanently halt in response to observing conflicting transactions. In consensus terminology, it trades a security problem for a liveness problem. In conjunction with a source of external trust support, such as rule of law, to restart the system in case of such an outage, this could work. The open conceptual question then becomes what the permissionless consensus part adds given the source of external trust support (i.e., the same question asked in the Conclusion).

B Double-Spending Attack Technical Appendix

B.1 Proof of Proposition 3 (Closed-Form Expression for Duration of Double-Spending Attack)

Let $s = 0$ denote the time of the last block prior to the attack. As a reminder, time is normalized so that one unit of time is the amount it takes on average for honest miners to mine one block, e.g., 10 minutes for Bitcoin.

The attacker spends Bitcoins in exchange for other goods or assets in the honest miners' first block after time 0. In parallel, the attacker mines an alternative chain starting from the last block prior to the attack.

Honest miners mine blocks as a Poisson process with rate 1, and the attacker mines at rate $A > 1$. Both the honest miners' and the attacker's chains are time-independent Poisson processes, with:

$$B_H(s) \coloneqq \text{Number of blocks on honest chain at time } s,$$
$$B_A(s) \coloneqq \text{Number of blocks on attacker chain at time } s.$$

The attack is completed when both (i) the honest chain has mined at least $k+e$ blocks, therefore passing the attacker transactions' escrow periods, and (ii) the attacker chain has mined strictly more blocks than the honest chain. Therefore, the expected duration of the double-spending attack, as a function of the attacker majority A, escrow period e, and number of blocks in which the attacker places transactions k, is given by the stopping time formula:

$$t(A, e, k) = E[\inf\{s : B_H(s) \geq k + e,\, B_A(s) > B_H(s)\}].$$

It will be useful to define a random variable that denotes the time at which the honest chain completes the escrow period. Call this S_H^{k+e}:

$$S_H^{k+e} := \inf\{s : B_H(s) \geq k + e\}.$$

Similarly, it will be useful to define the difference in length between the honest chain and the attacker chain at the random time at which the honest chain completes the escrow period. Call this D^{k+e}:

$$\begin{aligned} D^{k+e} &:= B_H(S_H^{k+e}) - B_A(S_H^{k+e}) \\ &= (k + e) - B_A(S_H^{k+e}). \end{aligned}$$

If the realization of $D^{k+e} < 0$, the attacker chain is strictly longer than the honest chain at the conclusion of the escrow period, and the attacker immediately completes the double-spending attack. The total duration of attack is simply the time elapsed in completing the escrow period.

Else, if the realization of $D^{k+e} \geq 0$, the attacker faces a deficit and must continue the attack after the conclusion of the escrow period. In this case, the total duration of attack is the length of the escrow period plus the time it takes for the attacker to overcome the deficit. Note, if the attacker deficit is i blocks, to overcome the deficit the attacker must mine $i + 1$ more blocks than the honest miners, as the attacker chain must be strictly longer than the honest chain to complete the attack.

Hence, we can partition $t(A, e, k)$ based on the sign of D^{k+e} for a tractable expression for

$t(A, e, k)$:

$$\begin{aligned} t(A,e,k) &= E[S_H^{k+e} | D^{k+e} < 0)] \times P(D^{k+e} < 0) \\ &\quad + \sum_{i=0}^{k+e} \Big(E[S_H^{k+e} | D^{k+e} = i] + E[\text{Time for attacker to overcome deficit} = i] \Big) \times P(D^{k+e} = i) \\ &= E[S_H^{k+e}] + \sum_{i=0}^{k+e} E[\text{Time for attacker to overcome deficit} = i] \times P(D^{k+e} = i). \end{aligned}$$

The second equality follows from the law of total probability, $\sum_{l=-\infty}^{k+e} E[S_H^{k+e} | D^{k+e} = l] \times P(D^{k+e} = l) = E[S_H^{k+e}]$. Now, there are three terms left to simplify: $E[S_H^{k+e}]$, $E[$Time for attacker to overcome deficit $= i]$, and $P(D^{k+e} = i)$.

Consider the first term, $E[S_H^{k+e}]$. A well-known property of Poisson processes is that arrivals are distributed according to the Gamma distribution, $S_H^{k+e} \sim Gamma(k + e, 1)$. This Gamma distribution has a simple expression for its mean:

$$E[S_H^{k+e}] = k + e.$$

Now consider the second term, $E[$Time for attacker to overcome deficit $= i]$. Via the Markov property, we know this random variable does not depend on *when* the honest chain finishes the escrow period, only the deficit itself. So, consider the stochastic process:

$$\begin{aligned} D_{i+1}(s) &:= \overline{B_H}(s) - \overline{B_A}(s) \\ &= \text{Difference between (auxiliary)} \\ &\quad \text{honest and attacker chains at } s. \\ \overline{B_H}(0) &= i + 1 \\ \overline{B_A}(0) &= 0 \end{aligned}$$

That is, start two auxiliary honest and attacker chains at $s = 0$, but initialize the difference between the length of the two chains to be $i + 1$, as the attacker must overcome a deficit of i. The stochastic movement of this difference process can be thought of as an $M/M/1$ queue, where 'arrivals' are blocks on the honest chain, and 'departures' are blocks on the attacker's chain. We want the time it takes the difference process $D_{i+1}(s)$ to reach 0 – i.e., how long it takes the attacker to overcome the deficit i. In the queueing literature, this is known as the "first passage time" of a queue, $\text{FPT}(i + 1) \coloneqq \inf\{s : D_{i+1}(s) = 0\}$. The mean of the first passage time of the $M/M/1$

queue is $E[\text{FPT}(i+1)] = \frac{i+1}{A-1}$ (equation 41 in Bailey, 1957). Hence,

$$E[\text{Time for attacker to overcome deficit} = i] = \frac{i+1}{A-1}.$$

Finally, consider the term $P(D^{k+e} = i)$. Recall D^{k+e} is the difference between the honest and attacker's chains' length at the time the honest chain completes the escrow period. Hence, we can write:

$$\begin{aligned}\{D^{k+e} = i\} &= \left\{B_H(S_H^{k+e}) - B_A(S_H^{k+e}) = i\right\} \\ &= \left\{(k+e) - B_A(S_H^{k+e}) = i\right\} \\ &= \left\{B_A(S_H^{k+e}) = k+e-i\right\}.\end{aligned}$$

Thus, we want to find $P\left(B_A(S_H^{k+e}) = k+e-i\right)$. To proceed, we first find the probability $P(B_A(r) = l)$ for any realization r of the random escrow length S_H^{k+e} and any possible value of the attacker chain length l as of the time the honest chain completes the escrow period. Then, we will integrate over all possible realizations of r according to the probability distribution of S_H^{k+e}. The attacker's chain is $Poisson(A)$ and S_H^{k+e} is distributed $Gamma(k+e, 1)$, so that:

$$\begin{aligned}P(B_A(S_H^{k+e}) = l) &= \int_0^\infty P(B_A(r) = l \mid S_H^{k+e} = r) \cdot P(S_H^{k+e} = r)\, dr \\ &= \int_0^\infty \frac{(Ar)^l \cdot \exp(-Ar)}{l!} \cdot \frac{r^{k+e-1} \cdot \exp(-r)}{\Gamma(k+e)}\, dr \\ &= \frac{A^l}{l!\,(k+e-1)!} \cdot \frac{\Gamma(l+k+e-1)}{(1+A)^{l+k+e-1}} \int_0^\infty r \cdot \frac{(1+A)^{l+k+e-1} \cdot r^{l+k+e-2} \cdot \exp(-(1+A)r)}{\Gamma(l+k+e-1)}\, dr \\ &= \frac{(l+k+e-1)!}{l!\,(k+e-1)!} \left(\frac{A}{1+A}\right)^l \left(\frac{1}{1+A}\right)^{k+e}.\end{aligned}$$

The second equality exploits the independence of $B_A(s)$ and S_H^{k+e} (inherited from the independence of B_A and B_H) and substitutes the expressions for the respective Poisson and Gamma densities. The third equality moves terms out of the integral and multiplies and divides by $\frac{\Gamma(l+k+e-1)}{(1+A)^{l+k+e-1}}$, so that the integrand is exactly the first moment of $Gamma(l+k+e-1, 1+A)$. The expression for the mean is well known: $\frac{l+k+e-1}{1+A}$. The fourth equality substitutes this expression and simplifies. Hence, plugging in $l = k+e-i$, the probability of an attacker deficit i at the time the honest chain completes the escrow period is:

$$P(D^{k+e} = i) = \frac{(2(k+e)-1-i)!}{(k+e-i)!\,(k+e-1)!} \left(\frac{A}{1+A}\right)^{k+e-i} \left(\frac{1}{1+A}\right)^{k+e}.$$

Substituting these three expressions into that of $t(A, e, k)$, we have

$$t(A,e,k) = (k+e) + \left[\sum_{i=0}^{k+e} \left(\frac{i+1}{A-1}\right) \cdot \frac{(2(k+e)-1-i)!}{(k+e-i)!\,(k+e-1)!} \left(\frac{A}{1+A}\right)^{k+e-i} \left(\frac{1}{1+A}\right)^{k+e}\right]$$

obtaining expression (6) in the text as required.

To complete the proof let us consider the limits as $A \to_+ 1$ and $A \to \infty$. Define $f(A, e, k)$ as the bracketed expression above,

$$f(A,e,k) \equiv \left[\sum_{i=0}^{k+e} \left(\frac{i+1}{A-1}\right) \cdot \frac{(2(k+e)-1-i)!}{(k+e-i)!\,(k+e-1)!} \left(\frac{A}{1+A}\right)^{k+e-i} \left(\frac{1}{1+A}\right)^{k+e}\right],$$

such that $t(A, e, k)$ takes the form:

$$t(A,e,k) \equiv (k+e) + f(A,e,k).$$

First, consider the limit $\lim_{A\to\infty} t(A, e, k)$. Observe that each term in $f(A, e, k)$ either goes to 0 or is bounded by a constant. The first and fourth terms go to 0 in the limit: $0 \leq \lim_{A\to\infty} \left(\frac{i+1}{A-1}\right) \leq \lim_{A\to\infty} \left(\frac{k+e+1}{A-1}\right) = 0$ and $\lim_{A\to\infty} \left(\frac{1}{1+A}\right)^{k+e} = 0$. The second and third terms are bounded by a constant: $\frac{(2(k+e)-1-i)!}{(k+e-i)!\,(k+e-1)!}$ is constant in A and $\lim_{A\to\infty} \left(\frac{A}{1+A}\right)^{k+e-i} \leq 1$. Hence, the product of these terms is 0 in the limit, so $\lim_{A\to\infty} t(A, e, k) = (k + e) + 0 = k + e$ as desired.

Second, consider the limit $\lim_{A\to_+1} t(A, e, k)$. The first term in $f(A, e, k)$ goes to ∞ in the limit while all other terms are strictly positive and bounded below. Formally, for the first term, $\lim_{A\to_+1} \left(\frac{i+1}{A-1}\right) \geq \lim_{A\to_+1} \left(\frac{1}{A-1}\right) = \infty$. For the other terms: $\frac{(2(k+e)-1-i)!}{(k+e-i)!\,(k+e-1)!} > 0$ is constant in A; $\lim_{A\to_+1} \left(\frac{A}{1+A}\right)^{k+e-i} = \left(\frac{1}{2}\right)^{k+e-i} > 0$; and $\lim_{A\to_+1} \left(\frac{1}{1+A}\right)^{k+e} = \left(\frac{1}{2}\right)^{k+e} > 0$. Hence, the product of these terms goes to infinity in the limit, so $\lim_{A\to_+1} t(A, e, k) = \infty$ as desired.

B.2 Numerical Analysis of Cost-Minimizing Attacker Majority

The gross cost of attack, for an attacker with majority $A > 1$ and an attack that takes t time in expectation, is defined as $At \cdot N^*c$. Proposition 3 provides an explicit formula for $t(A, e, k)$, the expected duration of a double-spending attack as a function of the attacker majority A, the escrow period e, and the number of blocks in which the attacker places transactions k.

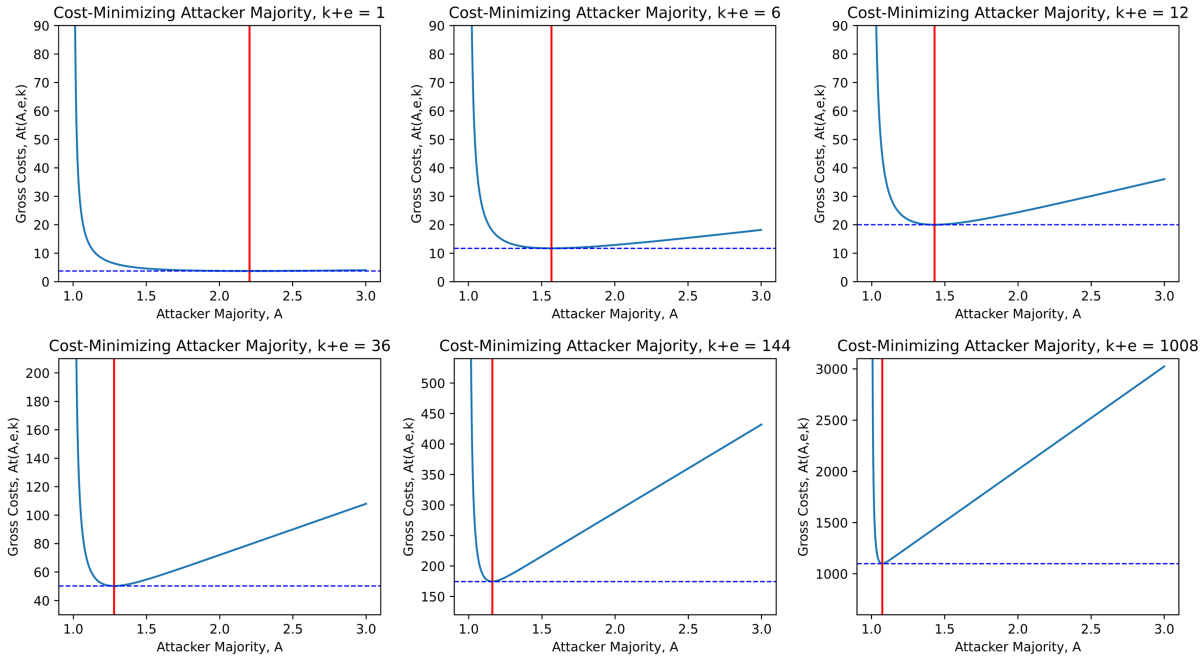
In this section of the Appendix, we use this definition and formula to numerically study the cost-minimizing attacker majority A as a function of $k + e$.

Formally, the gross-cost-minimization problem is given by:

$$
\begin{aligned}
A^*(k+e) &:= \arg\min_A A \cdot t(A,e,k) \\
&= \arg\min_A A \cdot \left[(k+e) + \sum_{i=0}^{k+e} \left[\left(\frac{i+1}{A-1} \right) \frac{(2(k+e)-1-i)!}{(k+e-i)!\,(k+e-1)!} \left(\frac{A}{1+A} \right)^{k+e-i} \left(\frac{1}{1+A} \right)^{k+e} \right] \right].
\end{aligned}
$$

While this minimization problem is not analytically tractable, it is straightforward to solve numerically. Figure 5 plots the cost of attack for a variety of values of $k+e$, as well as the cost-minimizing $A^*(k+e)$.

Figure 5: Attacker Gross Cost Minimization



Notes: The gross cost of attack as a function of majority A is in blue, plotted as $A \cdot t(A,e,k)$. As discussed in the main text, this quantity needs to be multiplied by equilibrium per-block trust-support costs N^*c to obtain gross costs in dollars. The gross-cost-minimizing attacker majority $A^*(k+e)$ is denoted in red, and is obtained via scipy.optimize.minimize_scalar, a numerical solver in Python.

Intuitively, an attacker majority that is too large will mine more blocks than is necessary for the attack to succeed, whereas an attacker majority that is too close to $A \approx 1$ will, as shown in Proposition 3, have an attack duration that converges to infinity, and hence also be more expensive than is optimal. Because the double-spending attack must be at least as long as the number of blocks double spend plus the escrow length, the cost-minimizing choice of $A^*(k+e)$ decreases as $k+e$ increases. The larger is $k+e$, the more sure a large majority is to mine more blocks than is necessary, by simple law-of-large numbers reasoning.

Table 7 provides the cost-minimizing majority $A^*(k+e)$, the duration of attack at this attacker majority $t\left(A^*(k+e), e, k\right)$, and the total gross cost of attack at this attacker majority $A^*(k+e) \cdot t\left(A^*(k+e), e, k\right)$ for a variety of values of $k+e$.

Table 7: Optimal Attacker Majority, Duration and Gross Costs

	# Blocks of Double Spending + Escrow Period $(k+e)$					
	1	6	12	36	144	1,008
$A^*(k+e)$	2.21	1.57	1.43	1.28	1.16	1.07
$t(A^*(k+e), e, k)$	1.70	7.47	13.99	39.28	150.21	1,023.37
$A^*(k+e) \cdot t(A^*(k+e), e, k)$	3.74	11.70	20.00	50.21	174.44	1,099.02

Notes: $A^*(k+e)$ is solved numerically as described in the text. The expected duration of attack then follows from Proposition 3. Gross costs are in units of equilibrium per-block trust-support costs N^*c.

As before, the gross cost of attack is given in units of per-block trust-support costs N^*c. Note that even very large escrow periods induce a cost-minimizing majority larger than 51%—for example, the case $k+e = 1008$ blocks (1 week) induces an optimal attacker majority of $A = 1.07$, or 51.7%.

C Selected 51% Attacks, Crypto Thefts and Crypto Collapses to Date

To date, there has not been a majority attack on the largest cryptocurrencies such as Bitcoin or Ethereum. There have been several majority attacks on smaller cryptocurrencies, including forks off of Bitcoin (Bitcoin Gold, Bitcoin SV, Bitcoin Cash ABC) and Ethereum (Ethereum Classic). A list of such attacks is provided as Appendix Table C.1. Of the attacks for which the amount stolen was reported, the largest such attack to date was for $18.6 million against Bitcoin Gold in May 2018. This amount represents about 74% of average daily transaction volume in the week prior to the attack. The longest such attack to date was against Ethereum Classic, reportedly 7,000 blocks or about a full day's worth of blocks at typical mining speeds for Ethereum Classic. For all of these attacks of forks off of Bitcoin and Ethereum, there was speculation in the crypto press that the attacker's motive was at least partly to sabotage the coin as opposed to stealing funds, but the details are thinly reported.

There have been many attacks on cryptocurrency financial entities that are based on exploiting flawed code, compromising private keys, manipulating prices of thinly traded tokens, or taking

temporary control of a project's governance. These are compiled as Appendix Table C.2. Many of these attacks have been for in excess of $100 million with several in excess of $500 million. The April 2022 attack on Beanstalk Farms for $182 million is interesting in the context of this paper because the attack vector involved the attacker borrowing funds (using what is known as a flash loan) to take temporary majority control of the token, which the attacker then used to vote for a resolution that drained the project's funds into accounts controlled by the attacker. The cost of borrowing majority control is a flow.

This paper's model analyzes collapse risk as a potential source of security (Section 5.2.1), albeit an unattractive one. While none of the major cryptocurrencies themselves have collapsed to date, many cryptocurrency projects and financial entities have indeed collapsed. A list is compiled as Appendix Table C.3.

Table C.1: 51% Attacks of Bitcoin and Ethereum Forks

Name of Coin	Date of Attack	Amount Stolen	Length of Largest Reorganization	Market Cap at Time of Attack	Market Cap at Peak
Bitcoin SV	8/3/2021	Unknown	Unknown	$2.7 billion	$8.3 billion
	6/24/2021	Unknown	Unknown	$2.4 billion	
Ethereum Classic	8/29/2020	Unknown	7,000 Blocks	$760 million	$15.6 billion
	8/6/2020	$1.7 million	4,200 Blocks	$840 million	
	8/1/2020	$5.6 million	3,700 Blocks	$860 million	
	1/5/2019	$1.1 million	140 Blocks	$560 million	
Bitcoin Gold	1/23/2020	$72 thousand	16 Blocks	$190 million	$7.6 billion
	5/16/2018	$18.6 million	22 Blocks	$1.0 billion	

Sources: Bitcoin Gold Forum, Bloomberg, CCN, Coinbase, CoinDesk, Cointelegraph, Decrypt, GitHub, Twitter, and https://dci.mit.edu/51-attacks. For a list of all articles consulted with URLs please see the author's website or the paper's online appendix. For interpreting length of longest reorganization, note that for Bitcoin SV and Bitcoin Gold there are typically 6 blocks per hour. For Ethereum Classic there are typically 275 blocks per hour. Amount Stolen is based on press reports of the dollar value stolen. Market cap data is from CoinMarketCap. The market cap at the time of attack is based on the day prior to the date of attack.

Table C.2: Attacks of Crypto Financial Entities

Name	Type of Business	Date of Attack	Amount Stolen	Attack Vector
Poloniex	Centralized Exchange	November 2023	$120 Million	Unknown
Mixin Network	Decentralized Exchange and Lending Protocol	September 2023	$200 Million	Compromised Cloud Database
Multichain	DeFi Bridge	July 2023	$230 Million	Compromised Private Keys or Rug Pull
Euler Finance	DeFi Lending Protocol	March 2023	$197 Million	Flash Loan Attack + Flawed Code
FTX	Centralized Exchange	November 2022	$477 Million	Compromised Private Keys
		March-April 2021	$600 Million	Price Manipulation
Mango Markets	Decentralized Exchange	October 2022	$100 Million	Price Manipulation
BNB Chain	DeFi Bridge	October 2022	$568 Million	Flawed Code
Wintermute	DeFi Market Maker	September 2022	$160 Million	Compromised Hot Wallet
Nomad	DeFi Bridge	August 2022	$190 Million	Flawed Code
Horizon Bridge	DeFi Bridge	June 2022	$100 Million	Compromised Private Keys + Governance Control
Beanstalk Farms	DeFi Stablecoin	April 2022	$182 Million	Flash Loan Attack + Governance Control
Ronin Network	DeFi Bridge	March 2022	$625 Million	Compromised Private Keys + Governance Control
Wormhole	DeFi Bridge	February 2022	$320 Million	Flawed Code
Qubit Finance	DeFi Lending Protocol	January 2022	$80 Million	Flawed Code
BitMart	Centralized Exchange	December 2021	$150 Million	Compromised Private Keys
C.R.E.A.M. Finance	DeFi Lending Protocol	October 2021	$130 Million	Flash Loan Attack + Price Manipulation
PolyNetwork	DeFi Bridge	August 2021	$600 Million	Flawed Code
KuCoin	Centralized Exchange	September 2020	$281 Million	Compromised Private Keys
BitGrail	Centralized Exchange	February 2018	$170 Million	Unknown
Coincheck	Centralized Exchange	January 2018	$530 Million	Unknown
The DAO	Decentralized Venture Capital	June 2016	$55 Million	Flawed Code
Mt. Gox	Centralized Exchange	2011-2014	$480 Million	Compromised Private Keys

Sources: BitMart, Bloomberg, Chainalysis, Coinbase, CoinDesk, CoinMarketCap, Cointelegraph, Elliptic Inc, Forbes, *Going Infinite* by Michael Lewis, Kraken Blog, Mango Markets, Medium Blog, Mixin Network, PolyNetwork, Reuters, The Verge, Twitter, Unchained Crypto, and WSJ. For a list of all articles consulted with URLs please see the author's website or the paper's online appendix. Amount Stolen is based on press reports of the dollar value stolen where available or based on press reports of the amount of cryptocurrency stolen converted into dollars using price data from CoinMarketCap.

Table C.3: Collapses of Crypto Financial Entities

Name	Type of Business	Date of Collapse	Entity Size
Genesis	Lending Firm	January 2023	$5.1 billion
BlockFi	Lending Firm	November 2022	$3.9 billion
FTX	Centralized Exchange	November 2022	$8.9 billion - $32 billion
Three Arrows Capital	Hedge Fund	June 2022	$3.4 billion - $10 billion
Voyager	Lending Firm	July 2022	$5.8 billion
Celsius	Lending Firm	July 2022	$5.5 billion - $19.1 billion
Terra + Luna	Blockchain + Stablecoin	May 2022	$40 billion
Africrypt	Investment Firm	April 2021	$3.6 billion
Thodex	Centralized Exchange	April 2021	$2 billion
Mt. Gox	Centralized Exchange	February 2014	$480 million

Sources: BlockFi Blog, Bloomberg, Chainalysis, CoinDesk, Cointelegraph, Forbes, PR Newswire, Reuters, WSJ, and bankruptcy filings. For Celsius: the $19.1 billion figure is Celsius's balance sheet size as of August 13, 2021 based on an investment memorandum provided to the author and reported in the Wall Street Journal in June 2022; the $5.5 billion figure is Celsius's balance sheet size when it filed for bankruptcy. For FTX: the $8.9 billion figure is FTX's customer assets at the time of its bankrupty filing; the $32 billion is its peak market valuation. For Three Arrows Capital: the $10 billion figure is based on reports of the hedge funds size as of March 2022 and the $3.4 billion figure is based on reports of claims from creditors. For Mt. Gox: the $480 million figure is the reported value of customer Bitcoins reported lost by the exchange when it filed for bankruptcy. For a list of all articles consulted with URLs please see the author's website or the paper's online appendix.