The Economic Limits of Bitcoin and Anonymous, Decentralized Trust on the Blockchain[*][†]

Eric Budish[‡]

June 6, 2022

Abstract

Satoshi Nakamoto invented a new form of trust. This paper presents a three equation argument that Nakamoto's new form of trust, while undeniably ingenious, is extremely expensive: the recurring, "flow" payments to the anonymous, decentralized compute power that maintains the trust must be large relative to the one-off, "stock" benefits of attacking the trust. This result also implies that the cost of securing the trust grows linearly with the potential value of attack — e.g., securing against a $1 billion attack is 1000 times more expensive than securing against a $1 million attack. A way out of this flow-stock argument is if both (i) the compute power used to maintain the trust is non-repurposable (as has been true for Bitcoin since mid-2013), and (ii) a successful attack would cause the economic value of the trust to collapse. However, vulnerability to economic collapse is itself a serious problem, and the model points to specific collapse scenarios. The analysis thus suggests a "pick your poison" economic critique of Bitcoin and its novel form of trust: it is either extremely expensive relative to its economic usefulness or vulnerable to sabotage and collapse.

[*] This paper originally circulated in June 2018 in a shorter form as Budish (2018). The present draft addresses a small bug in the attack math (see footnote 11), expands the analysis of double-spending attacks, sabotage attacks and collapse scenarios, and expands the discussion of the relevant computer science, institutional details and related literature to clarify the paper's arguments.

[†] Acknowledgments: thanks are due to Susan Athey, Vitalik Buterin, Glenn Ellison, Alex Frankel, Joshua Gans, Edward Glaeser, Austan Goolsbee, Hanna Halaburda, Zhiguo He, Joi Ito, Steve Kaplan, Anil Kashyap, Judd Kessler, Scott Kominers, Randall Kroszner, Robin Lee, Jacob Leshno, Neale Mahoney, Gregor Matvos, Sendhil Mullainathan, Vipin Narang, Neha Narula, David Parkes, John Shim, Scott Stornetta, Alex Tabarrok, Aviv Zohar, and seminar participants at Chicago Booth, the MIT Digital Currency Initiative, NBER Monetary Economics, Harvard, Carnegie Mellon, UPenn, Virtual Market Design, UIC, University of Tokyo, and Northwestern. Ethan Che, Natalia Drozdoff, Matthew O'Keefe, Anand Shah, Peyman Shahidi, and Jia Wan have provided excellent research assistance. Disclosure: the author is a technical advisor to a project pursuing frequent batch auctions for decentralized finance. The author does not have any other financial interests that relate to this research.

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

1 Introduction

From a scientific perspective, what Satoshi Nakamoto invented is a new kind of trust — trust that is completely anonymous and decentralized, without the need for support from traditional sources such as rule of law, reputations, relationships, collateral, or trusted intermediaries. More details will follow below, but at a high level Nakamoto (2008) invented an elaborate scheme, combining ideas from computer science and economics, to incentivize a large, anonymous, freely-entering and -exiting mass of compute power around the world to pay attention to and collectively maintain a common data set, enabling trust in this data set. This invention enabled cryptocurrencies, including Nakamoto's own creation Bitcoin. The specific data structure maintained by the large mass of compute power is called a blockchain.[1]

It is no understatement to say that Nakamoto's invention captured the world's attention. At its recent 2021 peak, the combined market capitalization of Bitcoin and other crypto assets was about $3 trillion, and even this figure seems to understate the amount of cultural, political and commercial attention that has been paid to blockchains and cryptocurrencies. Yet, 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 black-market transactions, as opposed to more mainstream uses by consumers or financial institutions.[2] Ironically, most of the speculative volume is through cryptocurrency exchanges, such as Coinbase, Binance or FTX — which are trusted financial intermediaries! That is, the largest use of cryptocurrencies at present is not even taking advantage of the anonymous, decentralized trust.

[1] 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 usefulness (Haber and Stornetta, 1991; Bayer, Haber and Stornetta, 1993.) This form of blockchain is sometimes called a permissioned or private 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.

[2] 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 credible public observers have also described cryptocurrency activity to date as mostly speculative or black-market. For example, Treasury Secretary Janet Yellen said in Feb 2021 "I don't think that bitcoin ... is widely used as a transaction mechanism ...To the extent it is used I fear it's often for illicit finance. ... It is a highly speculative asset." (Cox, 2021). SEC Chair Gary Gensler said in Aug 2021 "Primarily, crypto assets provide digital, scarce vehicles for speculative investment. ... These assets haven't been used much as a unit of account. We also haven't seen crypto used much as a medium of exchange. To the extent that it is used as such, it's often to skirt our laws with respect to anti-money laundering, sanctions, and tax collection. It also can enable extortion via ransomware ..." (Gensler, 2021). The Economist magazine's Sept 2021 feature on decentralized finance writes "All these services are efficient and creative solutions to financial problems. ... The problem is that, so far, they are all being used to facilitate an incorporeal casino." (The Economist, 2021).

Meanwhile, the compute power devoted to maintaining Bitcoin's data is estimated to account for about 0.3-0.8% of *global* electricity use,[3] and simple math in this and other papers suggests that it accounts for on the order of $15 billion of deadweight loss per year. Ethereum, the other most prominent cryptocurrency, accounts for another roughly $15 billion per year of deadweight loss per year to secure its trust.

This paper argues that Bitcoin and the Nakamoto blockchain — while undeniably ingenious — have serious intrinsic economic limitations. The paper shows that the cost of maintaining Nakamoto's anonymous, decentralized trust both (i) is very large in absolute terms relative to the stakes involved, and (ii) *scales linearly* with the stakes involved. Thus, *if* Bitcoin is to become significantly more economically useful than it is today, *then* the cost of maintaining Bitcoin must grow commensurately as well for it to remain trustworthy. The analysis ultimately suggests skepticism that Bitcoin and its anonymous, decentralized trust will play a major role in the global economy and financial system.

The core of the paper's argument is just three simple equations.

The first equation is a zero-profits condition for what is commonly known as "Bitcoin mining" — i.e., an equilibrium condition on the amount of computing power devoted to maintaining the anonymous, decentralized trust as a function of the compensation paid to this computing power. The computer science details behind the Bitcoin mining process are complicated (see Section 2), but the economics is standard free-entry logic. Indeed, variations on this first equation have appeared in numerous other prior papers. In words, equation (1) says that the dollar amount of compute power devoted to maintaining the trust is equal to the dollar value of compensation to miners. For a sense of magnitudes, at recent levels this compensation is about $250,000 per block of data, or about $40 million per day.

The second equation is an incentive compatibility condition: how much trust does a given level of compute power produce? The Achilles' heel of the form of trust invented by Nakamoto is that it is vulnerable to what is known as a "majority attack." Nakamoto's method for creating an anonymous, decentralized consensus about the state of a dataset relies on a majority of the computing power devoted to maintaining the data to behave honestly. This is not an obscure point; it is in the *abstract* of the famous Nakamoto (2008) paper. Equation (2) captures that it must not be economically profitable for a potential attacker to acquire a 51% majority (or greater) of the compute power. The cost of such an attack must exceed the benefits.

Equation (3) connects equations (1) and (2), i.e., connects the free-entry/zero-profits condition to the incentive compatibility condition. The reason these two equations can be linked is that the

[3] 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. See also Benetton, Compiani and Morse (2021).

amount of honest compute power appears in both. In equation (1), the amount of honest compute power reflects the recurring payments to this compute power. In equation (2), the amount of honest compute power determines the cost of attack. Equation (3) then tells us that the payments to the honest compute power 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 miners are a "flow", whereas the value of attacking the system is more like a "stock." So equation (3) tells us that the flow-like costs of maintaining the trust must exceeed the stock-like value of breaking the trust.

The intuition for why Nakamoto's method of creating trust is so expensive, relative to other methods of creating trust, is that Nakamoto's form of trust is *memoryless*.[4] The Bitcoin system is only as secure at a moment in time as the amount of computing power being devoted to maintaining it at that particular moment in time. Imagine if a country were only as secure as the number of soldiers guarding its border at a given moment, or a brand were only as trustworthy as its flow investment in advertising this hour. In actuality, military defense strategies involve some level of resources devoted 24/7 to maintaining a basic level of security at all times, plus the ability to summon *significantly* more resources in the event of an unexpected attack.[5] Bitcoin's anonymous, decentralized trust model only has the former, not the latter.

There are many alternative methods for creating trust that are more familiar to economists. These include: trust that is backed up by rule-of-law and force (e.g., Schelling, 1960; Becker, 1968); trust that is backed up by a reputation or a brand (e.g., Nelson, 1974; Kreps et al., 1982; Maskin, Fudenberg and Levine, 1994; Tadelis, 1999); trust that is based on a mutually-beneficial relationship and the threat of harming the relationship (e.g., Baker, Gibbons and Murphy, 2002; Levin, 2003); trust that is embodied in large organizations (e.g., Holmstrom and Milgrom, 1994; La Porta et al., 1997*b*), cultural norms (e.g., Kandori, 1992; Guiso, Sapienza and Zingales, 2006), or legal institutions (e.g., La Porta et al., 1997*a*, 1998); trust that is based on a credible threat of retaliation (Axelrod and Hamilton, 1981); trust that is based on collateral (Hart, 1995). The key point is that in each of these cases the trust is more secure than the flow level of investment in maintaining the trust.

There are also many alternative models for creating data security that are familiar to computer scientists and related experts. This includes traditional cryptography. To attack the Bitcoin

[4] Proof-of-stake blockchains are interesting to mention in this context. In its simplest form, proof-of-stake is vulnerable to the same critique in equations (1)-(3) as Nakamoto's proof-of-work; this was conjectured in the conclusion of Budish (2018) and proved formally in Gans and Gandal (2019). Just replace the cost of mining in (1) with the opportunity cost of locking up stake. However, stakes as opposed to computational work open up design possibilities that incorporate more traditional forms of trust, because stakes have *memory* — one can trace a particular staked coin's behavior over time. For more discussion please see Appendix A.6.

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

blockchain requires that the attacker has more compute power than the honest miners; to attack data that is secured by traditional cryptography requires more compute power than a trillion Amazon Web Services, run for more time than the age of the entire universe.[6] In economics terms, the issue with Bitcoin's security model is that it is *linear*. For example, to make the system 1000x more expensive to attack requires 1000x more compute power, which in turn requires a 1000x higher flow payment to Bitcoin miners.

The second part of the paper analyzes the economic implications of two specific attack possibilities: a double-spending attack and a sabotage attack. In a double-spending attack, an attacker engages in transactions in which they send Bitcoins in exchange for other assets (e.g., other financial assets or cryptocurrencies) and then uses their majority of compute power to in essence delete the original transaction from the offical record. This leaves the attacker with both the assets they purchased and the Bitcoins they sent, which they can now spend elsewhere.

This paper makes the ex-post obvious (but, to my knowledge, previously unappreciated) point that the value of engaging in a double-spending attack scales with the economic importance of Bitcoin. In the early days of Bitcoin, common use cases were the purchase of computer equipment or drugs on the dark web. If Bitcoin becomes a large part of the global financial system, such that it becomes commonplace to move millions or even billions of dollars of wealth around the world using Bitcoins, then the returns to engaging in a double-spending attack are commensurately higher.

This observation, in combination with the three equations discussed above, implies that the cost of securing the system against double-spending attacks can be interpreted as an implicit tax on using Nakamoto's anonymous, decentralized trust, with the level of the tax in dollar terms scaling linearly with the level of security. Numerical calculations suggest that this tax could be significant and preclude many kinds of transactions from being economically realistic.

A potential response to this analysis of double-spending attacks is that it fails to account for the cost to the attacker of harming Bitcoin — remember that the attacker must hold Bitcoins in order to engage in the double-spending attack in the first place, and that after the double-spending attack is successful they will have those same Bitcoins back to spend again. Therefore, if the value of Bitcoin falls because of the attack, which seems plausible, then the cost of the attack is commensurately higher. This argument is even more forceful if, in addition, the technology used for mining is specific to Bitcoin, which has indeed been the case for Bitcoin since mid-2013. Now

[6] Bitcoin's current level of compute power is about 2×10^{20} hashes per second. Hence, to attack the Bitcoin network requires an amount of compute power greater than 2×10^{20} hashes per second. As I will discuss below, this level of compute power has a flow cost of about $40 million per day and would require access to about $12 billion of specialized capital. 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 14 billion years, that would get you to about 10^{45} hashes.

the attacker not only harms the value of their Bitcoins, but also the value of their Bitcoin-specific compute power. The Bitcoin Wiki classifies the majority attack into its "Probably Not a Problem" category for this reason (Bitcoin Wiki, 2020*c*), making the following argument:

> "A miner with more than 50% hash power is incentived to reduce their mining power and refrain from attacking in order for their mining equipment and bitcoin income to retain its value." (Bitcoin Wiki, 2022)

This argument is easily seen to be correct in the sense that a decline in the value of Bitcoin or Bitcoin-mining equipment does indeed directly raise the cost of attack; so, holding fixed the benefit of attack, the potential harm to Bitcoin does indeed make the system more secure. However, this line of argument directly concedes the point that a majority attack will meaningfully harm Bitcoin. This in turn implies the possibility of an attack motivated by this harm per se.

Thus, the two attacks considered together suggest a "pick your poison" critique of Bitcoin and Nakamoto's anonymous, decentralized trust: *either there must be a high implicit tax rate on the use of Nakamoto's trust, or the system must be vulnerable to a sabotage attack which causes collapse.* Either scenario suggests that it is unlikely that Bitcoin, at least in its current form, becomes an important part of the mainstream global financial system. However, the analysis is completely consistent with Bitcoin continuing to be used for black-market purposes, or in any other use cases where users are willing to pay the high implicit tax.

The final part of the paper is speculative and asks the following question: if the paper's model is correct and the reason Bitcoin has not yet been majority attacked to date can be understood through the lens of the paper's model, then what changes to the environment could cause incentives to flip and lead to a majority attack? This analysis yields three main attack scenarios. First, changes in the conditions in the market for the specific technology used for Bitcoin mining. In particular, if there is a chip glut, including for previous generation "good enough" chips, that would make attack costs more like a flow than a stock. Second, a large enough fall in the rewards to mining due to a decline in either the value of Bitcoin or the number of Bitcoins awarded to successful miners. This would lead to a large amount of specialized mining equipment being mothballed, which would also make the opportunity cost of attack more like a flow than a stock. Third, a large enough increase in the economic usefulness of Bitcoin (without a commensurate increase in the rewards to miners).

The reader will notice that this paper ultimately makes a pretty simple economics argument, but about a computer science innovation that is complicated and confusing. Economists prior to this work who researched blockchains and cryptocurrencies from the perspective of macroeconomics and monetary policy mostly abstracted from the computer science details (e.g., Schilling and Uhlig, 2019). The economics papers that engaged most directly with the computer science details, such as Huberman, Leshno and Moallemi (2021) and Biais et al. (2019), yield numerous important insights but abstracted away the possibility of majority attack. In particular, both of these papers and several others cited below contain some version of equation (1). The computer science literature on Bitcoin and other consensus mechanisms, including that part of the literature worried about vulnerability to attacks, failed to connect the dots between the size of the majority needed to attack and the cost of incentivizing that level of honest compute power in the first place (e.g., Rosenfeld, 2014; Bonneau, 2016). Put differently, computer science security researchers thought about the ideas in equation (2), albeit with different language, but did not connect (2) with the simple zero-profit economics in (1). Thus, despite the simplicity of the argument, no prior work thought about the issues in equations (1) and (2) in the same framework, and hence (3) had eluded discovery prior to this paper.

The remainder of this paper is organized as follows. Section 2 provides a description of Bitcoin and the Nakamoto (2008) blockchain. The goal is to provide the level of computer science detail necessary to justify the economics that follows in a self-contained manner, with pointers for readers who would like additional detail. Section 3 presents the heart of the economic critique of Nakamoto, equations (1)-(3). Section 4 considers double-spending attacks. Theory, computation, and sensitivity analysis are used to analyze the implicit tax necessary to keep the system secure from such attacks. Section 5 considers the possibility of a sabotage attack, and presents the "pick your poison" idea. Section 6 discusses sabotage in combination with specialized capital. Section 7 considers collapse scenarios implied by the paper's analysis. Section 8 concludes with open questions. Appendix A informally 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.

2 Overview of the Nakamoto Blockchain

This section provides an overview of Bitcoin and the Nakamoto (2008) blockchain. The goal is to provide an overview that is self-contained and at a sufficient level of detail to justify the economics analysis in the rest of the paper. Readers interested in additional engineering detail should consult sources such as the textbook treatement of Narayanan et al. (2016), the website Bitcoin.Org (especially its Bitcoin Developer Guide), the Bitcoin Wiki (2020*a*), Tim Roughgarden's (2021) online course, and the original Nakamoto (2008) paper. There are several overviews with additional detail aimed specifically at economists as well, including Halaburda et al. (forthcoming) and Böhme et al. (2015).

Readers already familiar with Bitcoin and Nakamoto (2008) may skip to Section 3 without loss.

2.1 Transactions

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

Elements of a Bitcoin Transaction. The key elements of a Bitcoin 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.

There are two additional details regarding transactions to note, both of which will make more sense later after additional terms are defined. First, transactions also include a fee amount, payable to the miner who adds the transaction to the blockchain. Second, transactions indicate not just the amount of funds (e.g., 10 Bitcoins) but which specific Bitcoins the sender wishes to send (e.g., these specific 10 Bitcoins). The sender does this by referencing a previous transaction or transactions in which they received those specific Bitcoins, where previous means in an earlier block in the blockchain.

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. 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. A central goal of Nakamoto (2008) is to develop a trusted ledger of transactions that does not require a trusted party.

2.2 What is the Nakamoto Blockchain?

This section will describe the Nakamoto blockchain, 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.

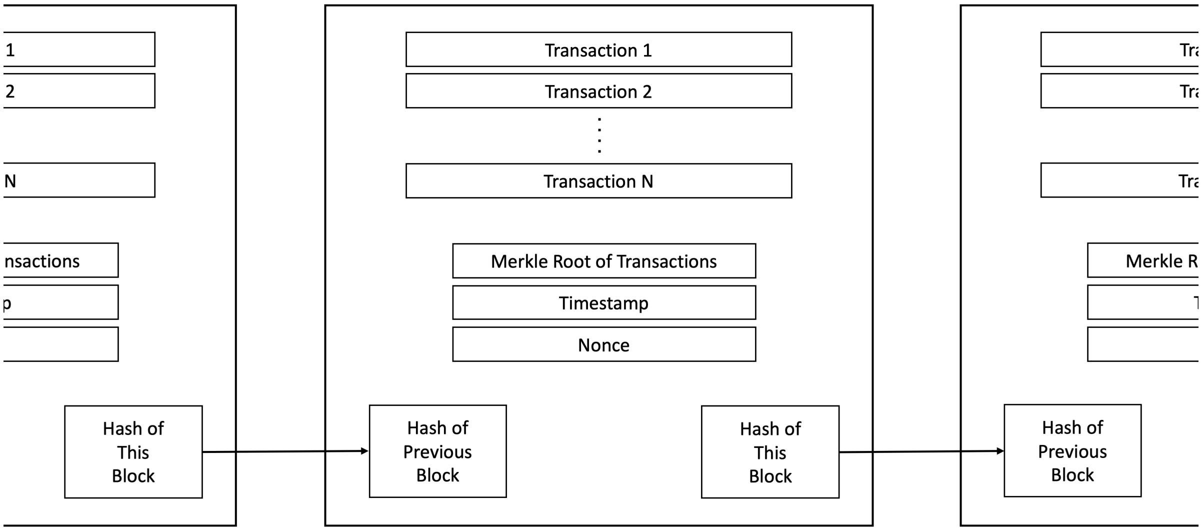
II: Valid Blocks. Bitcoin "miners" compete for the right to add transactions from the mempool to a data structure called the blockchain. The mining computational tournament will be described in the next step. Transactions are added in blocks consisting of about 1000-2000 transactions. Each block of transactions "chains" to the previous block, by including a hash of the data in the previous block. See Figure 1. 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 Bitcoin transactions to be valid, the following 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 Bitcoins 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 Bitcoins to multiple receivers.

III: Bitcoin Mining Computational Tournament. The Bitcoin mining computational tournament boils down to a massive, brute-force search for a lucky random alphanumeric string. More

Figure 1: 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.

precisely, Bitcoin miners — where "miners" is just the terminology for compute power that attempts to add new blocks of transactions to the Bitcoin blockchain — choose a valid 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 data in the new block of transactions they are adding (summarized by its Merkle Root), and the hash of the previous block of transactions that they are chaining to, 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 Economic Limits of Bitcoin and Anonymous, Decentralized Trust on the Blockchain" (not including the quotes) is 4abd55893a851... which has zero leading zeros. A block added to the Bitcoin blockchain in April 2022, block 729,999 has the hash

00000000000000000008b6f6fb83f8d74512ef1e0af29e642dd20daddd7d318f

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 16ˆ19, 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; for instance the hash might have to have 19 leading zeros and a 20th digit weakly less than 9.) As of spring 2022, the amount of computational power devoted to Bitcoin mining is about 200 million trillion hashes per second.

When a miner finds a lucky alphanumeric string, they publicly broadcast their block — consisting of the transactions, the hash of the previous block, their lucky alphanumeric string, and their block's hash — 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. Note, critically, that 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 May 2020 to 6.25 Bitcoins per block. Second, the miner earns any transactions fees associated with the transactions they included in their block. The economics of these transactions fees is 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.[7]

IV: Longest Chain Convention. 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 proposed the longest-chain convention — the convention 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, what this convention does is provides incentive for miners to focus their efforts on mining the current longest chain — that way, if they indeed find a lucky alphanumeric string and mine a block, the block will be part of the new longest chain, and hence new official record. Importantly, the compensation to miners only vests after 100 additional blocks have been mined on that chain, and then the compensation is only valid on that chain. Thus, to get compensated miners have incentive to focus their efforts on mining the current longest chain.

[7] Indeed, using transaction fees to bid for processing priority has led to forms of front-running in decentralized finance applications. See Daian et al. (2019) and Gans and Holden (2022).

The game-theoretic validity of this argument, that the longest-chain convention induces miners to focus their efforts on mining the longest chain, 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 an equilibrium, but there can be other equilibria as well. Carlsten et al. (2016) show that longest-chain mining 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 mining is a Nash equilibrium, but without a formal game-theoretic model. Notably, all of these prior works explicitly assume that all miners are "small" — that is, they assume away the possibility of majority attack, which will be at the heart of this paper's analysis.

2.3 Vulnerability to Majority Attack

Here is the abstract of Nakamoto (2008) in full:

> "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 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 abstract succinctly summarizes the accomplishment and its vulnerability. 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." 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."

Section 11 of Nakamoto (2008) then analyzes the "scenario of an attacker trying to generate an alternate chain faster than the honest chain" under the assumption that the attacker controls *less* than a majority of the computational power. The analysis shows that such an attack is analogous to a Gambler's Ruin problem and that the likelihood of the attacker pulling ahead of the honest participants given a deficit of z blocks, or the likelihood of a successful double-spending attack given an escrow period of z blocks, is exponentially declining in z. The conclusion is that it "quickly becomes computationally impractical for an attacker to change [the public history of transactions] if honest nodes control a majority of CPU power."

What about an attacker with a majority of computational power? It is widely acknowledged, including in Nakamoto (2008), the computer science literature (e.g., Rosenfeld, 2014; Eyal and Sirer, 2014) and on the Bitcoin Wiki (sections: "Attacker Has a Lot of Computing Power", 2020*c* and "Majority Attack", 2022) that such an attack would succeed.[8] From the Wiki's section on "Majority Attack": "Bitcoin's security model relies on no single coalition of miners controlling more than half the mining power." The economic limits that majority attack places on Nakamoto's novel model of trust are at the heart of this paper's analysis.

2.4 An Important Clarification: Decentralized versus Permissioned Blockchains

Before proceeding with the paper's analysis a clarification is necessary. 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 innovation of Nakamoto (2008). This concept is sometimes known as a permissioned or private blockchain, or sometimes as distributed ledger technology (see, e.g., Gans and Gandal, 2019; Bakos and Halaburda, 2021). An IBM marketing campaign calls 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. 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)

[8] Eyal and Sirer (2014) also show that Bitcoin is 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 probabilitistically 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, rather than to manipulate the blockchain to double spend.

As should be clear, this paper's analysis and concern are about blockchain in the sense of Nakamoto (2008). It should be uncontroversial that well-architected databases are economically useful, even if there is a heated debate about what to call them. Indeed, what this paper highlights is that it is exactly the aspect of Bitcoin and Nakamoto (2008) that is innovative relative to traditional distributed databases — the anonymous, decentralized trust that emerges from proof-of-work — that is the source of its economic limits.

3 Nakamoto Blockchain: A Critique in 3 Equations

3.1 Zero-Profit Condition (Rent-Seeking Competition Among Blockchain Miners)

For the purpose of this section, assume that all participants in the Bitcoin system behave honestly. Incentives to attack will be studied next.

Let p_{block} denote the economic reward to the miner who successfully mines a new block of transactions, i.e., wins a computational tournament. For the purpose of this paper, I will consider the compensation to the lucky miner in aggregate, without distinguishing between whether this compensation is in the form of newly issued Bitcoins (which are a form of seignorage tax on holders of the currency) or transaction fees.

Let c denote the cost per unit time of one unit of computational power. This includes variable costs such as electricity and a rental cost-of-capital for capital equipment (interest and depreciation).[9] It is convenient to conceptualize the unit of time as the amount of time it takes on average to mine one block if participants behave honestly, e.g., a unit of time is 10 minutes for Bitcoin.

If there are N units of computational power in the network, then each unit has a $\frac{1}{N}$ chance of winning the prize p_{block}. Under standard free-entry logic, the equilibrium amount of computational power devoted to blockchain mining, N^*, is characterized by:

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

Equation (1) is the standard characterization of a rent-seeking tournament: the prize in the

[9] For analysis of honest mining, this flow compute cost approach is appropriate whether the capital equipment is repurposable or specialized. Either way, there is a variable cost of electricity and a rental cost of capital equipment (though with specialized capital one can think more carefully about capital adjustment costs, as in Prat and Walter (2021) who use a sophisticated Ss model). When we analyze attacks, on the other hand, the flow cost approach is inappropriate if the capital is specialized and the attack also harms the post-attack value of Bitcoin — this makes the attack cost more like a stock than a flow. I will analyze the scenario of attacks that harm Bitcoin in Sections 5-6.

tournament, p_{block}, is dissipated by expenditures aimed at winning the prize, $N^* c$.[10] That Bitcoin mining can be modeled as a rent-seeking contest is widely known; see for instance 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 (forthcoming) Lemma 1; Ma, Gans and Tourky (2018) equation (7); and Halaburda et al. (forthcoming) equation (4). Prat and Walter (2021) provide empirical support that equation (1) describes actual equilibrium behavior in the Bitcoin mining market, with some additional nuances related to capital adjustment costs. There are also numerous websites that compare current block rewards to current mining costs, which also lends empirical support to the free-entry / zero-profits logic. An analogous example of a rent-seeking tournament is the high-frequency trading arms race; equation (7) of Budish, Cramton and Shim (2015) is very similar to equation (1) here.

The Bitcoin Wiki acknowledges the rent-seeking competition among miners and the logic of equation (1) in detail, under the heading "Weaknesses –> Energy Consumption":

> "... the economic equilibrium for the mining rate is reached when global electricity costs for mining approximate the value of mining reward plus transaction fees. So the higher the value of one bitcoin, the higher the value of mining rewards and transaction fees, the higher the energy consumption of the bitcoin network in the long run. More efficient mining gear does not reduce energy use of the bitcoin network. ... cheaper energy linearly increases mining energy use ... the same conclusions apply to all proof-of-work based currencies." (Bitcoin Wiki, 2020*b*)

For a sense of magnitudes, in 2022 year-to-date Bitcoin's block reward has averaged roughly $250,000 per 10 minute block, which corresponds to about $1.5 million per hour, $40 million per day, and about $15 billion per year. Ethereum's block reward has averaged about $6,000 per block, but since Ethereum's block interval is just 13 seconds on average this corresponds to a similar overall magnitude, of about $40 million per day and about $15 billion per year.

3.2 Incentive Compatibilty Condition (with Respect to Majority Attack)

As discussed in Section 2.3, it is widely understood that an agent with a majority of computational power could successfully attack Nakamoto's novel form of trust. Indeed this vulnerability is in the abstract of Nakamoto (2008).

[10] It is straightforward to allow for heterogeneous mining 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.

What is the economic cost of a majority of computational power? If there are N^* units of honest computational power devoted to each mining tournament, an insider gains a majority with as little as $\frac{N^*}{2} + \epsilon$ units of computational power, at cost $(\frac{N^*}{2} + \epsilon)c$ per unit time. An outsider gains a majority with $N^* + \epsilon$ units, at cost $(N^* + \epsilon)c$ per unit time. The analysis will mostly focus on the costs of outside attacks to be conservative, and because that case most cleanly expresses the conceptual critique of Nakamoto's novel form of trust. That said, readers thinking about the logistics of majority attacks in practice may find insider attacks more worrisome; they are also cheaper.

Consider an outside attacker with AN^* units of computational power, at cost AN^*c per unit time, where $A > 1$ and hence the attacker has an $\frac{A}{A+1}$ majority. Suppose it takes an A attacker t time in expectation to conduct an attack, where as a reminder time is counted in units that correspond to how long it takes in expectation for honest miners to mine one block. The time an attack takes will depend on the size of the majority, escrow periods, and potentially additional factors depending on the kind of attack. In Section 4 I provide a closed-form expression for the duration of a double-spending attack as a function of A and the escrow period. Let $At \cdot N^*c$ denote the *gross cost of attack*.

Let V_{attack} denote the value of an attack. For now, let us think about this value of attack in the abstract, and have in mind that the value of an attack will grow as Bitcoin's economic usefulness and importance grow. I will discuss two specific motivations for attack, double-spending and sabotage, in detail in Sections 4-5.

For the Bitcoin blockchain to be incentive compatible against such an attack, on a gross cost basis, requires

$$At \cdot N^*c > V_{attack}.$$

What I will call the attacker's *net* cost of attack can differ from the gross cost of attack $At \cdot N^*c$ for three potential reasons: block rewards, attacker cost frictions, and effects of the attack on the value of Bitcoin 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 earns At block rewards in expectation. These rewards are worth $At \cdot p_{block}$, which under equation (1) from above, is worth $At \cdot N^*c$. Notice that this is exactly equal to the gross cost of attack![11] I will return to this point shortly.

[11] I am indebted to Jacob Leshno for this observation. The analysis in the June 2018 draft artificially constrained the attacker to earn at most t block rewards, yielding net costs of $(At - t)N^*c$. 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

Second, the attacker may face frictions relative to the cost of honest mining. For example, if the attacker's compute power is less energy efficient than the honest miners' compute power, or because there are costs of 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 Bitcoin. This would increase the attacker's cost, both because the attacker may need to hold Bitcoins to conduct the attack (e.g., to double spend), and because the attacker may need to hold Bitcoin-specific capital equipment. However, this also opens up the possibility of an attack motivated by this harm per se. We will discuss this case in detail in Sections 5-6. Until then, assume that the attack does not harm the value of Bitcoin.

The attacker's net cost of attack is therefore

$$(1+\kappa)At \cdot N^{*}c - At \cdot p_{block},$$

which, using equation (1) above to substitute for p_{block}, implies the net incentive compatibility condition:

$$\kappa At \cdot N^{*}c > V_{attack}. \tag{2}$$

In words, the net cost of attacking the blockchain, $\kappa At \cdot N^{*}c$, must be greater than the benefits of doing so, V_{attack}. The equation captures that what enables the anonymous, decentralized trust of the Nakamoto blockchain is the computational power devoted to maintaining it.

Economically, the key thing to note about (2) is that the cost of attack on the left-hand-side is related to the *flow* cost of maintaining the blockchain, i.e., to $N^{*}c$. In contrast, consider, e.g., mutually-beneficial cooperation in a relationship and the associated temptation to cheat, or a trusted brand that is tempted to shirk on quality. In such cases, the cost of cheating to the cheating party is related to the *stock* value of the relationship or brand they are destroying, not the flow cost of its maintenance. 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). 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 economically cheaper to violate.

From a computer security perspective, the key thing to note about (2) is that the security of the blockchain is *linear* in the amount of expenditure on mining power, i.e., linear in $N^{*}c$ in the left-hand-side of (2). In contrast, in many other contexts investments in computer security yield convex returns (e.g., traditional uses of cryptography) — analogously to how a lock on a door

slightly different simulated net costs than here, and that draft did not have Proposition 1 below.

increases the security of a house by more than the cost of the lock. As just one example: Visa's 2021 Annual Report reports that it owns \$4.3 billion of technology capital (see Note 7 on pg. 90 of Visa, 2021), but it is presumably not the case that any entity with temporary access to \$4.3 billion of computational power can successfully attack Visa. The reason is that Visa uses cryptography, not majority rule.

Of the ingredients in the left-hand-side of (2), At can be computed exactly using a closed-form solution for t, and N^*c can be inferred exactly using (1) and the current level of block rewards. Attacker cost frictions κ are intrinsically hard to make an informed guess about. What I will note here is that as long as these attacker cost frictions, relative to honest miners' costs, are less than 100% of gross costs, then the net cost of attack is less than the gross cost of attack. Put differently, the frictions have to be really high for attacker friction to be a source of security. In the stylized case of zero cost frictions, we have the following remarkable conclusion:

Proposition 1. *If the attacker does not face any compute-cost frictions relative to the costs of honest miners ($\kappa = 0$), and the attack does not cause the value of Bitcoin to fall (as assumed above), then the net cost of attack is zero.*

Proof. Follows directly from the attacker's net cost of attack $(1 + \kappa)At \cdot N^*c - At \cdot p_{block}$ using $\kappa = 0$ and substituting $p_{block} = N^*c$ from equation (1). □

Moroz et al. (2020) and Auer (2019) derive similar results to Proposition 1 building off of Budish (2018). To my knowledge, Jacob Leshno (in a verbal communication with the author), Auer (2019) and Moroz et al. (2020) all discovered this consequence of the Budish (2018) model roughly contemporaneously.

To be clear, zero attack frictions seems unrealistic. But, zero friction is often useful as a benchmark case, and the result does highlight that the net cost of attack may be small.

3.3 Economic Limit I: $p_{block} > \frac{V_{attack}}{\kappa At}$

In the hoped-for equilibrium in which participants are honest, the amount of computational power devoted to maintaining the blockchain is characterized by the zero-profit equilibrium in the mining market, (1). The incentive-compatibility conditions then relate this amount of computational power to the level of security generated. In the gross version, for the system to be secure requires the value of attack to be less than At times the per-block cost of compute power N^*c. In the net version, (2), the term is κAt times the per-block cost — proportionally higher than gross costs if frictions are higher than 100%, proportionally lower if frictions are low. Since N^*c appears in both the zero-profit condition (1) and the incentive-compatibility condition (2), we can combine the two equations:

Proposition 2. *The zero-profit condition (1) and the incentive-compatibility condition (2) together imply the equilibrium constraint:*

$$p_{block} > \frac{V_{attack}}{\kappa A t} \tag{3}$$

In words: the equilibrium per-block payment to miners for maintaining the trust on the blockchain must be large relative to the one-off benefits of attacking it.

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

Equation (3) places potentially serious economic constraints on the applicability of the Nakamoto (2008) blockchain. The blockchain can only be used in economic contexts where users are willing to pay a per-block transactions cost, p_{block}, that is large relative to the value of a one-off attack, V_{attack}. By analogy, imagine if users of the Visa network had to pay fees to Visa, every ten minutes, that were large relative to the value of a successful one-off attack on the Visa network.

4 Attack I: Double Spending

4.1 Mechanics of a Double-Spending Attack

What a majority attacker can and cannot do. Before discussing double-spending attacks it is useful to clarify what, technologically, a majority attacker can and cannot do. Because a majority attacker can find lucky hashes faster, in expectation, than the honest minority, the attacker can create an alternative longest chain of transactions, and replace the honest chain with their alternative chain at a strategically opportune moment. This allows the attacker to control what transactions get added to the blockchain, and allows the attacker, within computational limits, to remove recent transactions from the blockchain — by creating an alternative chain starting from the recent past. The attacker even earns the block rewards for each period of their alternative chain after they make it the new longest public chain.[12]

What the attacker *cannot* do is to create new transactions that spend other participants' Bitcoins. Creating new transactions that spend other participants' coins would require not just a

[12] Block rewards do not vest for 100 block intervals (see Bitcoin Protocol Rules, section “"tx" messages”, item 11; and Bitcoin Developer Guide, section “Transaction Data” (Bitcoin Wiki, 2020*a*; Bitcoin.org, 2022)). Thus, as long as the attacker replaces the honest chain within 100 blocks, there is no ambiguity as to who gets the block rewards. If the attacker replaces the honest chain after more than 100 periods have elapsed things get more complicated. In this scenario, the bitcoins that vested to miners on the honest chain would become unusable because they do not exist on the new longest chain. It seems likely that, in practice, such a long-horizon attack (and possibly even shorter-horizon attacks) would generate an attempt to counter-attack or hard-fork to restore the original honest chain (see Moroz et al., 2020), and that in general there would be a period of chaos and uncertainty in this scenario. I will return to this issue in Section 5 under the discussion of Sabotage attacks and in Appendix A.3 under the discussion of counter-attacks.

majority of computational power, but enough computational power to break modern cryptography: creating a transaction that spends another participant's coins requires learning their private key. A majority attacker cannot simply "steal all the Bitcoins."[13]

Description of double spending. Double spending is the canonical form of majority attack, mentioned in the abstract, introduction and conclusion of Nakamoto (2008). In a double-spending attack, an attacker engages in the following actions in sequence:

(i) the attacker spends Bitcoins. That is, the attacker signs one or more transactions in which they send Bitcoins to other parties in exchange for other goods or assets (e.g., traditional financial assets).

(ii) the attacker allows those transactions to be added to the blockchain. That is, the transactions are added to the longest chain as parts of mined blocks in the usual way as described in Section 2.2.

(iii) the attacker works in secret to create an alternative longest chain. In this alternative chain, the Bitcoins that were sent to other parties in (i) are instead sent to other addresses controlled by the attacker.

(iv) the attacker waits for any escrow periods to elapse, so they receive the goods or assets they transacted for in (i). In Bitcoin a common escrow period is 6 blocks, or about 1 hour.

(v) the attacker then releases their alternative longest chain. The attacker now has the goods or assets they received in (iv) but also has the Bitcoins which they have sent to themselves in the chain in (iii).

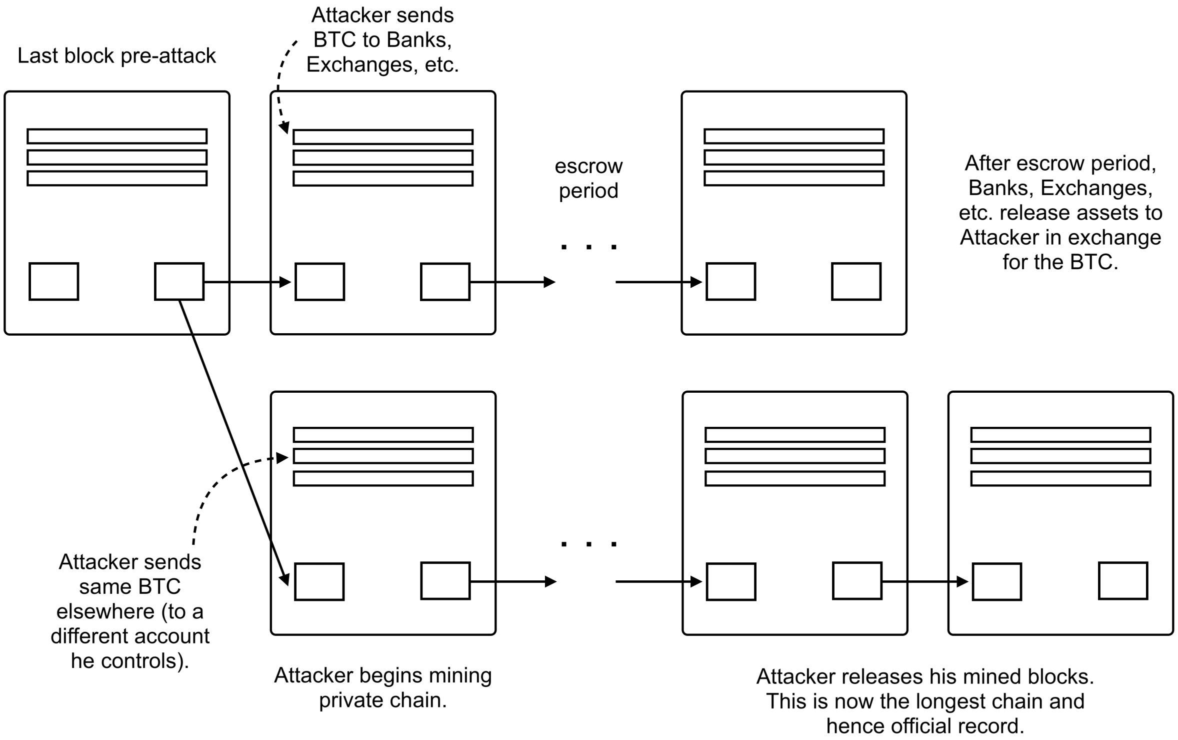
For an illustration, see Figure 2.

Before proceeding with the analysis of the economic implications of double-spending attacks, I would like to reiterate the "if-then" nature of this paper's argument. This paper is trying to

[13] Here is a detailed excerpt on what majority attackers can and cannot do from the Bitcoin Wiki, under "Attacker Has a Lot of Computing Power":

> "An attacker that controls more than 50% of the network's computing power can, for the time that he is in control, exclude and modify the ordering of transactions. This allows him to:
>
> - Reverse transactions that he sends while he's in control. This has the potential to double-spend transactions that previously had already been seen in the block chain.
> - Prevent some or all transactions from gaining any confirmations.
> - Prevent some or all other miners from mining any valid blocks.
>
> The attacker *can't*:
>
> - Reverse other people's transactions without their cooperation.
> - Prevent transactions from being sent at all (they'll show as 0/unconfirmed).
> - Change the number of coins generated per block.
> - Create coins out of thin air.
> - Send coins that never belonged to him." (Bitcoin Wiki, 2020*c*)

Figure 2: Illustration of Double-Spending Attack


Last block pre-attack
Attacker sends BTC to Banks, Exchanges, etc.
escrow period
After escrow period, Banks, Exchanges, etc. release assets to Attacker in exchange for the BTC.
Attacker sends same BTC elsewhere (to a different account he controls).
Attacker begins mining private chain.
Attacker releases his mined blocks. This is now the longest chain and hence official record.

Notes: See the text for description.

take seriously the "if" possibility in which Bitcoin and Nakamoto's anonymous, decentralized trust become a more important and useful part of the global economic and financial system. Some responses to the first draft of this paper's double-spending analysis are about why double-spending attacks would be hard to execute at the scale imagined in this section at *present* — not in the hypothesized future in which Bitcoin and Nakamoto trust are much more integrated with the global economy and financial system.

4.2 Analysis Framework

Equation (3) tells us that the possibility of a double-spending attack places economic constraints on Nakamoto's anonymous, decentralized trust. To understand these constraints we need to analyze the benefits of a double-spending attack (the V_{attack} term) and the expected cost of a double-spending attack in block-compute-cost units (the κAt term).

4.2.1 Benefits of Double Spending: 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 the Nakamoto blockchain. Furthermore, they might engage in many such transactions using multiple addresses.

V_{attack}, therefore, should be understood as a statistic on the amount of transaction volume that a large *honest* user of Bitcoin can conduct in a short period of time. The more value that honest users can transact using Bitcoin, the more value an attacker can double spend.[14]

Therefore, I will consider a wide range of 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 as 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. This figure also represents about 10% of Bitcoin's current market capitalization.

4.2.2 Cost of Double-Spending Attack: κAt

It is possible to obtain a closed-form expression for the expected duration t of a double-spending attack. Let A denote the attacker's majority and e denote the escrow period. For simplicity, assume that if the attacker engages in multiple transactions, they are all added to the honest chain at the same time. Motivated by the description of the mining process above, assume that honest miners mine new blocks as a Poisson process with arrival rate 1 and the attacker mines new blocks as a Poisson process with arrival rate A.

In the Appendix I show the following:

Proposition 3. *The expected duration t of the double-spending attack, as a function of the attacker majority A and escrow period e, is given by:*

$$t(A,e) = (1+e) + \left[\sum_{i=0}^{1+e} \left(\frac{i+1}{A-1} \right) \cdot \frac{(1+2e-i)!}{(1+e-i)!e!} \left(\frac{A}{1+A} \right)^{1+e-i} \left(\frac{1}{1+A} \right)^{1+e} \right]. \tag{4}$$

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

Proof. See Appendix B. □

Expression (4) can be understood as follows. In the attacker's best case, their attack takes $1+e$ time. That is, as soon as their assets are released from escrow, the attacker releases their alternative longest chain. This best case occurs if the attacker mines $1+e+1$ blocks before the

[14] 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 how they would scale with Bitcoin's usefulness. Within economics, a model of Chiu and Koeppl (forthcoming) 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.

honest miners mine $1+e$ blocks. Suppose, on the other hand, that the attacker is behind the honest chain by $i \geq 0$ blocks at the time the honest miners mine their $1+e$ 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, as a function of the escrow period e and attacker majority A.

Table 1 provides example calculations of duration t and the gross block-compute-cost term At for a wide variety of escrow periods and attacker majorities. For example, if the escrow period is $e = 6$, which is a fairly common escrow period for Bitcoin, then attacker majorities ranging from $A = 1.2$ to $A = 1.5$ result in attack durations t ranging from 8.77 to 14.37, and gross block-compute-costs At ranging from 13.15 to 17.24. Notably, smaller majorities lead to significantly longer attack durations and costs. If $A = 1.05$, which corresponds to a 51.2% attacker majority, the duration t is 45.06 blocks and the cost term At is 47.31.

If the escrow period is significantly longer than common practice, say $e = 100$ blocks (roughly 16 hours), then attack durations range from 101.0 to 105.1 for attackers with majorities ranging from $A = 1.2$ to $A = 1.5$, and gross block-compute costs At range from 126.1 to 151.5 for attackers with majorities in this range. Notice that as the escrow period grows longer, the average attack duration t gets proportionally closer to the escrow period. The intuition is simple law-of-large numbers.

Also note that, even at very long escrow periods, the gross-cost-minimizing attacker majority is larger than 51%. For escrow periods ranging between $e = 6$ to $e = 1000$, the cost-minimizing attacker majority ranges from about $A = 1.5$ (60%) to about $A = 1.1$ (52%). It is true that a 51% majority is enough to ensure statistically that the attack will eventually succeed, but a cost-minimizing attacker might choose a somewhat larger majority.[15] Appendix B provides numerical analysis of the cost-minimizing attacker majority as a function of the escrow period.

4.3 Economic Limits

Having discussed V_{attack} and At we are now ready to discuss their implications for the economic limits expressed in equation (3). For this discussion it will be helpful to consider several cases for the κAt term, noting that attacker frictions κ are particularly difficult to have much confidence about. I will consider the following cases:

- A base case in which $\kappa At = 16$. If the escrow period is the standard $e = 6$ and the attacker

[15] This is speculative, but it seems possible that the widespread use of the phrase "51% attack" generated a false sense of security about how long a successful attack would take, and hence how expensive attacks would be on a gross-cost basis. Indeed, as shown in Proposition 3, in the limit as the attacker majority converges to 50% from above, the attack duration t goes to infinity, hence so too do the attacker's gross costs.

Table 1: Expected Duration and Gross Cost of Attack

A. Expected Duration of Attack (t)

	$e = 0$	$e = 1$	$e = 6$	$e = 10$	$e = 100$	$e = 1000$
$A = 1.05$	25.51	29.77	45.06	54.44	181.32	1,067.82
$A = 1.1$	13.02	15.42	24.48	30.35	125.81	1,004.04
$A = 1.2$	6.79	8.28	14.37	18.65	105.13	1,001.0
$A = 1.25$	5.54	6.86	12.41	16.44	102.79	1,001.0
$A = 1.33$	4.34	5.49	10.57	14.40	101.47	1,001.0
$A = 1.5$	3.08	4.07	8.77	12.49	101.03	1,001.0
$A = 2$	1.89	2.78	7.39	11.23	101.0	1,001.0
$A = 5$	1.12	2.06	7.00	11.00	101.0	1,001.0

B. Gross Block-Compute Costs (At)

	$e = 0$	$e = 1$	$e = 6$	$e = 10$	$e = 100$	$e = 1000$
$A = 1.05$	26.78	31.26	47.31	57.17	190.38	1,121.22
$A = 1.1$	14.32	16.96	26.92	33.39	138.39	1,104.45
$A = 1.2$	8.14	9.93	17.24	22.38	126.15	1,201.20
$A = 1.25$	6.93	8.57	15.51	20.55	128.49	1,251.25
$A = 1.33$	5.78	7.31	14.06	19.15	134.96	1,331.33
$A = 1.5$	4.62	6.11	13.15	18.73	151.54	1,501.5
$A = 2$	3.78	5.56	14.78	22.45	202.0	2,002.0
$A = 5$	5.59	10.29	35.01	55.00	505.0	5,005.0

Notes: Expected duration t, as a function of attacker majority A and escrow period e, is computed using formula (4) in the text and double-checked using a computational simulation.

majority is about $A = 1.25$, or 55%, then the gross costs of attack At are about 16. This base case assumes attacker frictions are 100% of honest mining costs and hence cancel the benefits of block rewards, i.e., $\kappa = 1$.

- A low friction case in which $\kappa At = 8$. This corresponds to the base-case scenario of an escrow period of $e = 6$ and attacker majority of 55%, but assumes lower cost frictions of $\kappa = 0.5$.

- An expensive attack case in which $\kappa At = 150$. This case corresponds to escrow periods in the range 100-150 blocks (144 blocks is 24 hours), attacker majorities in the range $A = 1.1 - 1.25$, and attacker frictions of roughly 100% of honest miner costs. Alternatively, it represents the base case scenario of $e = 6$ and $A = 1.25$ but with attacker frictions an order of magnitude higher, i.e., $\kappa = 10$.

- A very expensive attack case in which $\kappa At = 1000$. This corresponds to one full week of block-compute-costs.

Analysis of the Base Case Results for the base case analysis are presented in Table 2. To keep the blockchain secure in the base case requires a per-block cost that is about 6% of the value secured against a double-spending attack. This follows directly from equation (3), rewritten as $\frac{p_{block}}{V_{attack}} \geq \frac{1}{\kappa At}$. The per-day costs must be about 900% of the value secured, and the per-year costs must be over 3000 times the value secured.

These security costs are very high relative to the amount secured. For example, to secure the system against the possibility of a \$1 million double-spending attack requires \$3.3 billion of annual security expense.

These costs look more reasonable when viewed on a per-transaction basis. With 2000 transactions per block, the fee per transaction must be 0.003% of the value secured. This now sounds very small, but keep in mind that this is a fee per transaction that is expressed as a percentage not of the transaction amount, but of the maximum possible double-spending attack. For example, if the value secured is \$1 million, then the per-transaction fee has to be \$31. Huberman, Leshno and Moallemi (2021) and other observers remind us to compare the costs of Bitcoin's trust model against the costs of market power in traditional payment systems. This \$31 per transaction roughly corresponds to the amount a U.S. merchant would pay in credit card fees for a \$1,000 credit card purchase.

The difficulty with Bitcoin's trust model is the way these costs scale. If the system is to be secure against a \$1 billion attack, then the necessary fee on a per-transaction basis grows a thousand-fold to \$31,000. The security cost on a per-year basis is now over \$3 trillion. For the system to be secure against a \$100 billion attack requires a per-year security cost of over \$300 trillion — or roughly 4 times global GDP.

Sensitivity Analysis Table 3 provides the results of the sensitivity analysis. The expensive and very expensive cases improve the picture by one or two orders of magnitude, but the costs are still extremely high. In the expensive case, $\kappa At = 150$, to secure against a \$1 billion attack requires per-transaction costs of \$3,333 and annual security costs of \$350 billion. In the very-expensive case, $\kappa At = 1000$, to secure against a \$1 billion attack requires per-transaction costs of \$500 and annual security costs of \$53 billion. Even in the very-expensive case, to secure the system against a \$100 billion attack requires a per-year security cost of over \$5 trillion, which is more than 5% of global GDP.

Table 2: Cost Per (3) to Secure Against Attack: Base Case Analysis

	Per-Block	Per-Day	Per-Year	Per-Transaction
Security Costs as % of Value Secured	6.25%	900%	328,500%	0.003%
To Secure:				
\$1 thousand	\$62.5 dollars	\$9.0 thousand	\$3.3 million	3.1 cents
\$1 million	\$62.5 thousand	\$9.0 million	\$3.3 billion	\$31.3 dollars
\$1 billion	\$62.5 million	\$9.0 billion	\$3.3 trillion	\$31.3 thousand
\$100 billion	\$6.3 billion	\$900.0 billion	\$328.5 trillion	\$3.1 million

Notes: See equation (3) and the text of Section 4.3 for description. This table is based on a base case with $\kappa At = 16$.

Table 3: Cost Per (3) to Secure Against Attack: Sensitivity Analysis

A. Security Costs as % of Value Secured

Attack Scenarios	Per-Block	Per-Day	Per-Year	Per-Transaction
Base Case	6.25%	900%	328,500%	0.003%
Low-Friction	12.50%	1,800%	657,000%	0.006%
Expensive	0.67%	96%	35,040%	0.0003%
Very Expensive	0.10%	14.4%	5,256%	0.00005%

B. Cost to Secure Against \$1 Billion Attack

Attack Scenarios	Per-Block	Per-Day	Per-Year	Per-Transaction
Base Case	\$63 million	\$9 billion	\$3 trillion	\$31 thousand
Low-Friction	\$125 million	\$18 billion	\$7 trillion	\$63 thousand
Expensive	\$7 million	\$960 million	\$350 billion	\$3 thousand
Very Expensive	\$1 million	\$144 million	\$53 billion	\$500 dollars

Notes: See equation (3) and the text of Section 4.3 for description. The sensitivity analyses are based on (i) a base case, $\kappa At = 16$; (ii) a low-friction case, $\kappa At = 8$; (iii) an expensive case, $\kappa At = 150$; and (iv) a very expensive case, $\kappa At = 1000$.

Discussion of Double-Spending Analysis 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 \$0.03.

The double-spending analysis is also consistent with larger-scale black-market uses of Bitcoin, 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 \$300. This is modest relative to the costs of transporting large amounts of cash (Rogoff, 2017).

Where the analysis suggests greater skepticism is the use of Bitcoin and its novel form of trust as a major component of the mainstream global financial system. If Bitcoin 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 \$1 billion — which is <1% of daily trading volume in the U.S. Treasury market — requires a per-transaction security cost of \$31,000, and an annual security cost of over \$3 trillion.

Another important insight that emerges from this analysis is that for the system to be secure for large transactions requires implicit tax rates that render it unusable for smaller transactions. There are some creative models thinking about how to manage this issue, in which only large transactions are put on the blockchain while smaller transactions are netted off chain (requiring traditional forms of trust). While the analysis here suggests skepticism about the use of Nakamoto blockchain even for large transactions, this does seem a constructive response to the issues raised in this paper, and will be discussed more in Appendix A.4.

5 Attack II: Sabotage

An obvious response to the analysis in the previous section is that a double-spending attack would be "noticed" by Bitcoin observers, perhaps after a period of initial confusion. The attack might therefore cause a decline in the value of Bitcoin, which reduces the value of engaging in the double-spending attack in the first place: while the attacker still has the falsely-obtained goods or assets, the Bitcoins the attacker is left with to double spend are now worth less than before. Moreover, the attack would reduce the value of any capital equipment the attacker holds that is Bitcoin-specific. The Bitcoin Wiki classifies the majority attack into its "Probably Not a Problem" category for this reason, as quoted in the introduction.

In this section I will show that this argument is valid: if the double-spending attack harms the value of Bitcoin sufficiently, then this can actually make Bitcoin secure against double-spending attacks. However, this line of reasoning raises the possibility of an attacker motivated by harming

Bitcoin per se. I call this possibility Sabotage, and derive a "pick your poison" result.

Assume that the double-spending attack analyzed in Section 4 causes a proportional decline in the value of Bitcoin of Δ_{attack}. Assume that the attacker holds the minimum amount of Bitcoin necessary to conduct the attack, namely V_{attack} worth, the amount they will double spend. For this section, maintain the assumption from Section 3.1 that the attacker's cost of mining is c per unit of compute power per unit time. In the next section I will explicitly model the stock cost of capital, C, and assume that the attack harms the value of the attacker's capital stock too.

The Δ_{attack} decline in Bitcoin's value modifies the attacker's incentive compatibility condition, (2), in two ways. First, the attacker double spends for V_{attack} of value (e.g., traditional financial assets from a traditional financial institution), but to realize this benefit has to hold Bitcoins worth this amount, which decline in value $\Delta_{attack}V_{attack}$. Hence the net benefit of the attack is $(1-\Delta_{attack})V_{attack}$. Second, the attacker's net cost of attack has to be adjusted for the decline in the value of the block rewards they earn. An A attacker who attacks for t time still earns At block rewards in expectation, but each block reward declines in value by Δ_{attack}. Equation (2) thus becomes

$$(\kappa+\Delta_{attack})At \cdot N^{*}c > (1-\Delta_{attack})V_{attack}.$$

Substituting in equation (1) yields the following modification of (3):

$$p_{block} > \frac{(1-\Delta_{attack})}{At(\kappa+\Delta_{attack})}V_{attack} \tag{5}$$

The larger is Δ_{attack}, the smaller is the per-block cost necessary to deter the double-spending attack. This is easiest to see by considering the extreme case of $\Delta_{attack}=1$, i.e., if the attack causes a total collapse in the value of Bitcoin. In this case, the attacker loses exactly as much in Bitcoin value as they gain from the double spending, so the numerator on the right-hand-side of (5) is zero — in effect, there is no chance to "double spend" at all. More generally, we have the following simple result:

Proposition 4. *For any potential value of a double-spending attack $V_{attack}>0$, and any level of block reward $p_{block}>0$, the Bitcoin blockchain is secure against the double-spending attack if the post-attack decline in Bitcoin's value, Δ_{attack}, is sufficiently high.*

Proof. Follows directly from (5), noting that the numerator of the right-hand-side goes to zero as Δ_{attack} goes to one. □

Proposition 4 may sound reassuring about Bitcoin's security against double spending, but it raises another more troubling possibility: an attacker motivated by the harm to Bitcoin's value per se. Δ_{attack} is thus a "pick your poison" parameter: If Δ_{attack} is small, then the system is vulnerable to the double-spending attack analyzed in Section 4, and the implicit tax on economic activity using the blockchain has to be high. If Δ_{attack} is high, then the system is indeed secure against double-spending attacks — but this concedes that an attack would significantly harm Bitcoin, which in turn raises the possibility of an attacker movitated by this harm per se.[16]

A well-known early paper on the double-spending problem, Rosenfeld (2014), notes exactly this possibility:

> "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)

Call the value of such a sabotage attack $V_{sabotage}$. How big is $V_{sabotage}$? It is hard to say of course, but easy to imagine that the magnitudes are already large, and would be larger still if Bitcoin becomes more significantly integrated into the global financial system. Open interest in CME Bitcoin futures as of June 2022 is about 11,500 contracts, each tracking 5 Bitcoins, worth about $1.8 billion at current prices. According to data from Coinglass, open interest in Bitcoin futures aggregated across the major crypto exchanges is nearly $15 billion as of June 2022, and open interest in Ethereum futures constitutes another $7 billion.[17] These figures give a sense of magnitudes for what could be made at present from a short-selling attack.

The market capitalization of Bitcoin gives another sense of magnitudes for the amount of economic harm a bad actor could cause by sabotaging the system. Bitcoin's market capitalization

[16] You can't have your cake and eat it too. If your view is that Bitcoin's value would fall in the immediate aftermath of the double-spending attack, but then would recover and this is all predictable, then the attacker can just hold on to their Bitcoins until the value recovers, and their cost of attack becomes what it was as originally analyzed in Section 4. A more elaborate version of this argument involves the Bitcoin community coodinating on a hard fork after the attack, to both nullify the attack and enable a recovery in the value of Bitcoin. This "community response" is a valid argument but is inconsistent with Nakamoto's anonymous, decentralized trust. See Appendix A.1 for further discussion.

[17] CME open interest data is available via its website: https://www.cmegroup.com/trading/equity-index/us-index/bitcoin.html_quotes_volume_voi.html. I found open interest data from crypto exchanges at https://www.theblockcrypto.com/data/crypto-markets/futures/aggregated-open-interest-of-bitcoin-futures-daily, sourced to Coinglass. 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.

has ranged from about \$600 billion to \$1 trillion in 2022. Across all crypto assets tracked by CoinMarketCap, market capitalization peaked at about \$3 trillion in Nov 2021 and has ranged from about \$1.2 trillion to \$2.2 trillion in 2022. Paypal co-founder Peter Thiel (2022) recently predicted that Bitcoin will be worth more than \$100 trillion.

6 Sabotage and Blockchain-Specific Capital

Nakamoto (2008) envisioned that blockchain mining would be performed by ordinary computers: "one-CPU-one-vote." Since 2013, however, Bitcoin mining has been dominated by computational equipment that is extremely specialized to Bitcoin mining. These machines have a large number of specialized chips called ASICs (application specific integrated circuits) which have the SHA-256 hash function programmed directly into their hardware — making them extremely fast at Bitcoin mining, and useless for any task that does not involve computing a large number of SHA-256 hashes. Such chips are reportedly on the order of 10,000 times more efficient at Bitcoin mining than re-purposable alternatives such as GPUs or FPGAs.

As emphasized in the introduction, if the capital used to maintain Nakamoto's anonymous, decentralized trust is non-repurposable, like it is for Bitcoin, and the attack causes a collapse of the trust, then the attacker cost model needs to be rethought. In addition to charging the attacker the flow cost $(1+\kappa)At \cdot c$, the attacker must also be charged for the decline in the value of their specialized capital. This makes the attacker's cost more like a stock than a flow, and thus makes the blockchain more secure — but this security rests on the fragile precipice of specific capital and vulnerability to sabotage.

6.1 Economic Limit II: $N^*C > V_{sabotage}$

Let $c = rC + \eta$ denote the per-unit-time compute cost from above, where C denotes the capital cost of one unit of compute power (e.g., 1 ASIC machine), r denotes the rental cost of capital per unit time, inclusive of risk and depreciation, and η denotes the electricity cost per unit time. The honest-mining equilibrium (1) tells us that

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

An outside attacker would need at least N^*C worth of capital to conduct the attack, while an inside attacker would need at least $\frac{N^*C}{2}$ of capital. We can compute N^*C as a function of payments to miners as follows. Let μ denote the capital share of mining, i.e., $\frac{rC}{rC+\eta}$. Thus

$$N^*C = \frac{\mu p_{block}}{r} \tag{6}$$

If we interpret formula (6) on an annualized basis, we have that the stock value of capital N^*C equals the capital share times the annual payments to miners divided by the annual discount rate. As an example calculation, the annual payments to miners are currently about $15 billion. If the capital share of mining is $\mu = 0.4$ (De Vries, 2018; Digiconomist, 2022) and the discount rate is 50% (ASICs depreciate quickly and mining is risky), then we have N^*C is about $12 billion.[18]

Consider the extreme case in which the attack causes a total collapse of the economic value of the blockchain, including the specialized equipment; this is the case for which the incentive constraint against attack is least constraining. Given how large N^*C is relative to the flow costs of attack we found in Section 4, ignore these latter costs and focus only on the capital costs. This yields incentive compatibility constraints for an outside attacker of

$$N^*C > V_{sabotage} \tag{7}$$

and for an inside attacker of

$$\frac{N^*C}{2} > V_{sabotage}.$$

These conditions are considerably less constraining than we found in Section 4, but they still scale linearly. Substituting in (6) yields

$$\frac{\mu p_{block}}{r} > V_{sabotage} \tag{8}$$

for an outside attacker and

$$\frac{\mu p_{block}}{2r} > V_{sabotage}$$

for an inside attacker. See Table 4 for some example calculations. If the capital share is 0.4 and the discount rate is 50%, then to protect the system against a $1 billion sabotage attack by an outsider requires annual compensation to miners of $1.25 billion and a capital stock of $1 billion. To protect the system against a $10 billion sabotage attack by an outsider requires annual compensation to miners of $12.5 billion and a capital stock of $10 billion — these are about the right numbers for annual compensation and the capital stock for Bitcoin at present. The same calculations suggest Bitcoin is secure at current levels against an insider sabotage attack for about

[18] At recent retail prices for Bitmain Antminer ASIC machines it would take roughly $15 billion to match Bitcoin's current hash rate, so this calculation seems in the right ballpark. I do not have any information on how retail prices relate to the prices paid by large-scale miners.

Table 4: Cost Per (8) to Secure Against Sabotage

A. Cost to Secure Against Outsider Sabotage

Value of Sabotage	Annual Miner Payments	Specialized Capital Stock
$1 billion	$1.25 billion	$1 billion
$10 billion	$12.5 billion	$10 billion
$100 billion	$125 billion	$100 billion
$1 trillion	$1.25 trillion	$1 trillion

B. Cost to Secure Against Insider Sabotage

Value of Sabotage	Annual Miner Payments	Specialized Capital Stock
$1 billion	$2.5 billion	$2 billion
$10 billion	$25 billion	$20 billion
$100 billion	$250 billion	$200 billion
$1 trillion	$2.5 trillion	$2 trillion

Notes: The calculations in columns 2 and 3 are based on equation (8) in the text (both the outside attacker version and the inside attacker version). Calculations assume the capital share is 0.4 and the discount rate is 50%, inclusive of risk and depreciation.

$5-6 billion.

Thus, if one concedes that a majority attack on Bitcoin would effectively be a sabotage that would cause the entire trust model to collapse, then, given the specialized computational equipment currently used for Bitcoin mining, Bitcoin is significantly more secure than is implied by the analysis in Section 4 of double-spending attacks.[19]

However, there are three important concerns. First, the security model still does scale linearly, as before. For Bitcoin to be secure against a $100 billion sabotage attack would require a ten-fold increase of the capital stock and annual miner compensation as compared to current levels. To be secure against a $1 trillion sabotage attack would require a hundred-fold increase. Second, this analysis rests on the nature of the chip market; in particular, if the specialized chips become plentiful, or useful for other purposes, that could change the attack cost from a stock to a flow. Third, this analysis is premised on the assumption that an attack would cause a significant collapse in the value of the blockchain. This itself is a serious vulnerability.

[19] A blog post of Joseph Bonneau (2014) is the earliest written version I am aware of of the argument that ASICs might make Bitcoin more secure. I thank Arvind Naranayan for calling my attention to it after I circulated the earlier version of this paper.

7 Collapse Scenarios

Suppose, for the purpose of a speculative discussion, that the Bitcoin blockchain currently does satisfy the IC constraint (7) that is based on a stock cost of attack,

$$N^*C > V_{attack},$$

but does not satisfy the IC constraint (2) that is based on a flow cost of attack:

$$\kappa A t N^* c < V_{attack}.$$

That is, suppose that the reason Bitcoin has not been attacked to date is the combination of the specialized capital and the presumption that a successful majority attack would cause collapse of the trust. But, at the same time, if the attacker costs were a flow not a stock, Bitcoin would not be secure at present levels of security costs p_{block}. For a sense of magnitudes, at current levels of p_{block}, these assumptions place the value of a successful attack on Bitcoin at less than \$12 billion (a rough estimate for N^*C) but greater than anywhere from about \$5 million (base case) to \$300 million (very expensive case). While subjective, this seems plausible.[20]

This paper's analysis framework then suggests three possible scenarios that could precipitate a successful attack.

7.1 Attack Scenario I: Cheap-enough specialized chips

Let $c = rC + \eta$ denote the cost per unit time of one unit of compute power for the most efficient specialized chips, where rC denotes the capital cost and η the electricity cost, as above in Section 6. For the purpose of this section, it is helpful to have in mind that a unit of compute power is some fixed amount of hashes per second, like 1 terahash/sec.

Suppose there are previous-generation ASIC chips that are not economically efficient for mining, but are powerful enough for the purpose of an attack, and exist in large quantity. Formally, these chips have an electricity cost per unit of compute power, say η', that on its own makes the chips cost-prohibitive for mining even if the capital is free: $\eta' > c$.

[20] To date, the largest 51% attack on a proof-of-work cryptocurrency has been the one on Bitcoin Gold in June 2018, for \$18 million. In April 2022 there was a \$182 million 51% attack on Beanstalk, an algorithmic stablecoin project. This attack was not a double-spending attack, but rather the attacker temporarily obtained >51% of the stake in the project, and then voted to adopt a change to the project's code that siphoned off all of its reserve funds to the attacker. This is not exactly a proof-of-stake consensus protocol, as will be discussed in Appendix A.6, but has some similarities when thinking about attack vulnerability. There have been several other blockchain attacks worth >\$100 million that are based on attacking faulty code, as opposed to a 51% attack. See Vigna, 2022 on the Beanstalk attack; Lovejoy, 2020 for a list of 51% attacks (including the Bitcoin Gold attack); and *rekt.news* for general exposition on a variety of recent attacks.

In honest-mining equilibrium, therefore, these previous-generation chips will have a market value of zero. This implies, however, that these chips could be used by an attacker if they exist in sufficient quantity: if there are at least N^* compute units of previous generation chips with $\eta' > c$ and hence zero capital cost, then an attacker could attack at a flow cost of $N^*\eta'$.

The attacker's gross cost of attack would be $At \cdot N^*\eta'$. Even if η' is 10 times higher than c, this cost of attack would still be very low relative to the capital costs in a sabotage attack N^*C. For example, in the base case scenario where At is 16, the attacker's net costs would be $16 \cdot 10 \cdot p_{block}$ (using $N^*c = p_{block}$ per (1)), or \$40 million at $p_{block} = \$250,000$.

At present, there is no reason to think that there are enough previous-generation chips for the purpose of attack. Both the quantity and efficiency of ASICs in the market have been growing dramatically. On an energy-efficiency basis, Bitmain Antminer ASIC machines have improved by a factor of nearly 100x from 2013 to 2022, and by a factor of 3.5x since 2018, when the most recent generation of ASIC chips came online.[21] On a quantity basis, Bitcoin's hash rate is up by a factor of over 10,000x since end-2013, and up by a factor of 4 since 2018.

However, as the Bitcoin ASIC market matures, it seems plausible that this could change.

Variations on Attack Scenario I: Other changes to the chips market that could precipitate attack There are two other conceptually plausible changes to the chip market that could also precipitate attack. These two scenarios seem subjectively less likely than the cheap-enough specialized chips scenario, but are worth mentioning for completeness.

First, it is conceptually plausible that re-purposable chips close some of the efficiency gap versus non-repurposable ASICs. At present, the gap is on the order of 10,000 times — meaning, even if one controlled all of the (re-purposable) compute power owned by Amazon Web Services, one would still have <1% of Bitcoin's hash rate.[22] If technology for re-purposable chips improves, then the math could flip. As in the math described for Scenario I above, if re-purposable compute power were within a factor of 10 of specialized compute power's economic efficiency, then an attack would cost about \$40 million.

Second, it is conceptually plausible that specialized ASIC chips used for SHA-256 hashing become used in a much wider variety of applications than Bitcoin. At present, large-scale use of hash power is primarily for proof-of-work cryptocurrencies, and most large cryptocurrencies have their own hash function.[23] If the use of Nakamoto blockchain becomes much more widespread

[21] Source: "Table of Antminer Models" in Mineur (2021). Energy efficiency is measured as watts per terahash.

[22] Amazon Web Services has \$65 billion of technology capital on its balance sheet as of the end of fiscal year 2021. If this capital is 1/10000th as efficient at Bitcoin mining as Bitcoin ASICs, this is the equivalent of \$6.5 million worth of Bitcoin ASICs, or about 0.05% of the current total Bitcoin-specific capital stock.

[23] I thank Glenn Ellison for this observation and for connecting it to the theory in this paper. If cryptocurrencies A and B each use the same hash function X, and A has more hash power than B, then hash power from A can be

than it is at present, it is at least plausible that Bitcoin's share of the world's SHA-256 hash power is much smaller than it is today. In this case, even though the chips themselves are specialized to SHA-256, they would usable for other purposes even if Bitcoin itself were to collapse.

7.2 Attack Scenario II: Sufficient Fall in Honest Mining Rewards

Suppose there is a large decline in Bitcoin's price for reasons unrelated to this paper's analysis. Since at present most of the block reward, p_{block}, consists of newly issued Bitcoins, such a fall in Bitcoin's price would directly cause a fall in p_{block}. This, in turn, could lead to a glut of ASICs relative to the amount needed to maintain equilibrium in the honest mining market, (1).

Here is an example calculation. Suppose that we are presently in equilibrium as defined by (1), with $N^*(rC + \eta) = p_{block}$. As above, define the capital share $\mu = \frac{rC}{rC+\eta}$. Suppose the block reward declines from p_{block} to αp_{block}, where $\alpha < (1 - \mu)$. Then $N^*\eta > \alpha p_{block}$, meaning that some capital will be "mothballed", i.e., turned off, because if all N^* units of capital are utilized then mining loses money on the basis of electricity costs alone. If the decline is such that $\alpha < \frac{1-\mu}{2}$, then more than 50% of capital will be mothballed. For example, if the capital share μ is 0.4, then Bitcoin would need to fall by 40% for some capital to be mothballed, and Bitcoin would need to fall by 70% for more than 50% of capital to be mothballed. If more than 50% of capital is mothballed for a sufficiently long period of time, this would seem to raise the vulnerability to attack. Economically, the opportunity cost of using otherwise-mothballed equipment to attack is very low. Logistically, large amounts of mothballed equipment might make an attack easier to execute.

Additionally, the number of Bitcoins issued per block reward halves every four years. The next such halving will occur in around March 2024, to 3.125 Bitcoins. In 2032 the reward will halve to less than 1 Bitcoin, and by 2044 the reward will be less than 0.1 Bitcoin. Unless the dollar value per Bitcoin grows significantly, or transaction fees increase significantly, these halvings will cause significant drops over time in p_{block}, and hence could also cause significant amounts of mining capital to be mothballed.

In effect, a large enough fall in the reward to honest mining, whether due to a decline in the value of Bitcoin or a decline in the number of new Bitcoins issued per block or both, could cause us to have the "cheap-enough specialized chips" envisioned in Attack Scenario I.

temporarily diverted to attack B, making the cost of attacking B a flow not a stock.

7.3 Attack Scenario III: Bitcoin Grows in Economic Importance (Relative to Cost)

The first two scenarios identify conditions under which the cost of attack changes from a stock cost to a flow cost, moving us from the incentive constraint (7) to the incentive constraint (2).

The other possible scenario is that Bitcoin grows in economic importance, relative to its level of compensation to miners and capital stock, to the point where the stock-cost incentive constraint (7) itself no longer holds. That is, $V_{attack} > N^*C$.

Speculatively, this seems most likely to occur if Bitcoin becomes more fully integrated into the global financial system. While $12 billion is certainly a lot of money, it is small in the scheme of global finance.

8 Conclusion

The anonymous, decentralized trust enabled by the Nakamoto (2008) blockchain, while ingenious, is *expensive*. Equation (3) says that for the trust to be meaningful requires that the flow cost of running the blockchain is large relative to the one-shot value of attacking it. In the double-spending attack considered in Section 4, the implication is that the transaction costs of the blockchain must be large in relation to the highest-value economic uses of the blockchain, which can be interpreted as a very large implicit tax — e.g., from $500 to $63,000 per transaction if the system is to be secured against $1 billion attacks, all growing linearly with the value of attack. The argument that an attack is more expensive than this flow cost, considered in Sections 5-6, requires one to concede both (i) that the security of the blockchain relies on its use of scarce, non-repurposable technology (contra to the Nakamoto (2008) vision of "one-CPU-one-vote"), and (ii) that the blockchain is vulnerable to a sabotage attack that causes its novel form of trust to collapse. These concessions, in combination with the analysis framework of this paper, in turn imply specific collapse scenarios: if conditions change in the specialized chip market, if a fall in Bitcoin's value or mining rewards leads to enough capital being mothballed, or if Bitcoin grows economically important enough to tempt a saboteur. Overall, the results place potentially serious economic constraints on the use of the Nakamoto (2008) blockchain.

It bears emphasis that the paper's analysis is consistent with the continued use of Bitcoin and the Nakamoto (2008) blockchain for black-market purposes, and more generally in use cases where users are willing to pay the high implicit costs of anonymous, decentralized trust. Rather, this paper suggests skepticism and caution about large-scale uses of the Nakamoto (2008) blockchain by traditional global businesses or the traditional global financial system. Such entities have access to cheaper forms of trust.

Relatedly to this last point, it also bears emphasis that the analysis is 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 or private blockchain (see fn. 1 and Section 2.4). Indeed, what this paper highlights is that it is exactly the aspect of Bitcoin and Nakamoto (2008) that is so innovative relative to these kinds of distributed databases — the anonymous, decentralized trust that emerges from proof-of-work — that is the source of its economic limits.

A very interesting open question for future research is what design possibilities lie between these two poles: that is, between Nakamoto trust at the one extreme, and traditional trust, grounded in rule-of-law and named entities, with reputations, relationships, collateral, etc., at the other. Are there other ways to generate anonymous, decentralized trust in a public dataset that are less economically constrained by the arguments in this paper? Are there useful relaxations of the anonymity and decentralization ideals in Nakamoto (2008), that make the trust much cheaper? Axiomatic characterization theorems of Leshno and Strack (2020) and Chen, Papadimitriou and Roughgarden (2019) provide some useful definitions and constraints on the problem space. These papers show that axioms that relate to strict interpretations of anonymity and decentralization (invariance to name changes, free-entry, and collusion-proofness) essentially imply Nakamoto's compensation scheme to the maintainers of the trust. Thus the options are either (i) modify Nakamoto (2008) without touching the compensation scheme per se, or (ii) relax anonymity or decentralization. Appendices A.4-A.6 discuss three specific ideas along these lines. It will be very interesting to watch this research develop, and see whether it constitutes a compelling response to the concerns raised in this paper.

Another very interesting topic for future research relates to the tension alluded to in the paper's introduction: clearly, there is a lot of cultural, intellectual and financial excitement about Nakamoto's novel form of trust, and decentralization more broadly. Yet at the same time, the majority of cryptocurrency volume to date appears to involve cryptocurrency exchanges (Makarov and Schoar, 2021) — which are centralized, trusted, financial intermediaries. This paper perhaps helps us understand why the direct use of Nakamoto's anonymous, decentralized trust has been relatively economically limited to date. But, how can we make sense of the high use of centralized exchanges, to trade assets whose original premise was the elimination of traditional financial intermediaries? Perhaps the key distinction is between *users* of Nakamoto's novel form of trust, and *speculators* about the importance of Nakamoto's novel form of trust — the latter of whom are perfectly happy to transact via traditional financial intermediaries. If so, then this paper suggests some cause for concern for such speculators, if their speculation is premised on eventual high economic usefulness — though I caution that it is not possible to draw a direct line from the analysis of this paper to an appropriate valuation for Bitcoin or other cryptocurrencies (see Athey et al. (2016) for an early effort to model cryptocurrency valuation as a function of usefulness).

References

Athey, Susan, Ivo Parashkevov, Vishnu Sarukkai, and Jing Xia. 2016. "Bitcoin Pricing, Adoption, and Usage: Theory and Evidence." SIEPR Working Paper No. 17-033.

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

Axelrod, Robert, and William D. Hamilton. 1981. "The Evolution of Cooperation." *Science*, 211(4489): 1390–1396.

Bailey, Norman T.J. 1957. "Some Further Results in the Non-Equilibrium Theory of a Simple Queue." *Journal of the Royal Statistical Society: (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. 2021. "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.

Benetton, Matteo, Giovanni Compiani, and Adair Morse. 2021. "When Cryptomining Comes to Town: High Electricity-Use Spillovers to the Local Economy." *Working Paper.*

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.org. 2022. "Bitcoin Developer Guide." Retrieved May 5, 2022, from `https://bitcoin.org/en/developer-guide`.

Bitcoin Wiki. 2020*a*. “Bitcoin Protocol Rules.” Last modified June 23, 2020. Retrieved April 22, 2022 from `https://en.bitcoin.it/wiki/Protocol_rules#.22block.22_messages`.

Bitcoin Wiki. 2020*b*. “Weaknesses → Might Be a Problem → Energy Consumption.” Last modified June 27, 2020. Retrieved April 22, 2022, from `https://en.bitcoin.it/wiki/Weaknesses#Energy_Consumption`.

Bitcoin Wiki. 2020*c*. “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`.

Bitcoin Wiki. 2022. “Irreversible Transactions → Attack Vectors → Majority Attack.” Last modified April 8, 2022. Retrieved April 22, 2022, from `https://en.bitcoin.it/wiki/Irreversible_Transactions#Majority_attack`.

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. 2014. “Why ASICs May Be Good for Bitcoin.” Last modified December 12, 2014. Retrieved April 22, 2022 from `https://freedom-to-tinker.com/2014/12/12/why-asics-may-be-good-for-bitcoin/`.

Bonneau, Joseph. 2016. “Why Buy When You Can Rent? Bribery Attacks on Bitcoin Consensus.” In *International Conference on Financial Cryptography and Data Security.* 19–26. Springer.

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

Budish, Eric, Peter Cramton, and John Shim. 2015. “The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response.” *Quarterly Journal of Economics*, 103(4): 1547–1621.

Buterin, Vitalik. 2014. “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. 2020. “Combining GHOST and Casper.” *arXiv preprint arXiv:2003.03052.*

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

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.

Chen, Xi, Christos Papadimitriou, and Tim Roughgarden. 2019. “An Axiomatic Approach to Block Rewards.” In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies.* 124–131.

Chiu, Jonathan, and Thorsten V. Koeppl. forthcoming. “The Economics of Cryptocurrencies - Bitcoin and Beyond.” *Canadian Journal of Economics.*

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

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`.

Daian, Philip, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2019. “Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges.” *arXiv preprint arXiv:1904.05234.*

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`.

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.

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).* 436–454. Springer.

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.

Gans, Joshua S., and Neil Gandal. 2019. “More (or Less) Economic Limits of the Blockchain.” NBER Working Paper No. 26534.

Gans, Joshua S., and Richard T. Holden. 2022. “A Solomonic Solution to Ownership Disputes: An Application to Blockchain Front-Running.” NBER Working Paper No. 29780.

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.

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. forthcoming. “The Microeconomics of Cryptocurrencies.” *Journal of Economic Literature.*

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

Holmstrom, Bengt, and Paul Milgrom. 1994. “The Firm as an Incentive System.” *The American Economic Review*, 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.

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

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`.

La Porta, Rafael, Florencio Lopez-de-Silanes, Andrei Shleifer, and Robert W. Vishny. 1997*a*. "Legal Determinants of External Finance." *The Journal of Finance*, 52(3): 1131–1150.

La Porta, Rafael, Florencio Lopez-de-Silanes, Andrei Shleifer, and Robert W. Vishny. 1997*b*. "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 D., and Philipp Strack. 2020. "Bitcoin: An Axiomatic Approach and an Impossibility Theorem." *American Economic Review: Insights*, 2(3): 269–86.

Levine, Matt. 2017. "Bank Blockchains and an Alibaba Box." *Bloomberg View*, Last modified January 10, 2017. Retrieved from 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.

Lovejoy, James. 2020. "51% Attacks." *MIT Digital Currency Initiative.* Last modified Feb 21, 2020. Retrieved May 1, 2022 from `https://dci.mit.edu/51-attacks`.

Ma, June, Joshua S. Gans, and Rabee Tourky. 2018. "Market Structure in Bitcoin Mining." NBER Working Paper No. 24242.

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

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

Mineur, Ally. 2021. “Does the Antminer S19 XP use 5nm Asics?” Last modified Dec 13, 2021. Retrieved May 1, 2022 from `https://minerdaily.com/2021/does-the-antminer-s19-xp-use-5nm-asics/#Chip_Nm_Vs_TH`.

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).

Narayanan, Arvind, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. 2016. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* Princeton, NJ:Princeton University Press.

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

Philippon, Thomas. 2015. “Has the US Finance Industry Become Less Efficient? On the Theory and Measurement of Financial Intermediation.” *American Economic Review*, 105(4): 1408–38.

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. 2021. “COMS 6998-006: Foundations of Blockchains.” Retrieved May 1, 2022 from `https://timroughgarden.github.io/fob21/`.

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

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

Schilling, Linda, and Harald Uhlig. 2019. “Some Simple Bitcoin Economics.” *Journal of Monetary Economics*, 106: 16–26.

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

Vigna, Paul. 2022. “Crypto Thieves Get Bolder by the Heist, Stealing Record Amounts.” Last modified April 22, 2022. Retrieved May 1, 2022 from `https://www.wsj.com/articles/crypto-thieves-get-bolder-by-the-heist-stealing-record-amounts-11650582598`.

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`.

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 Bitcoin's costs to the costs of market power in traditional finance, which are also high.[24] I hope the present draft of the text makes more clear the conditional nature of the paper's argument: if Bitcoin becomes more economically useful, then it will have to get even 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, make clear that the way Bitcoin's security cost model scales is importantly different from how costs scale for traditional finance protected by rule-of-law.

In 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.

A.1 Community

As noted above in Section 5, 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.[25]

The community response argument seems valid as an argument that attacks might be more expensive or difficult to execute than is modeled here, but it raises two important issues.

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

[25] 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.

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, 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 global finance.

A.2 Rule of Law

A related line of argument 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 example, the financial institutions depicted as the victims of a double-spend in Figure 2, 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. It also seems particularly guilty of wanting to "have your cake and eat it too." 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 2. 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. This is kind of like requiring every major financial institution to have not just security guards, but access to a standing army.

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 \$31, instead of \$31,000. To secure against a \$100 billion attack would require costs per transaction of \$3,100, instead of \$3.1 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 "Level 2." In this paradigm, the Bitcoin blockchain ("Level 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.[26]

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

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

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 convention to try to encode what the community would *want* to do in the event of an attack.

The modification to the longest-chain convention 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 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 convention's determination of the sequence of events.

One pursuit along these lines is Leshno, Pass and Shi (in preparation).

A.6 A Different Consensus Protocol: Proof-of-Stake

Proof-of-stake is widely discussed as an alternative consensus protocol to Nakamoto's (2008) proof-of-work. In this paradigm, rather than earning the probabilistic right to validate blocks from performing computational work, one earns the probabilistic right to validate blocks from locking up stake in the cryptocurrency.

The usual motivation for proof-of-stake relative to proof-of-work — the deadweight loss and environmental harm associated with proof-of-work mining, which as noted currently utilizes about 0.3-0.8% of *global* electricity consumption — is in fact completely orthogonal to the concerns in this paper. In its simplest form, proof-of-stake is vunerable to exactly the same critique (1)-(3) as proof-of-work. Just conceptualize c as the rental cost of stake (i.e., the opportunity cost of locking up one unit of the cryptocurrency), as opposed to the rental cost of capital plus variable electricity cost of running the capital. The amount of stake that will be locked up for validation will depend on the compensation to stakers, as in equation (1). This amount of stake in turn determines the level of security against majority attack, as in equation (2). Thus, equation (3) obtains, with the per-block compensation to stakers needing to be large relative to the value of a majority attack. See Gans and Gandal (2019).

However, while in its simplest form proof-of-stake is vulnerable to the same economic limits as proof-of-work, the use of stakes rather than computational work may open new possibilities for establishing trust and thwarting attacks. The advantage is that stakes, unlike computational work, have *memory*.[27] It is possible, for instance, to grant more trust to stakes that have been locked up for a long period of time, and that have never behaved suspiciously (see Appendix A.5 just above), than to stakes that have only recently been locked up. Stakes can also be algorithmically confiscated by the protocol, whereas ASIC machines exist in the "real world", outside of the grasp of the protocol.

Thus, it seems possible that proof-of-stake could make majority attack significantly more expensive (relative to the level of economic activity) than it is under proof-of-work. That said, proof-of-stake has other potential weaknesses relative to proof-of-work, such as the "nothing-at-stake" and "grinding" problems, and its game-theoretic foundations are less well understood. See Halaburda et al. (forthcoming), Section 3.6 for a detailed discussion, and Saleh (2021) for an early game-theoretic analysis.

Notably, Ethereum, the second-largest cryptocurrency after Bitcoin, has been considering a move to proof-of-stake for some time. See Buterin (2014, 2016, 2020); Buterin and Griffith (2019). Much of the other research on proof-of-stake also seems to be happening outside of the traditional academic process. It will be interesting to see if a proof-of-stake protocol proves to be a convincing

[27] For this reason, proof-of-stake violates the anonymity axioms of Leshno and Strack (2020) and Chen, Papadimitriou and Roughgarden (2019). These papers each show that Bitcoin's compensation scheme for miners is the unique such scheme that is anonymous, that does not incentivize miners to merge ("collusion proof") and that does not incentivize miners to assume multiple fake identities ("sybil proof"). The extent to which proof-of-stake can be considered philosophically "anonymous" or "decentralized" is a question that seems to generate passionate debate.

response to the logic of this paper.

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 $1+e$ blocks, therefore passing the attacker transaction's escrow period, 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 and escrow period e, is given by the stopping time formula:

$$t(A, e) = E[\inf\{s : B_H(s) \geq 1 + e,\, B_A(s) > B_H(s)\}].$$

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^{1+e}:

$$S_H^{1+e} \coloneqq \inf\{s : B_H(s) \geq 1 + e\}.$$

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^{1+e}:

$$\begin{aligned} D^{1+e} &:= B_H(S_H^{1+e}) - B_A(S_H^{1+e}) \\ &= (1+e) - B_A(S_H^{1+e}). \end{aligned}$$

If the realization of $D^{1+e} < 0$, 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 $D^{1+e} \geq 0$, 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 $t(A,e)$ based on the sign of D^{1+e} for a tractable expression for $t(A,e)$:

$$\begin{aligned} t(A,e) = {} & E[S_H^{1+e}|D^{1+e} < 0)] \times P(D^{1+e} < 0) \\ & + \sum_{i=0}^{1+e} \Big(E[S_H^{1+e}|D^{1+e} = i] + E[\text{Time for attacker to overcome deficit} = i] \Big) \times P(D^{1+e} = i) \\ = {} & E[S_H^{1+e}] + \sum_{i=0}^{1+e} E[\text{Time for attacker to overcome deficit} = i] \times P(D^{1+e} = i). \end{aligned}$$

The second equality follows from the law of total probability, $\sum_{k=-\infty}^{1+e} E[S_H^{1+e}|D^{1+e} = k] \times P(D^{1+e} = k) = E[S_H^{1+e}]$. Now, there are three terms left to simplify: $E[S_H^{1+e}]$, $E[\text{Time for attacker to overcome deficit} = i]$, and $P(D^{1+e} = i)$.

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

$$E[S_H^{1+e}] = 1 + e.$$

Now consider the second term, $E[\text{Time for attacker to overcome deficit} = i]$. 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 $P(D^{1+e} = i)$. Recall D^{1+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:

$$\begin{aligned} \{D^{1+e} = i\} &= \left\{B_H(S_H^{1+e}) - B_A(S_H^{1+e}) = i\right\} \\ &= \left\{(1+e) - B_A(S_H^{1+e}) = i\right\} \\ &= \left\{B_A(S_H^{1+e}) = 1+e-i\right\}. \end{aligned}$$

Thus, we want to find $P\left(B_A(S_H^{1+e}) = 1+e-i\right)$. To proceed, we first find the probability $P(B_A(r) = k)$ for any realization r of the random escrow length S_H^{1+e} and any possible value of the attacker chain length k 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^{1+e}. The attacker's chain is $Poisson(A)$ and S_H^{1+e} is distributed $Gamma(1+e,1)$, so that:

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

The second equality exploits the independence of $B_A(s)$ and S_H^{1+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 $\frac{\Gamma(k+e)}{(1+A)^{k+e}}$, so that the integrand is exactly the first moment of $Gamma(k+e, 1+A)$. The expression for the mean is well known: $\frac{k+e}{1+A}$. The fourth equality substitutes this expression and simplifies. Hence, plugging in $k = 1+e-i$, the probability of an attacker deficit i at the time the honest chain completes the escrow period is:

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

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

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

obtaining expression (4) in the text as required.

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

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

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

$$t(A,e) \equiv (1+e) + f(A,e).$$

First, consider the limit $\lim\limits_{A\to\infty} t(A,e)$. Observe that each term in $f(A,e)$ either goes to 0 or is bounded by a constant. The first and fourth terms go to 0 in the limit: $0 \leq \lim\limits_{A\to\infty} \left(\frac{i+1}{A-1}\right) \leq$

$\lim_{A\to\infty}\left(\frac{2+e}{A-1}\right)=0$ and $\lim_{A\to\infty}\left(\frac{1}{1+A}\right)^{1+e}=0$. The second and third terms are bounded by a constant: $\frac{(1+2e-i)!}{(1+e-i)!\,e!}$ is constant in A and $\lim_{A\to\infty}\left(\frac{A}{1+A}\right)^{1+e-i}\leq 1$. Hence, the product of these terms is 0 in the limit, so $\lim_{A\to\infty} t(A,e)=(1+e)+0=1+e$ as desired.

Second, consider the limit $\lim_{A\to_{+}1} t(A,e)$. The first term in $f(A,e)$ 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: $\frac{(1+2e-i)!}{(1+e-i)!\,e!}>0$ is constant in A; $\lim_{A\to_{+}1}\left(\frac{A}{1+A}\right)^{1+e-i}=\left(\frac{1}{2}\right)^{1+e-i}>0$; and $\lim_{A\to_{+}1}\left(\frac{1}{1+A}\right)^{1+e}=\left(\frac{1}{2}\right)^{1+e}>0$. Hence, the product of these terms goes to infinity in the limit, so $\lim_{A\to_{+}1} t(A,e)=\infty$ 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 $t(A,e)$, the expected duration of a double-spending attack as a function of the attacker majority A and the escrow period e.

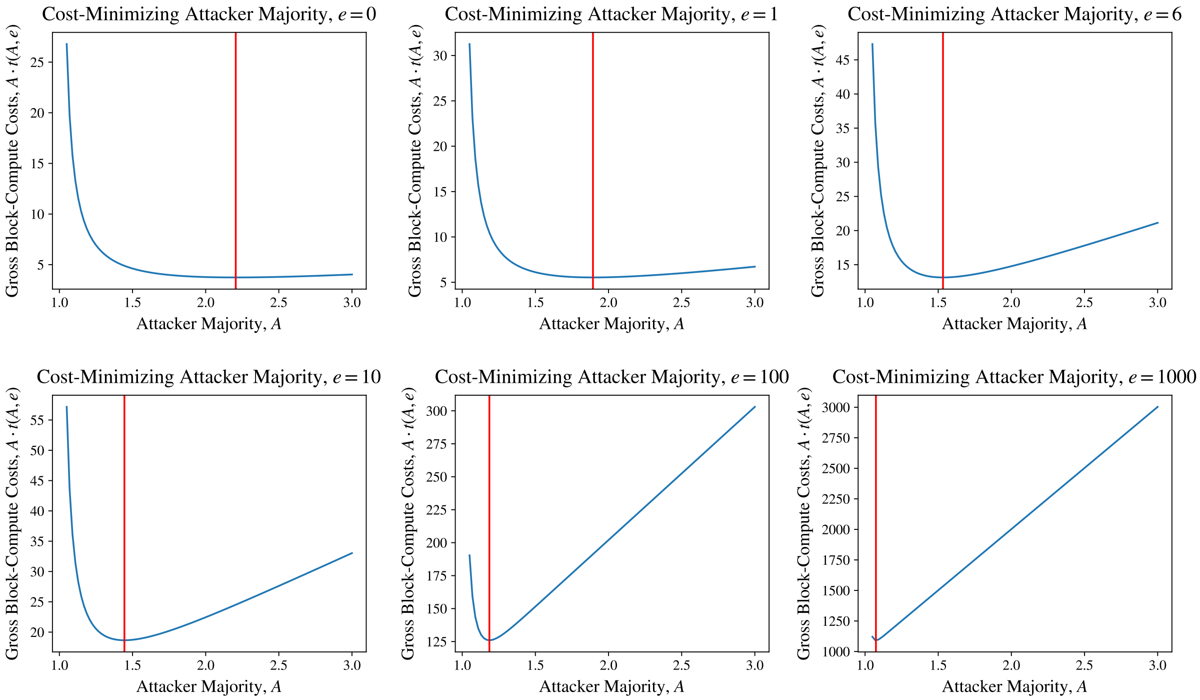
In this section of the Appendix, we use this definition and formula to numerically study the cost-minimizing attacker majority A as a function of the escrow period e.

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

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

While this minimization problem is not analytically tractable, it is straightforward to solve numerically. Figure 3 plots the cost of attack for a variety of escrow periods, as well as the cost-minimizing $A^{*}(e)$.

Figure 3: Attacker Gross Cost Minimization



Notes: The gross cost of attack as a function of majority A is in blue, plotted as $A \cdot t(A, e)$. As discussed in the main text, this quantity needs to be multiplied by equilibrium per-block compute costs N^*c to obtain gross costs in dollars. The gross-cost-minimizing attacker majority $A^*(e)$ 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 escrow length, the cost-minimizing choice of $A^*(e)$ decreases as the escrow length increases. The longer the escrow length is, the more sure a large majority is to mine more blocks than is necessary, by simple law-of-large numbers reasoning.

Table 5 provides the cost-minimizing majority $A^*(e)$, the duration of attack at this attacker majority $t(A^*(e), e)$, and the total gross cost of attack at this attacker majority $A^*(e) \cdot t(A^*(e), e)$ for a variety of escrow periods.

Table 5: Optimal Attacker Majority, Duration and Compute Costs

	$e = 0$	$e = 1$	$e = 6$	$e = 10$	$e = 100$	$e = 1000$
$A^*(e)$	2.21	1.89	1.53	1.44	1.19	1.07
$t(A^*(e), e)$	1.70	2.92	8.57	12.92	106.27	1,016.32
$A^*(e) \cdot t(A^*(e), e)$	3.74	5.53	13.14	18.66	125.99	1,091.66

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

As before, the duration of attack is given in block units of the honest miner, and the total gross cost of attack is given in compute units (N^*c). Note that even very large escrow periods induce a cost-minimizing majority larger than 51% — for example, an escrow period of 1000 blocks induces an optimal attacker majority of $A = 1.07$, or 51.7%.