Breaking BFT: Quantifying the Cost to Attack Bitcoin and Ethereum

Lucas Nuzzi Kyle Waters Matías Andrade

February 2024

Abstract

Much has been hypothesized and feared about 51% attacks on Bitcoin and 34% attacks on Ethereum. However, the costs and benefits associated with perpetrating these attacks remain a mystery. In this paper, we present a novel model to quantify the costs to breach Byzantine fault tolerance thresholds in Bitcoin and Ethereum. We introduce a new metric called Total Cost to Attack (TCA) which encompasses the operational and capital expenditures associated with these attacks. We explore the motivations and expected utility of both profit-driven and ideologically-motivated actors. Our findings suggest that the current state of security in Bitcoin and Ethereum make attacks economically unfeasible and provide empirical evidence of Nash Equilibrium in these networks. This study also challenges the notion that there is a linear relationship between fee revenue and network security, an assumption frequently made when discussing Bitcoin's declining subsidies. Instead, our findings suggest that block producers engage in speculative behavior ahead of fee cycles, which ends up increasing network security even when fees are low and trending downwards. Our analysis contributes to the discourse around the long term viability of deflationary monetary policies used by Bitcoin and Ethereum and their impact on miner incentives and network security.

1 Previous Work

This study builds upon a rich tapestry of research dedicated to understanding consensus attacks from an economical lens. The papers cited here laid the groundwork for pricing security in blockchain systems and constitute significant contributions to the field.

The foundational work of Eric Budish [1] introduced the first model for understanding 51% attacks from an operational cost perspective. This work gave rise to the "Budish cost," which is the notion that block rewards can be used as a proxy for the operational costs related to block production. Budish's hypothesis hinges on the assumption of abundant hashrate available for rent, a premise that has been challenged by the practical limitations of marketplaces like NiceHash.

Expanding on Budish's framework, Hasu et al. [2] incorporated additional dimensions in their analysis, such as Maximum Extractable Value (MEV), and provided the first model to attain the Expected Value of a 51% attack. They argue that in order for Proof-of-Work to successfully deter such attacks, the attacker must have no capital expenditures related to hardware. Further, their analysis deemed necessary for the network's native asset to depreciate post-attack as a required deterrent.

James Lovejoy's [3] contributions substantially improved the understanding of how these attacks have historically materialized in a practical sense. By pioneering the application of an onchain reorg detection tool, Lovejoy produced the first empirical analysis of reorg events across popular networks. By leveraging the "Budish Cost" and "NiceHash Cost" frameworks, Lovejoy was the first to provide a cost to attack that leverages onchain data and price expected returns from double spend attacks.

2 Our Contributions

The model presented in this paper expands upon previous contributions by creating the notion of Total Cost to Attack (TCA), a holistic approach to quantifying the cost to break Byzantine fault tolerance (BFT) in Bitcoin and Ethereum. We take the attacker's perspective and define TCA as the sum of Capital Expenditures (CapEx) plus the Operational Expenditures (OpEx) incurred over time as the attacker attempts to breach the BFT threshold; 50% in Bitcoin and 33% in Ethereum.

The biggest limiting factor of previous analysis was the challenge of measuring the CapEx associated with perpetrating such attacks. This is problematic given that the biggest cost driver of an attack would be the purchase of large quantities of ASICs in Bitcoin and ETH in Ethereum. In the case of Ethereum, CapEx became very easy to measure nominally after its transition to Proof-of-Stake. However, with Bitcoin, methodologies to infer the distribution of ASICs contributing to block production are fairly new. As such, previous studies had to assume the attacker had zero CapEx and instead focus on OpEx, leading to limited reorg costs.

Our previous work on MINE-MATCH in collaboration with Karim Helmy [4] made CapEx assessments substantially easier for Bitcoin. MINE-MATCH is an algorithm designed to identify the specific ASIC model that mined a block thereby enabling us to infer the full distribution of ASIC models working on Bitcoin over time [5]. By sourcing market prices of these machines, we

were able to provide the first CapEx assessment in the context of a 51% attack under different market scenarios. Beyond existing ASICs, we also quantify the scenario where a well-resourced attacker manufactures ASICs with the sole intent of perpetrating an attack.

Put simply, Total Cost to Attack (TCA) is a metric that can be applied to Bitcoin and Ethereum to measure the upfront cost to take over these networks. We deem TCA to be a critical contribution to the field of blockchain security as it enables quantitative reasoning about blockchain network security in a comparable way. We also demonstrate how dissecting a network's security model in the process of calculating TCA contributes to a better understanding and appreciation of specific cost drivers. In turn, we argue this enables for critical security mechanisms, such as Ethereum's churn limit, to be better appreciated as drivers of cost.

Crucially, having quantified TCA for Bitcoin and Ethereum, we provide a more realistic evaluation of the expected value derived from breaches in BFT. Given the substantial and previously not quantified CapEx, we find no viable avenues for an attacker to monetize such attacks in Bitcoin and Ethereum at this time. We also find no expected utility in the scenario where an attacker is ideologically-motivated, given that retaliation techniques make their attacks only temporal. As adversarial actions become unattractive when compared to other strategies, such as honest participation in the network or abstention from attacking, we provide the first empirical evidence of Nash Equilibrium in Bitcoin and Ethereum.

Finally, as an explicit and comparable measure of security, TCA can be used to reason about the effectiveness of crypto monetary policy in achieving the goal of increasing attack costs. Long term security is a salient concern with regards to Bitcoin's monetary policy as it decreases mining subsidies. We provide the first correlation analysis of Bitcoin security, as measured by TCA, and user generated fees. Our results challenge the notion that there is a linear relationship between fees and security. This indicates miners engage in speculatory behavior and have disproportionate returns in high-fee cycles, pushing them to continuously deploy hashrate. Rather than short term fees, security in both Bitcoin and Ethereum appear to be a function of the expectations that block producers have of the value of these network's native tokens.

3 Background

3.1 A Soft Intro to BFT

One of the key properties used to reason about the security of a blockchain system is Byzantine Fault-tolerance. This concept was introduced by Lamport, Shostak, and Pease [6] in their seminal 1982 paper, The Byzantine Generals Problem. In it, they formalized the challenges faced by distributed networks to reach consensus when a subset of participants cannot be trusted. They argue that these systems are only resistant to so many faulty or potentially malicious participants.

The term Byzantine in this context stems from the analogy used in the paper. Picture this: a group of generals, each commanding their own portion of the Byzantine army, needs to come to a unanimous decision about whether to attack or retreat a powerful enemy city they have surrounded. They must come to a unanimous decision because they know a lack of synchronization would grant the city defenders an opportunity to attack and lead to their

defeat.

However, there's a twist. Some of these generals might be traitors, deliberately sowing discord with misleading information. The challenge faced by the Byzantine army is to ensure that loyal generals, despite the presence of these traitors, can arrive at a consensus on whether to attack, or retreat. As "attack" or "retreat" orders are disseminated, it becomes a matter of how many correct orders reach loyal generals relative to incorrect ones.

This problem isn't just a historical or theoretical curiosity; it mirrors the challenges in modern distributed systems on the internet, where malicious actors exist. Enter Byzantine fault tolerance (BFT). In essence, BFT is a property of a system that ensures it can function correctly and reach consensus, even when some participants, or nodes, are acting maliciously or are faulty. It's about building a system so robust that it can withstand a level of "Byzantine failures" and continue to operate seamlessly. This resilience is what essentially makes BFT a cornerstone of blockchain security.

3.2 Practical BFT and Proof-of-Stake

For long after the introduction of the concept of Byzantine fault tolerance, the development of such systems was impractical. Early attempts at developing BFT algorithms failed predominantly due to the requirement of synchronicity. Nodes needed to operate on a strict, often slowed-down schedule. Such a requirement proved unrealistic given how network disruptions and delays can be unpredictable. All of that changed when Miguel Castro and Barbara Liskov published the paper Practical Byzantine Fault Tolerance (pBFT) [7] in 1999, which demonstrated an asynchronous algorithm that achieved BFT.

The genius of pBFT was that it could function effectively without the stringent timing constraints that had hampered previous systems. By removing the dependence on synchronicity, pBFT could handle the erratic nature of real-world networks like the Internet. Asynchronicity allowed for a more robust and flexible approach to fault tolerance, capable of withstanding not only random faults but also coordinated malicious attacks. This was achieved via a 5-step process (request, pre-prepare, prepare, commit, and reply) where nodes were frequently probed and checkpoints created and shared among participants. This combination proved valuable as faulty nodes became more easily identifiable.

A critical aspect of Byzantine fault tolerance is that it is not binary: a system's resistance against faulty nodes is not a 'yes' or 'no'. Instead, it falls on a spectrum. Fault tolerance is always a function of how many faulty nodes are in the network relative to non-faulty nodes. In the case of Practical Byzantine Fault Tolerance (pBFT) algorithms, that tolerance is up to $\frac{1}{3}$ of nodes being faulty. This is often expressed mathematically as $N \geq 3f + 1$ where N is the number of good nodes and f is the number of faulty nodes.

As long as the number of good nodes (N) is greater than three times the number of faulty nodes (f) plus a majority buffer of 1, Byzantine fault tolerance is achieved.

Most Proof-of-Stake (PoS) consensus protocols are governed by the same principles introduced by pBFT. This is why when a single entity in a PoS system reaches the $\frac{1}{3}$ threshold, discussions around the overall security of the system emerge. Critically, PoS systems have introduced additional incentive mechanisms to disincentivize attacks by financially punishing faulty nodes. This is possible in PoS as participants need to lock up a financial bond before being able to

participate in the process. If a participant misbehaves, that bond is destroyed, or slashed. Beyond acting as a disincentive, slashing effectively ejects participants from the consensus cohort, thereby reducing the number of faulty nodes in the system.

3.3 Nakamoto Consensus and PoW

Bitcoin featured the first open-membership consensus protocol to achieve Byzantine fault tolerance. Prior to the advent of Bitcoin's consensus system, often referred to as Nakamoto Consensus, all asynchronous consensus protocols required closed membership. In other words, all members must be authenticated before participating in the consensus process. This property provided the necessary protection against Sybil-attacks, which occur when several fake identities are introduced into the system in order to manipulate it. If a malicious entity targeting a pBFT protocol can cheaply create faulty nodes, it can easily breach the threshold required to break pBFT.

Bitcoin addressed this challenge by making participation in the consensus process costly. This was achieved by requiring nodes to engage in a resource-intensive activity, proverbially called mining, before being able to introduce state changes to the system. By providing a proof that they have engaged in this process, the Proof-of-Work (PoW), any node gains authority to append Bitcoin's blockchain—regardless of whether or not this node was known or trusted by the other nodes in the network [8]. The cost associated with this process prevents Sybil nodes from joining and flooding the system, thereby enabling Bitcoin to be mined by anonymous entities in the open Internet.

Like pBFT, Bitcoin's Nakamoto Consensus is only Byzantine fault tolerant up until a certain point. In order to understand this, we must examine the dynamics of Bitcoin's mining process, which follows a Poisson distribution. As mentioned previously, Bitcoin miners allocate computational power to attain a proof that grants them participation in the consensus process. The difficulty associated with this activity varies periodically so that new blocks can be found in roughly 10-minute intervals. Combined, all computational power implicitly allocated to this activity is called hashrate. Put simply, hashrate refers to the total implicit number of attempts, or hashes, to attain a valid proof at every second, often measured in trillions of attempts per second (TH/s).

In Bitcoin's Nakamoto Consensus, a node's probability of finding a block is a function of its own hashrate relative to the entire network. This was modeled in the Bitcoin white paper [8] using a Poisson distribution formula where $P(X = k)$ is the probability of a node mining a (k) number of blocks over a period of time.

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

In this context, the average rate of success (λ) is defined as the ratio of a node's hashrate relative to the entire network and Euler's number (e) as the base of the natural logarithm.

As such, in order to break Byzantine fault tolerance, a faulty or malicious node must amass over half of the network's hashrate. At this point, its rate of success surpasses any other nodes in the network thereby breaking the system and enabling a host of attacks we will discuss in subsequent sections.

For comparison purposes, we can express this BFT threshold mathematically as $N \geq 2f + 1$

Considering hashrate equally distributed across participants, Byzantine fault tolerance is achieved in these systems if the number of good nodes (N) is greater than two times the number of faulty nodes (f).

4 Evaluating Existing Attack Vectors

If a blockchain consensus protocol breaches the limit of Byzantine fault tolerance, it can no longer be considered secure as it loses both safety and liveness properties. In consensus systems, 'safety' is the guarantee that the state of the ledgers maintained by all non-faulty parties remains consistent over time. This prevents network partitions stemming from disagreements, like which blocks were produced when. Blockchains with stronger safety guarantees have stronger settlement assurances. Safety, in other words, is what grants users a higher degree of confidence that their transactions will not be reversed [9]. Liveness, on the other hand, ensures that even when communication delays are present between participants, transactions will still be added to the ledger in a timely manner. Breaking BFT strips away safety and liveness guarantees of a protocol, leading to state inconsistencies across a large set of participants and no guarantees of transaction inclusion.

Such breaches of Byzantine fault tolerance have been generally referred to as '51% attacks'. Although the threshold used to describe attacks is '51%', any percentage above 50% would technically break BFT in crypto networks that employ Bitcoin-like Nakamoto consensus, as covered in the previous section. Also of note is that, although the term is predominantly used in the context of Bitcoin-like Nakamoto consensus, it has also been applied to Proof-of-Stake (PoS) pBFT networks as well [10]. Similarly, any threshold above 33.33% would break BFT and enable attacks. Perhaps due to the predominance and widespread use of the term "51% attack", the term "34% attack" is not frequently used to describe a break in pBFT protocols.

Another critical factor to understand here is the idea that, in the overwhelming majority of crypto networks, multiple 'versions' of the blockchain can co-exist for a period of time [11]. For example, in a scenario where two miners produce a valid block at the same time when there are no network delays, users would see two conflicting versions, or branches, of the blockchain coexisting. In such cases, participants must reach consensus on a single, often referred to canonical version. This is achieved by applying what is known as a "fork choice rule" which is effectively a part of the consensus algorithm designed to resolve such conflicts and ensure that only a single version of the blockchain persists. Once reconciled to a single version, users that had copies of the conflicting branch have to delete it and download the blocks of the winning branch. This process is called a "blockchain reorganization event", or reorg.

In order to better understand which types of attack could be possible once the threshold is breached, it can be helpful to denote which variables attackers would be able to manipulate. The possibility of more than one branch of the chain coexisting at a point in time makes it so that attackers can broadcast a "competing" version of the blockchain with blocks having completely different compositions than the one previously witnessed. This, as noted earlier, is a break in safety, as there is no longer a single persistent state of the ledger. Liveness is also broken as the attacker can also prevent user transactions from their branch by simply censoring them. In essence, when these properties are broken, attackers have two main "powers" that can be highly destructive in nature: the ability to change block composition, and the ability to

reorg the chain by arbitrarily creating competing branches.

The following section is an attempt at mapping out attack vectors that are still relevant to breaks in BFT. While a multitude of network-level attacks have been proposed over the past decade, many of these vectors have been patched and are no longer relevant. Specifically, intra-protocol "exploits" are outside the attack surface from breaking BFT. Examples of this type of exploit include Bitcoin's 2010 value overflow incident [12] and Ethereum's 2017 EVM out-of-bounds read vulnerability [13]. Nevertheless, vulnerabilities of this nature are far less relevant today given the scrutiny the clients backing these networks have faced as open source software—effectively constituting a years-long bug bounty to exploit any weakness in the protocol. Instead, we will focus on the subset of attacks that remain theoretically possible by an attacker that can continuously manipulate block composition and reorg the chain.

4.1 Block Composition

As safety breaks, an attacker is able to produce alternative versions of the blockchain with different block compositions. One common misconception around such attacks relates to what is possible for the alternative composition. An attacker cannot produce transactions which violate consensus rules, otherwise, the rest of the network will outright reject them. For example, the attacker cannot produce a transaction granting them 99% of the network's native token supply. Only balances that were considered valid and compliant with consensus rules can be included in the attacker's version. While this limits the possibility of inflation exploits, controlling block composition can still be highly disruptive to the network and lead unsuspecting users to financial loss. We discuss the variants of these attacks below.

4.1.1 Network-Wide Censorship Attack

The most obvious type of attack that relates to block composition is a Censorship Attack, where the goal of the attacker is to mimic a Denial-of-Service (DoS) attack. In essence, the attacker uses the power of selecting which transactions will be included in a block to effectively reject all transactions and only produce empty blocks. Since the attacker's version of the chain will be considered valid, it will be as if the blockchain suddenly came to a halt. Like in traditional DoS attacks, the motivation of the attacker could be to degrade trust in the network and push users elsewhere. Given that the attacker would likely incur an opportunity cost of not accepting fee-paying transactions, their motivations might be strictly ideological rather than profit-seeking.

4.1.2 Fee Market Manipulation

Alternatively, an attacker might censor transactions and produce empty blocks with the goal of manipulating fee markets. In most networks, transaction inclusion is determined via a first-price auction where transactions with the highest fees are prioritized. Under normal conditions, this is a function of how many transactions are awaiting block inclusion relative to the overall demand for block space. If only empty blocks were to be mined, urgency around transaction inclusion would push transactors to increase their fees at which point the attacker begins adding high fee-paying transactions. This power would also enable an attacker to implement

any arbitrarily large fee floor as a prerequisite for block inclusion.

It is important to note that newer fee auction mechanisms have, to an extent, mitigated this type of censorship attack. Namely, Ethereum's EIP-1559 introduced the concept of fee destruction whereby the fees that users pay to transact on Ethereum are destroyed, or burned, instead of going to block producers [14]. Under this system, block sizes and fees are dynamic, and adjust in accordance with demand. A fixed fee called the base fee is a prerequisite for block inclusion, but in case of overwhelming demand for block space, users can pay a priority tip to further incentivize block producers to include their transactions. The properties of EIP-1559 as it relates to this attack are discussed in detail in [14] and [15].

4.1.3 Time Freeze Attacks

We have hypothesized a new type of attack solely related to block composition, where no reorgs are necessary. This attack could be highly profitable in blockchains where there is consistent Maximum Extractable Value (MEV), which is broadly defined as additional profit-seeking strategies block producers can engage in given their control over block composition [16]. Ethereum is currently the network where there is the most MEV, and it consequently boasts the most developed infrastructure for this type of activity. Predominantly, MEV in Ethereum takes advantage of arbitrage opportunities that emerge when users leverage Decentralized Exchanges, for example. Depending on the block composition, wide price fluctuations can emerge within a single block, thereby granting opportunities to arbitrageurs.

The goal of the Time Freeze attack is to artificially exacerbate arbitrage opportunities by mining empty blocks. As prices in Decentralized Exchanges (DEXs) diverge from their centralized counterparts, arbitrage opportunities become increasingly profitable. The attacker can then wait n blocks until price action in centralized exchanges maximizes their payout at which point they broadcast a block with their own transactions collecting arbitrage gains. While this attack would be feasible with a breach of the BFT threshold, it is also possible to perpetrate it if a single block producer is guaranteed to mine several blocks in sequence. This can be easily assessed in networks with predetermined schedules for block production, such as Ethereum under RanDAO [17].

4.2 Blockchain Reorganization

As alluded to earlier, reorgs entail reorganizing the blockchain with a different set of blocks. Due to block production following a Poisson process in many blockchains, there is naturally a possibility of two distinct block producers finding a block at the same time. Lovejoy's [3] paper provides a comprehensive analysis of historical reorgs and empirically shows how reorgs tend to be shallow in major networks. In Bitcoin, for example, such naturally-occuring reorgs rarely exceed one block being reorganized off the chain per month. As such, naturally-occuring reorgs tend to be innocuous especially when considering that the composition of the two conflicting blocks will be very similar given that miners tend to select transactions from similar pools.

Nevertheless, reorgs can also be symptomatic of a break in Byzantine Fault Tolerance; a 51% attack'. As mentioned previously, the term has been used both in the context of Proof-of-Work (PoW), Nakamoto Consensus protocols, as well as Proof-of-Stake (PoS), BFT protocols, even though the latter's threshold is in fact 34%. If an attacker is able to breach that threshold,

it becomes possible for them to opportunistically trigger reorgs that are much deeper than naturally-occurring reorgs. As attackers force the removal of previously-produced blocks from the chain in favor of their own, several risks emerge. There are still safeguards in place which would outright prevent attackers from, for example, crediting their accounts with an arbitrarily large number of tokens that previously did not exist. Nevertheless, the ability to reorg the chain can still enable attackers to target users and profit from it. Such attacks are discussed below.

4.2.1 Double Spend Attacks

Double Spend Attacks represent one of the most straightforward ways to monetize the ability to reorg the blockchain once the BFT threshold has been breached [18]. As the name suggests, a Double Spend Attack involves "spending" the very same funds twice in a short period of time. Attackers are able to make an on-chain payment only to then use their power to reorg the chain to cancel it out, almost like a refund where they get to keep the purchased item. Since the attack involves using the network's native token as a payment that is subsequently canceled-out, its victims have historically been centralized exchanges. Nevertheless, it is applicable to any purchase involving on-chain payments.

Consider the following textbook example of a Double Spend Attack involving an exchange. Like many other exchanges, this exchange offers a plethora of crypto assets to its customers. One particular asset, ticker '51ME', appears to be particularly vulnerable to a 51% attack. An attacker takes note and begins accumulating a large quantity of 51ME tokens in what we will refer to as the 'setup phase'. Once the attacker has accumulated enough tokens, say USD 1M worth of 51ME, the attacker proceeds to deposit them onto the exchange. The exchange acknowledges receiving the tokens and, after a period of time, enables the attacker to exchange their 51ME tokens for roughly USD 1M worth of BTC. The attacker then immediately withdraws that BTC from the exchange and into one of their wallets, thereby concluding the setup for the attack.

Upon the withdrawal of the BTC off the exchange, the Double Spend Attack can commence. Recall that reorgs enable attackers to go "back in time" and revert transactions that occurred in the past by reorging those blocks from the chain. This very power enables the attacker to "cancel" the initial deposit of 51ME tokens to the exchange. Through a 51% attack, the block containing that initial transaction can be replaced with a version where that transaction never took place. This will make it so that the balance of 51ME the exchange assumed they had suddenly disappeared. By canceling that transaction, the attacker now has the original USD 1M 51ME tokens and the USD 1M BTC withdrawn from the exchange. The exchange has no recourse as it enabled the attacker to withdraw the BTC in exchange for something it no longer possesses and now has a 1M USD hole in its balance sheet.

4.2.2 Long Range Attacks

Long Range Attacks are an extreme form of 51% attack whereby the attacker is able to reorg a large set of blocks and potentially erase the entire history of the blockchain [19]. This entails secretly producing a large number of blocks and subsequently broadcasting them all at once. The attacker's blocks need to only surpass the current blockchain to be accepted but also erase past history which entails enormous resources in mature networks. Many modern consensus

protocols like Ethereum's LMD-GHOST [20] have eliminated the feasibility of such attacks through a technique generally known as "checkpointing" as we will cover in the next section. While it is still theoretically possible to perform a Long Range Attack on networks that do not implement such techniques, in practice, such an attack was only observed once in the context of major networks.

On Feb. 15, 2021, the Verge (XVG) blockchain went through the largest reorg ever captured in a mainnet, leading to the removal of over 560,000 blocks roughly equivalent to 200 days [21]. By replacing these blocks with the attacker's own set of empty blocks, they were able to capture close to USD 58M worth of XVG in the form of mining rewards emitted by the protocol to incentivize block production. However, although significant in scale, this particular attack does not neatly fit the classical definition of Long Range Attacks as the reorg did not erase the entire blockchain.

4.2.3 Selfish Mining Attacks

Selfish Mining occurs when a block producer opportunistically refuses to share a successfully-produced block with the rest of the network for a period of time. The main goal of this attack is to gain an edge over other block producers, typically miners in networks that employ Proof-of-Work (PoW) [22]. After mining a successful block, the miner will immediately begin working on a new block but will wait to broadcast it. That wait time will increase that miner's chances of mining two successful blocks in a row. If too much time elapses, however, it is probable that competing miners will find a block. If the attacking miner then broadcasts their block, a race condition will ensue. If the attacking miner still succeeds, the network will reorg and converge on that version of the chain. There is some evidence a mild form of the Selfish Mining Attack has become common practice and a rudimentary form of MEV on Bitcoin [23].

Beyond the aforementioned mild form of Selfish Mining, where miners delay block propagation likely by no longer than a minute, such attacks have not proliferated and caused consensus failures as originally predicted. Recent work by Bahrani et al. [24] speculates Selfish Mining has not proliferated as originally predicted due to mining pools not yet reaching the required dominance in the network. The same study demonstrates the possibility of Selfish Mining to be undetectable under a level of mining centralization. Nevertheless, given the scale required to break BFT, it is unlikely that an attacker would choose this attack strategy upon breaching the BFT threshold, given the profitability associated with other attacks.

4.2.4 Time Bandit Attacks

In the seminal paper on MEV, Flashbots 2.0 [25], the authors describe yet another way to monetize a 51% attack. The popularization of Decentralized Finance (DeFi) on Ethereum gave rise to MEV strategies involving operations such as frontrunning. Such strategies, akin to High-Frequency Trading in traditional finance, have netted MEV operators millions of dollars in profit at times within a single block [26]. In a Time Bandit Attack, a block producer reorgs the chain and "replays" transactions that collected MEV in previous blocks. However, instead of the profits going to the original trader behind that MEV trade, the block producer simply replaces the trader's address with their own, "stealing" those profits. Although novel in nature, it is important to note that Time Bandit Attacks have not been observed in the wild and Ethereum's shift to Proof-of-Stake has made such attacks more prohibitive due to the

advent of slashing [27].

5 Attack Mitigation via Confirmation Requirements

The existence of the aforementioned attack vectors has impacted how users and businesses interface with blockchains. Due to the historical short-term nature of these attacks, a simple mitigation strategy involves simply waiting for a number of blocks to elapse before considering a transaction final. This has been colloquially referred to as the "confirmation requirement" and it varies from one network to another. Multiple models have been devised to establish a threshold of blocks at which an attack becomes improbable, with six block confirmations being the most used for Bitcoin as it provides a balance between security and practicality [28].

The property of a transaction being final, or practically irreversible, is called Finality. Finality can be probabilistic (the probability of a transaction being final) or deterministic (the guarantee that a transaction is final). While historically there has been a debate on whether deterministic finality is entirely possible [29], recent work has shown promising results regarding deterministic finality in Proof-of-Stake (PoS) BFT consensus via tools such as accountability and finality 'gadgets' [30]. In Proof-of-Work Nakamoto Consensus, however, finality is indeed probabilistic by its very nature and confirmation requirements are determined by the user's risk profile. Exchanges like Coinbase, for example, require 2 blocks to have elapsed prior to considering a transaction final [31] whereas Kraken requires 3 blocks [32].

6 A General Total Cost-to-Attack (TCA) Model

Having evaluated the significance of breaks in Byzantine Fault Tolerance (BFT) and the role of confirmation requirements as mitigation tools, we will now attempt to price how much an attack would cost. As mentioned previously, our approach iterates upon previous models that have evaluated the cost of attack from an economic lens, chiefly by estimating block production costs. Block production costs are interesting for two reasons. First, they provide an insight into the profitability of block producers and their margins by comparing their revenue relative to their costs. Second, and most importantly in this context, block production costs serve as a proxy cost-to-attack when applied to the threshold at which BFT breaks, e.g. 51% of block production resources for Bitcoin.

Previous attempts at such cost estimates were problematic for major networks as they assumed attackers could easily rent resources at the magnitude required to break BFT. The Crypto51.app website [33], for example, attempts to price how much a 51% attack would cost across many networks based on hashrate rental marketplaces like NiceHash. Such marketplaces offer already-plugged-in machines for rent which can immediately be used for block production. While evaluating attack costs on the basis of rental marketplaces could indeed be valid for low hashrate networks, such a rental market simply does not exist at the magnitude required to attack a network like Bitcoin. In fact, less than 0.01% of Bitcoin's hashrate can be rented via NiceHash [34].

Given that our primary goal with this model is to provide a Cost-to-Attack estimate that is as realistic as possible, our analysis takes a more pragmatic approach that can be generalized to

both Proof-of-Work (PoW), as well as Proof-of-Stake (PoS) assets. Our model proposes that, for any given network, the cost of producing a block can be simplified as the sum of Capital Expenditures (CapEx) and Operational Expenditures (OpEx) incurred over time. CapEx relates to the upfront cost block producers must incur before being able to produce blocks, whereas OpEx relates to the continuous operational costs associated with block production. Like previous approaches, we then extrapolate that cost to that network's BFT threshold.

In systems that use Proof-of-Work (PoW) as the primary Sybil protection mechanism, CapEx represents the cost of acquiring, rather than renting, machines associated with mining, such as Application Specific Integrated Circuits (ASICs) or Graphics Processing Units (GPUs), until the 51% threshold is reached. In systems using Proof-of-Stake, CapEx represents the cost of acquiring a stake in the system by purchasing the network's native asset in the open market until the 34% threshold is reached. These represent the upfront costs that attackers would have to incur in order to effectively take over a network.

Critically, we also incorporate the concept of attack duration to the model via the OpEx. In PoW, this is predominantly the electricity costs associated with running mining machinery for a period of time. In PoS, this is the cost of running node infrastructure leveraging a cloud services provider. Accounting for attack duration is critical because, as covered previously, attackers must be able to break confirmation requirements if they want to monetize their attacks. For example, to beat the 6-block confirmation requirement often imposed on Bitcoin transactions, an attacker would need to produce over 6 blocks, which would likely take over an hour depending on network conditions at the time.

We define the total Cost to Attack (C_{attack}) as the sum of Capital Expenses (C_{CapEx}) and Operational Expenses (C_{OpEx}) over a given period of time. These components can be mathematically represented as follows:

- C_{attack} = Total Cost to Attack
- C_{CapEx} = Capital Expenses at a point in time
- $C_{\text{OpEx}}(t)$ = Operational Expenses as a function of time

The total cost of attack is thus given by the equation:

$$C_{\text{attack}} = C_{\text{CapEx}} + \sum_{t=1}^{T} C_{\text{OpEx}}(t) \tag{1}$$

Where:

- C_{CapEx} includes expenses like ASICs and ETH that would be incurred at a point in time.
- $C_{\text{OpEx}}(t)$ represents the daily operational costs, and these are summed over a period time T to determine the total operational expenses over that period.

6.1 Applying it to Bitcoin

6.1.1 Capex: ASICs

When applying the model to Bitcoin, CapEx represents the cost to acquire Bitcoin ASICs. These are specialized machines designed to perform a single operation: compute the SHA256 hash function. The industrialization of Bitcoin mining led to the development of a variety of different ASIC models, each with their own unique properties. As our study found in 2023, these properties leave unique on-chain footprints that enable us to track their prevalence and usage [4].

Through MINE-MATCH, the algorithm described in that analysis, we were able to estimate the dominance of specific machines over time. This enables for hashrate to be partitioned so that we can have a better understanding of the extent to which a specific ASIC model contributes to Bitcoin's hashrate as a whole.

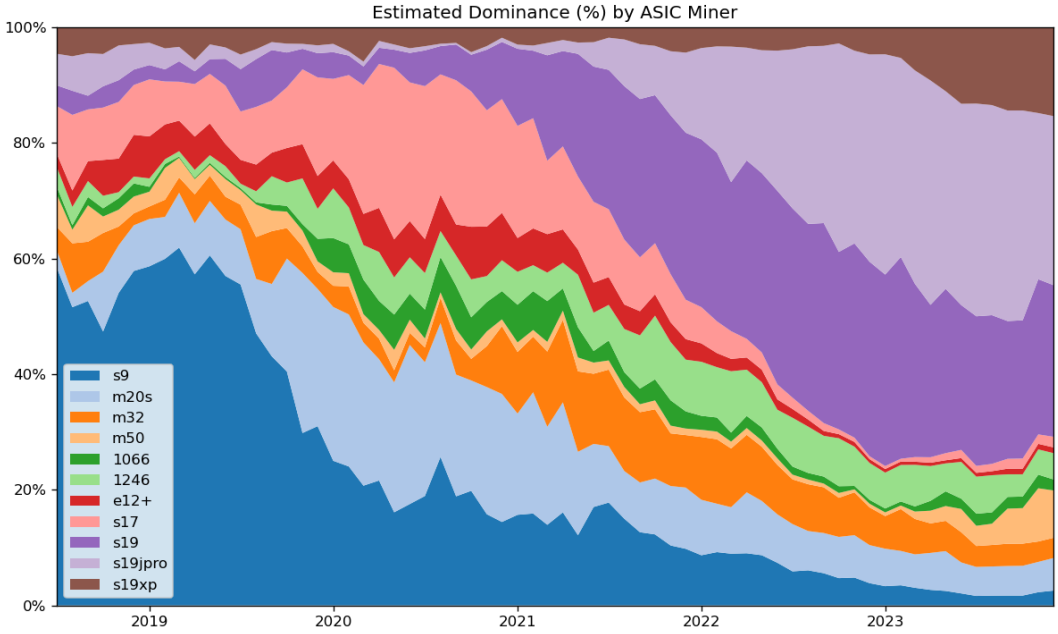

Figure 1: ASIC Model Distribution (%) (Source: Coin Metrics)

While the original study presented ASIC distribution in percentage dominance and hashrate terms, another way of showcasing distribution is in unitary terms. In other words, based on total hashrate partitioned per machine, we can estimate the total count of specific models implicitly plugged in and mining bitcoin over time.

Note in Figure 2 that this notation changes the dominance per unit quite drastically given that newer models are immensely more efficient than previous generations, leading to fewer units producing significantly more hashrate (e.g. S9 vs S19).

This estimate of mining distribution per machine type enables us to simulate how many machines an attacker would have to purchase to buy 51% of the hashrate at a point in time.

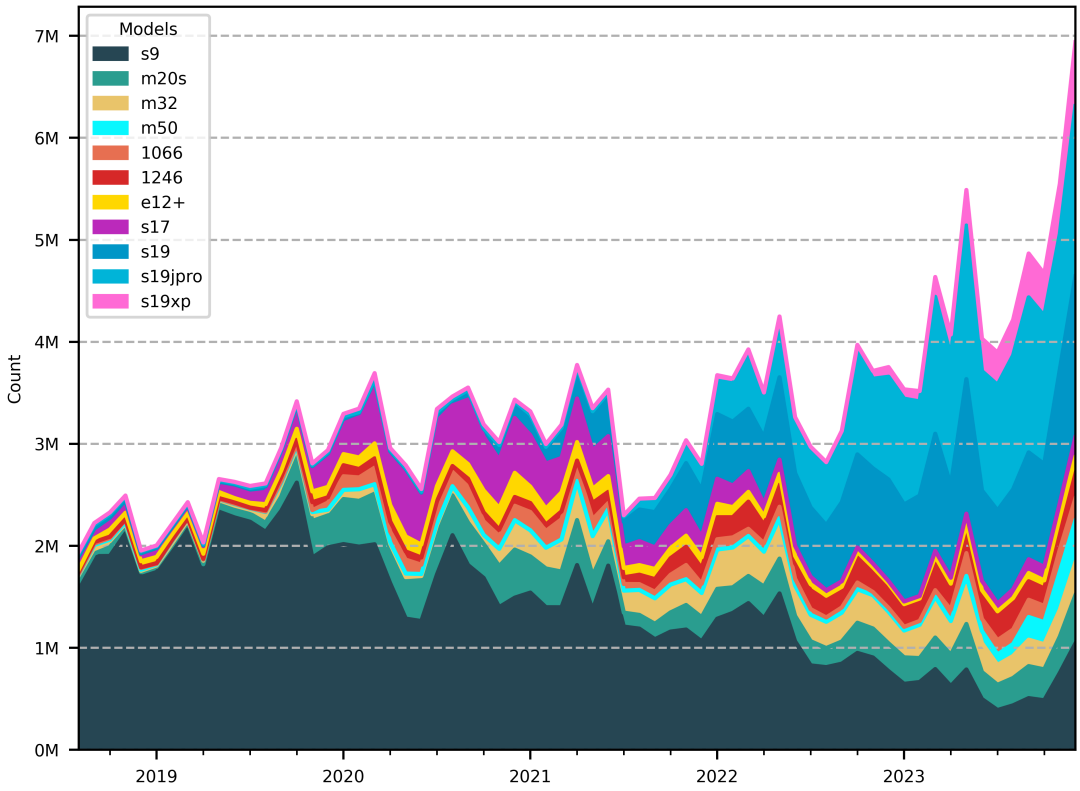

Figure 2: ASIC Model Distribution (Count) (Source: Coin Metrics)

Using a variety of different sources, we made estimates about the unit cost of each of these machines by taking the mean price of various ASIC listings. Table 1 showcases the resulting estimate of how much each of these units costs, as well as additional data points we will subsequently use in the analysis.

Table 1: Mining Hardware Specifications

Model	Price Estimate	Hashrate (TH)	Consumption (kW)
s19xp	$4,000	140	3.01
m50	$2,000	114	3.31
s19jpro	$1,700	100	3.05
s19	$1,200	95	3.25
m32	$1,200	62	3.35
1246	$1,200	90	3.28
e12+	$1,000	50	2.5
s17	$700	53	2.39
1066	$700	50	3.25
m20s	$500	68	3.36
s9	$80	14	1.3

6.1.2 The "Naive Approach"

In order to get the attacker's CapEx, will begin by estimating the cost of buying a little over half of these machines under different scenarios. The goal is to acquire enough machines to surpass the 50% threshold required to break BFT in Bitcoin. Given the lack of historical price indexes for specific ASIC models over time, we will use the prices in the table above for the historical analysis.

The caveat is that, in reality, these prices are relatively volatile and as cyclical as the price of BTC itself as we discuss in subsequent sections. The use of more recent market prices for ASICs also entails a present bias, as it will be reflective of recent market prices.

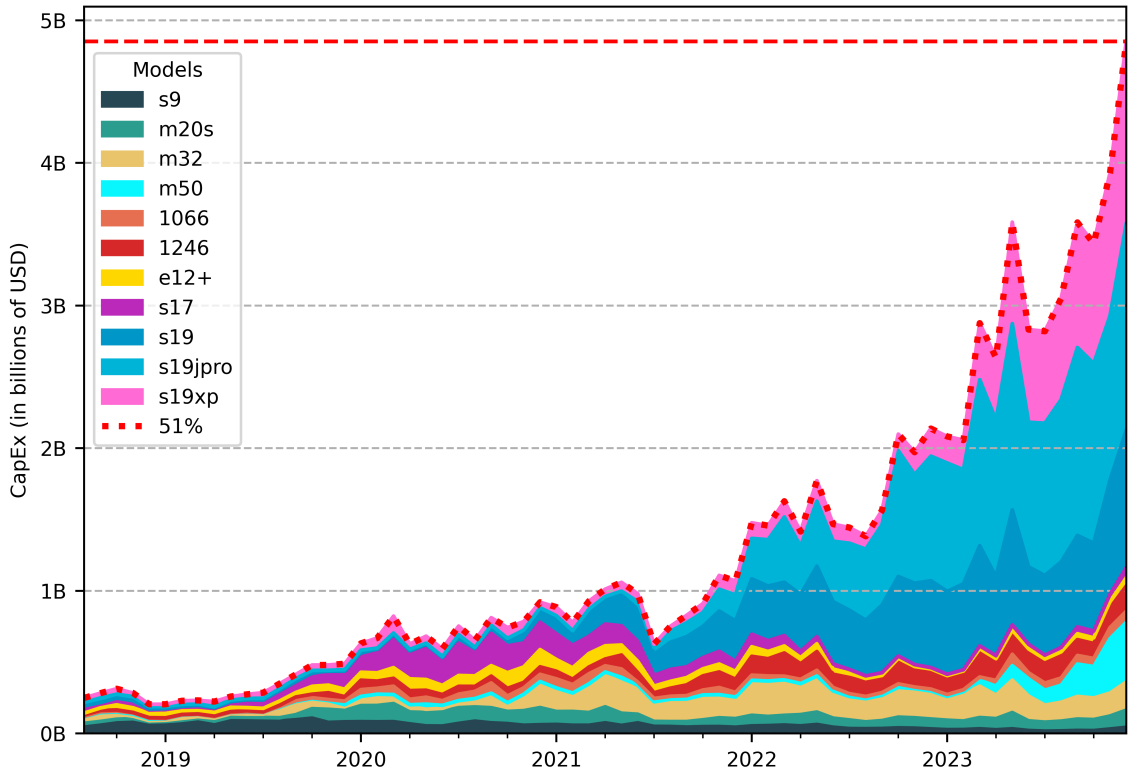

Figure 3: ASIC Model Market Cap (Count) (Source: Coin Metrics)

In this "naive" approach, we simply multiply these fixed prices with the total count of machines implicitly plugged in and take 51% of that figure. Just like aforementioned issues with previous attempts at pricing an attack, this also makes the assumption that these machines are available to be bought, which is not the case. Hence why we will call the approach naive—its goal is to simply extrapolate prices from a small market to the entire network.

6.1.3 Accounting Illiquid ASIC Markets: The Price Model Approach

Following the "naive" approach, we were able to get the equivalent of the "market cap" of each model contributing to Bitcoin. Now, we will attempt to model price increases that would naturally occur if an attacker was motivated to purchase 51% of the ASICs active in Bitcoin.

We will use a relatively simple supply and demand model to account for price increases. As an attacker begins buying up machines, we assume prices increase linearly and that supply is fixed per machine model. This can be described mathematically below:

In order to account for price increases, let's denote:

- P_0 as the flat price estimate of a machine model.
- Q as the total number of machines available for a model.
- Q_a as the number of machines the attacker wants to buy (51% of Q).
- P as the new price of the machine model after the attacker tries to buy 51% of them.
- k is a constant that determines the slope of the price increase. A larger k value means a steeper price increase.

The model for price increase can be defined as:

$$P = P_0 \times \left(1 + k \times \frac{Q_a}{Q}\right) \tag{2}$$

One key consideration here is the elasticity of supply and demand for ASICs as that will inform our k values. Intuitively, both appear to be relatively inelastic. On the supply side, there are several manufacturing constraints that lead to inelasticity, such as availability of semiconductor chips, setting up new mining facilities, and long R&D cycles. On the demand side, the nature of mining itself contributes to inelasticity given the lack of alternative use cases for ASICs, as well as the constant need to remain competitive by upgrading hardware as mining difficulty increases.

To account for these factors in the estimation of K, we needed hard data on how changes in demand affect ASIC prices. To do that, we decided to utilize the price of Bitcoin as a proxy for ASIC demand, under the rationale that rising Bitcoin prices spur increased demand for mining equipment. To evaluate how ASIC prices respond to demand, we sourced aggregated ASIC pricing indexes from HashrateIndex [35], an aggregator of primary and secondary ASIC sales.

We calculated quarter-to-quarter percentage changes in Bitcoin price to represent fluctuations in demand, then correlated these with the corresponding percentage changes in ASIC prices. A simple linear regression analysis was employed to ascertain the historical sensitivity of ASIC prices to changes in demand, with the regression coefficient providing us with a k value. This coefficient, approximately 0.71, quantifies the average propensity of ASIC prices to respond to the inferred demand changes driven by Bitcoin price movements.

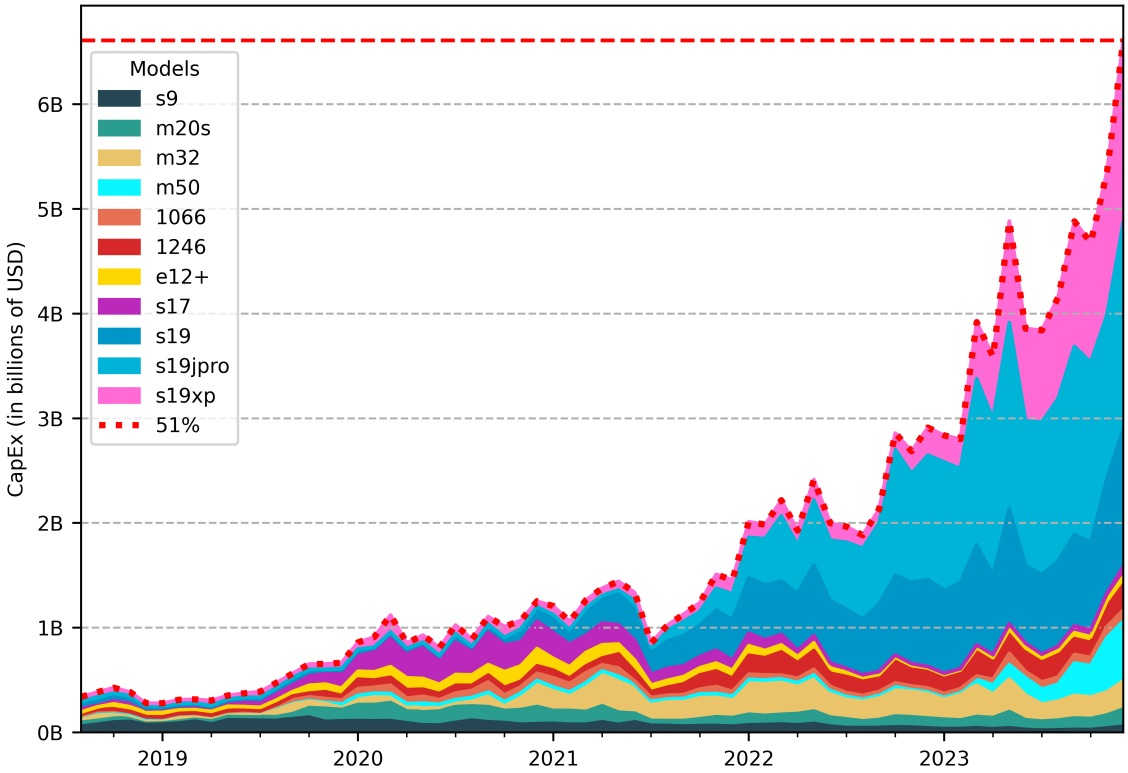

Figure 4: CapEx w/ Resulting Price Increases, USD ($k = 0.71$)

While this methodology provides a quantitative basis for our model, it carries some inherent limitations. Chiefly, the approach presupposes that historical price change dynamics in ASIC markets can be externalized to 51% of machines that are plugged-in. In reality, it is probable that a motivated buyer exhausting the ASIC markets would drive prices substantially higher, leading to a much higher k value reflective of unprecedented market activity.

Despite these constraints, estimating k on the basis of the historical ASIC market remains the most viable method at our disposal. For the sake of speculating the impact of potentially higher k values resulting from the attacker's relentless buying, we also assessed CapEx with $k \times 2$ (where $k = 1.42$), $k \times 2^2$ (where $k = 2.84$), and $k \times 2^3$ (where $k = 5.68$).

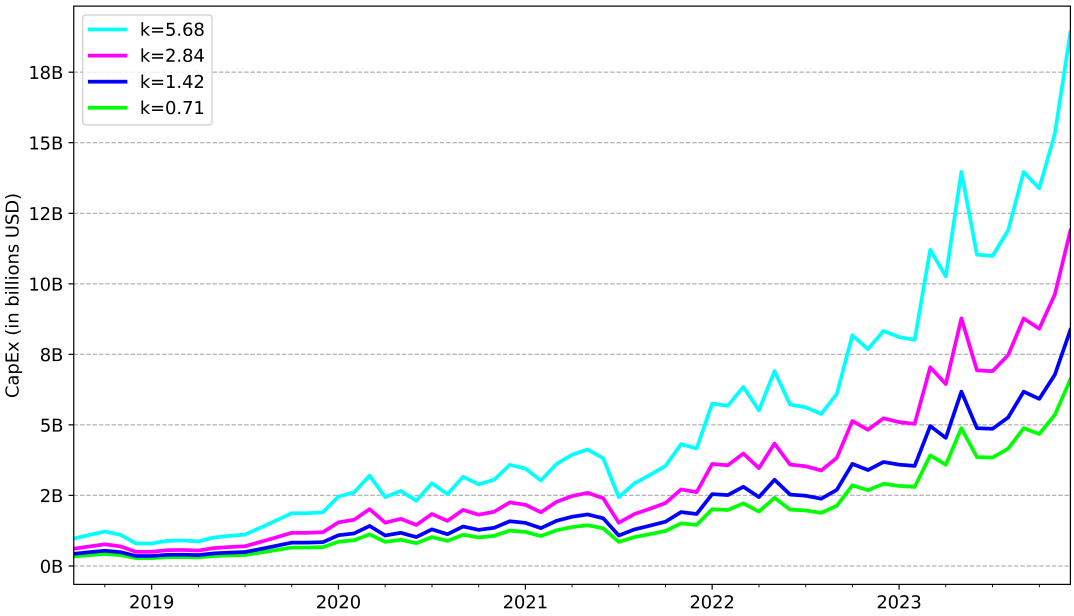

Figure 5: CapEx w/ Resulting Price For Different k values

6.1.4 Scenario: Manufacturing ASICs

Let us now engage in yet another interesting thought experiment that entails a completely different approach to estimating an attacker's CapEx. What if it turns out to be impossible from a logistical or market availability standpoint for the attacker to purchase 51% of the machines engaged in mining Bitcoin? What if the attacker was a nation-state resourceful enough to be able to manufacture its own ASICs at scale with the goal of attacking the network? In this section, we will attempt to model this CapEx scenario as well.

Given that ASIC designs are closely guarded intellectual property, we must make some assumptions regarding which model the attacker would most likely manufacture to estimate production cost. Out of all models available, we deem the Bitmain S9 is the most likely candidate in this scenario given that it is one of the oldest ASIC designs still seeing a level of usage. The S9's manufacturer, Bitmain, has also claimed in a lawsuit that its S9 IP was being used by a competitor, MicroBT, in a product based that allegedly copied the S9 [36]. Additionally, the S9 is arguably the most studied ASIC model to date and the properties of its BM1387 chip are extensively documented [37].

A 2017 analysis of the profitability of Bitmain estimated the production cost of the S9 at roughly $500 per unit [38]. We will assume this as the unitary cost a nation-state would incur if it were able to reverse engineer the S9. Producing the number of units that would reach Bitcoin's current 51% threshold is not enough, though. As the attacker introduces hashrate into the network, the threshold is pushed higher. To accurately model this dynamic, we calculate the total network hashrate post-attack as the sum of the original hashrate and the attacker's additional hashrate. This new total is then used to determine the revised 51% threshold. Mathematically, it is expressed as 0.51 times the sum of the original hashrate and the attacker's end hashrate.

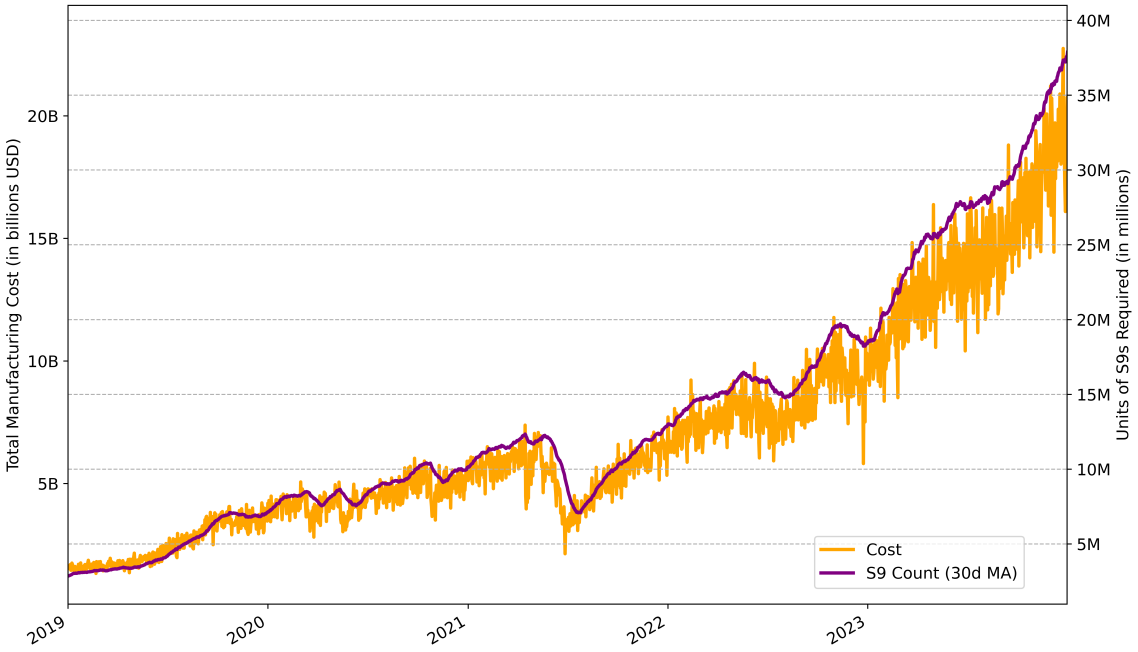

Figure 6: CapEx: Manufacturing S9s

In this far-fetched scenario, the attacker's production costs would surpass 20B USD as it would need to produce close to 40M units of the S9. It would be improbable for the attacker in this scenario not to run into constraints related to the availability of microprocessors. It is important to note that, in the manufacturing scenario, the attacker would be able to drastically increase the rate of block production given the 2-week interval where Bitcoin adjusts for mining difficulty. If released all at once, the added hashrate would shorten inter-block time quite considerably. In turn, this would increase the opportunity of double spends within the same hour.

While the S9 is the most plausible machine in a reverse-engineering scenario, it is also one of the least efficient. If such a motivated attacker were to use the most powerful ASIC available, the upcoming Bitmain S21 [39], the resulting CapEx reflecting December of 2023 would be $5.6B; roughly 25% that of the S9. This assumes a unitary cost of $2,240 per unit and a total of 2.5M machines produced. While still higher than the "naive" scenario, manufacturing at this efficiency and scale would require the repatriation of the manufacturer itself. Even then, the attacker would likely encounter supply chain issues and risk retaliation, as we will discuss in subsequent sections.

6.1.5 Opex: Electricity for a 6-block reorg

Having covered CapEx under different scenarios, let us now discuss the OpEx side of the model. As mentioned previously, OpEx is intended to capture the operational costs incurred by the attacker for the duration of the attack. In Bitcoin mining, that cost is predominantly the electricity required to run ASICs for a period of time. Certainly, there are other operational expenditures associated with mining, such as cooling, facility rentals, maintenance, and personnel, among many others. However, given limited data publicly available on these other operational costs, we will only account for electricity consumption over time in this analysis as the primary driver of OpEx for Bitcoin.

As covered in Table 1, there is a wide variation across Bitcoin ASIC models when it comes to electricity consumption and overall efficiency. As such, the attacker's electricity OpEx will vary accordingly as it depends on the composition of machines used in the attack. For consumption estimates, we use the manufacturer's consumption figures per device. In both the aforementioned "naive" and "market" approaches to estimating CapEx, the assumption was that the attacker would source 51% of all machines in operation. To estimate the total electricity used in this scenario, we follow the same approach and simply multiply each machine purchased by its respective consumption.

Once we have the total electricity consumed per hour of attack, we must now estimate how much that electricity would cost. Given the wide variability of electricity costs globally, we price that electricity based on a global average. Data on the cost of electricity in 147 countries enabled us to calculate a global average of USD 0.15 per kilowatt hour as of March of 2023 [40]. This average cost was then applied to the total electricity consumed per hour of attack to estimate OpEx.

Table 2: OpEx of Running 51% of Bitcoin ASICs (Dec-2023)

Attack Duration (h)	Total Electricity (kW)	OpEx (USD)
1	10,144,401	$1,521,660.20
2	20,288,803	$3,043,320.40
3	30,433,204	$4,564,980.61
4	40,577,605	$6,086,640.81
5	50,722,007	$7,608,301.01

The costs showcased in Table 2 account for a mix of all major machines and apply to both "naive" and "market" approaches.

In the manufacturing scenario, where the attacker is producing replicas of the S9 and adding hashrate to the network, OpEx is substantially higher. Once again we should highlight that, although the S9 is the most plausible candidate to be reverse-engineered for the reasons presented in the previous section, it is the least efficient ASIC still in operation.

Table 3: OpEx in Manufacturing Scenario (Dec-2023)

Attack Duration (h)	Total Electricity (kW)	OpEx (USD)
1	52,045,161	$7,806,774.11
2	104,090,321	$15,613,548.21
3	156,135,482	$23,420,322.32
4	208,180,643	$31,227,096.42
5	260,225,804	$39,033,870.53

6.1.6 Total Cost to Attack Bitcoin (Capex + Opex) for 1 hour

Now that we have presented the costs associated with CapEx and Opex, let us now aggregate the two and provide a Total Cost to Attack (TCA) for Bitcoin under these different scenarios.

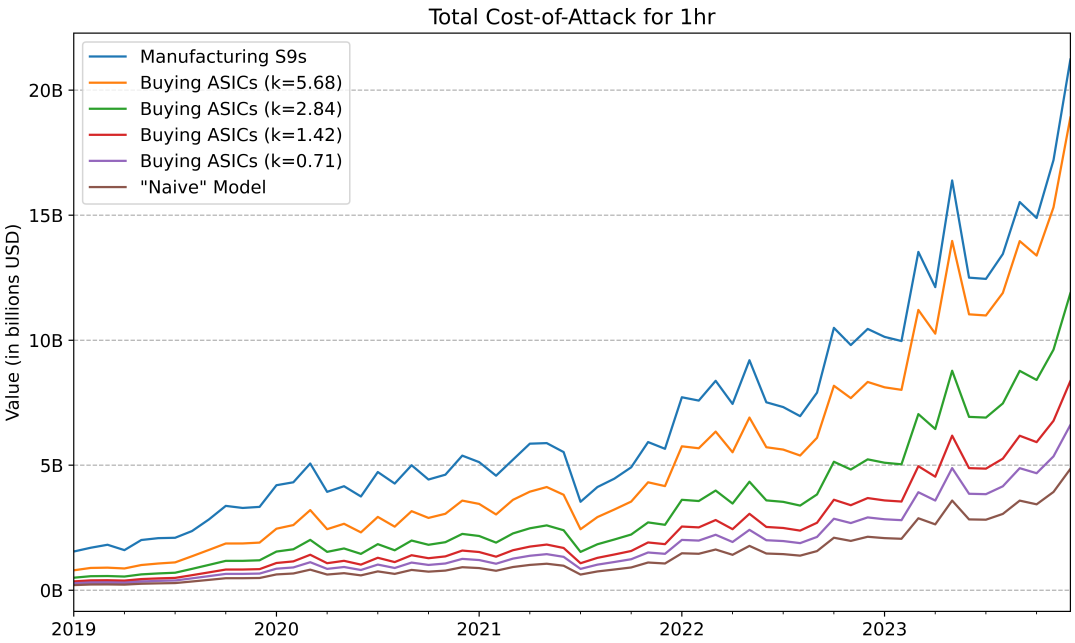

Figure 7: Total Cost to Attack (TCA) Bitcoin, All Scenarios, USD

6.2 Applying it to Ethereum

6.2.1 Capex: ETH

After The Merge [41], Ethereum's migration to Proof-of-Stake, the network's primary Sybil-protection mechanism became ETH itself. Ethereum block producers must post chunks of 32 ETH worth of collateral to be able to participate in the consensus system, generally called the *Consensus Layer*. Each chunk of 32 ETH is called a *validator*, which means a unit of block producer in Ethereum's nomenclature. Once confirmed, each validator unit is added to a lottery that determines the sequence of future block producers in intervals of around 30 minutes. The more validator units a single entity possesses, the more lottery tickets it will be entitled to and consequently the more blocks it will be able to produce.

An interesting analogy for the financialization of Ethereum consensus is the issuance of sovereign debt. Just like how nation-states can borrow funds from its population to build critical infrastructure through bonds, Ethereum can borrow ETH from its users to finance its security through staking. By borrowing funds from validators, Ethereum makes it more expensive for its network to be attacked. In exchange, it pays validators interest by issuing ETH and makes them eligible to receive user fees. A critical difference between staking and sovereign debt, however, is that validators are not simply lending money to the network. Instead, they directly engage in block production and might lose funds if they misbehave through slashing as mentioned previously.

At face value, this system makes it easier to evaluate the cost of breaching the BFT threshold. Unlike Bitcoin, where one must estimate hashrate and ASIC composition, Ethereum staking makes that explicit. One can simply look at total ETH staked and reason about the system's security without the need for complex heuristics. While from a measurement standpoint this

appears easier, there are safeguards in place that further complicate an attack scenario. Such safeguards make attacks more time consuming and, by extension, more expensive. In the following sections, we will attempt to price these attack scenarios and estimate the total cost to attack Ethereum under different scenarios.

6.2.2 The "Naive" Approach

The "naive" approach to quantifying a 34% attack on Ethereum involves first assessing the supply held by active validators in the Consensus Layer. Active validators, as the name suggests, are validators actively participating in block production. We can then simply multiply total staked supply held by active validators, in USD terms, by 34% to get a naive CapEx. As of Dec. 31, 2023, there was 28.8M ETH (65.8B USD) staked in the Consensus Layer, leading to a nominal cost of 9.8M ETH (22.3B USD).

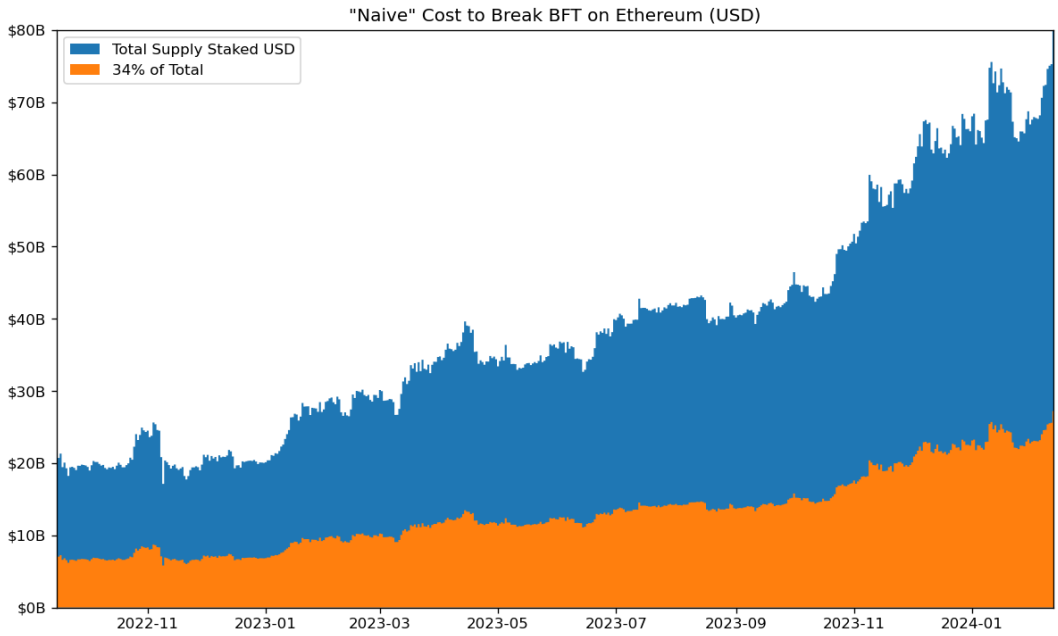

Figure 8: "Naive" CapEx, Ethereum (USD)

However, just like with Bitcoin ASICs, staked ETH is not for sale. An attacker would not be able to just purchase 34% of staked ETH to stage an attack on the network. As such, we must account for price increases and supply constraints as we evaluate how an attacker could plausibly breach that threshold by purely relying on markets. One important misconception worth addressing here is that new financial products dubbed Liquid Staking Derivatives (LSDs) do enable derivatives of staked ETH to be issued and traded. While it is true that such products may result in novel attack vectors in the future, current iterations of LSDs do not grant access to the validator's block template, which is a prerequisite to stage an attack. We will discuss the risks of LSDs more thoroughly in subsequent sections.

6.2.3 Accounting for Liquidity

Given that the attacker's CapEx in Ethereum is ETH itself, we can rely on ETH's liquidity profile over the years in our pricing model. Given that ETH trades in hundreds of markets across the world, one of the simplest ways to assess ETH available for purchase is looking at exchange addresses holding ETH. Coin Metrics has for many years clustered the addresses of centralized exchanges and tracked cumulative supply held by those addresses [42]. While the coverage of exchanges is not exhaustive, this metric can be seen as a simple proxy for ETH liquidity available across some of the major exchanges.

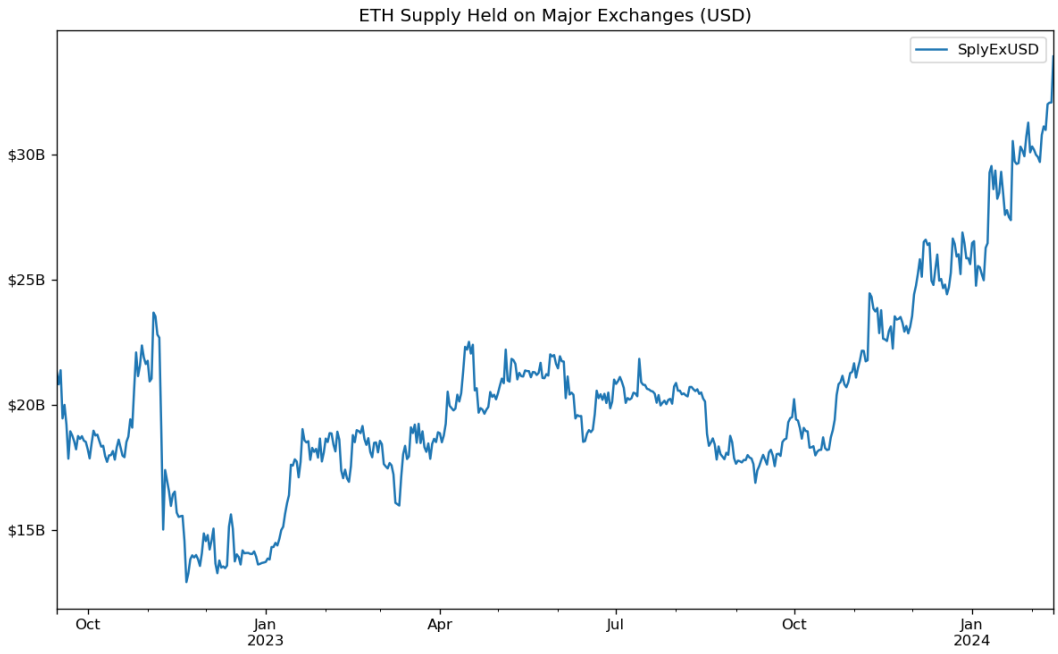

Figure 9: ETH Supply in Major Exchanges, USD (Source: Coin Metrics)

Just like in the scenario previously described where an attacker was manufacturing Bitcoin S9 ASICs, we must also account for the attacker's actions pushing Ethereum's BFT threshold higher as ETH is deployed. As the attacker begins buying and staking ETH with the intent of attacking the network, the added stake increases the 34% threshold. Mathematically, this can be described as 0.34 times the sum of the original total supply staked plus the attacker's end stake.

Recall that the "naive" attack threshold as of Dec. 31 2023 was 9.8M ETH. In order to successfully break BFT as the threshold is pushed higher, the attacker would have to attain 15.09M ETH that day, which is more than the total ETH held by exchanges covered by Coin Metrics (Bitfinex, Bitstamp, BitMEX, Binance, Bittrex, Gemini, Huobi, and Kraken). Although there are major markets missing which the attacker could use to further source ETH, chiefly Coinbase, Uniswap, as well as decentralized lending markets, it is likely that a liquidity shock would emerge before this far fetched scenario can even materialize.

6.2.4 Opex: Cloud Resources

As we turn our attention to OpEx, it's important to understand the fundamentals of staking operation. As covered previously, each validator in Ethereum is accounted for as a unit and must start with a balance of 32 ETH. When a validator is bootstrapped, validators receive two keys: a *validator key* and a *withdrawal key* [43]. Validator keys are used in the block production process where validators must verify and attest to blocks being produced. Withdrawal keys, as the name entails, enable validators to perform treasury management operations, such as the withdrawal of staked ETH. This dual-key architecture was primarily designed to enable delegation: as a holder of ETH, you can outsource the block production process without giving away custody of your ETH. This also enables validators to build more restrictive security perimeters around their withdrawal keys.

This separation of custody and block production has considerable implications when it comes to evaluating an attacker's OpEx. For starters, a single server under this architecture can be used to produce blocks on behalf of multiple validators by simply loading up their respective validator keys. Unlike Bitcoin, where highly specialized machines must be used in the consensus process, general purpose machines can be used to produce Ethereum blocks as long as they are powerful enough to support running the required node clients. Relatively inexpensive cloud resources, like AWS's m6i.2xlarge have been used to support thousands of Ethereum validators in previous benchmarks by Nethermind, one of Ethereum's consensus layer clients [44].

The key assumption in our OpEx estimate for Ethereum is that the attacker would leverage these cloud resources for the attack. Our estimates will leverage the benchmark provided by Nethermind of 1500 validators per node to simulate the cost on AWS. For compute, we assume the EC2 instance used is the m6i.2xlarge located in US East (Chicago). Given that the node will also need to host about 1.5 TB of Ethereum block-level data, we will include the cost of storage using AWS's EBS service. With these parameters in hand, we estimate the total cost of running a single node is equal to $0.74 USD per hour as per AWS's calculator [45].

Armed with an hourly cost estimate per node, we can now assess how many nodes will be required. Under the "naive" assumption, the attack threshold as of Dec. 31 2024 was 9.8M ETH, which at 1500 validators per node would require 204 nodes $\left(\frac{9{,}814{,}997 \text{ ETH}}{32}\right)/1500$. In the aforementioned scenario where the attacker is buying ETH and adding it to the consensus layer and pushing the threshold higher, the attacker would require 314 nodes $\left(\frac{15{,}088{,}787 \text{ ETH}}{32}\right)/1500$. At $0.74 USD per node per hour ($540 USD per node per month), we can now estimate OpEx under different attack durations.

Attack Duration	Naive OpEx	Added ETH OpEx
1 hour	$150.96	$232.36
1 day	$3,623.04	$5,576.64
1 week	$25,361.28	$39,036.48
1 month	$108,691.20	$167,299.20
1 year	$1,322,409.60	$2,035,473.60

Table 4: OpEx under different attack durations.

6.2.5 Accounting for Churn

As mentioned previously, OpEx is a function of attack duration; the longer the attack, the more it will cost. One factor that would drastically increase the time it would take for an attacker to stake ETH and operationalize 34% of validators is the network's so-called *churn limit* [46]. This is a consensus-enforced rule that governs how many validators can enter or exit the consensus layer. At its core, the churn limit was designed to prevent abrupt entries and exits and ultimately make Ethereum's consensus layer more stable in the long run.

The churn limit is enforced at every epoch, which is the way Ethereum keeps track of time. One epoch equates to up to 32 blocks, which translates to about 6.4 minutes. While the churn limit is fixed within those intervals, it can go up or down as a function of how many validators are active in the consensus layer. At the lowest validator threshold, which is less than 327,680 active validators, only 4 validators can enter or exit the consensus layer per epoch (900 per day). The more validators that are active in the network, the higher the allotted churn rate per epoch.

When assessing the attack duration for OpEx, we must account for the churn rate increasing as the attacker adds validators to the network. We must also account for the fact that, given the attacker would have to add a substantial number of validators, a long queue will form. This queue will require the attacker to continuously run node infrastructure, which increases the cost and difficulty of attack given how many nodes would have to be run concurrently. We can represent this dynamic mathematically below.

Let us create a function for N validators that allows us to consider the following characteristics:

- Given N active validators, we ought to control $\frac{1}{3}$ of the network, which implies we need M validators to satisfy $\frac{M}{N+M} = \frac{1}{3}$ which implies $M = \frac{N}{2}$
- Given we need to activate M validators, this implies we must enter the queue at a rate of $M \times q$, where q is the rate at which new validators are activated.
  - The churn limit is defined in terms of the number of active validators N, where the churn limit per epoch is determined by:

$$\text{Activation Rate}(N) = \begin{cases} 4 & \text{if } N \leq 327,680 \\ 4 + \left\lfloor \frac{N-327,680}{65,536} \right\rfloor & \text{if } N > 327,680 \end{cases}$$

  - The activation rate starts at 4 validators per epoch when N is less than or equal to 327,680.
  - For every additional 65,536 active validators beyond 327,680, the activation rate increases by one.
  - To get the daily number of validators that can enter the validator pool, we can simply multiply by the number of epochs per day, 225.

Given we want to introduce $\frac{1}{2}N$ validators, and given a function for the churn limit of validator activation as defined above,

$$\text{Epochs Required} = \frac{\frac{N}{2}}{\text{Activation Rate}(N)}$$

for $N > 327,680$, we have

$$\text{Epochs Required} = \frac{N}{2 \cdot \left(4 + \left\lfloor \frac{N-327,680}{65,536} \right\rfloor\right)}$$

Given that there are currently active validators, we can estimate the queue duration in number of epochs as:

$$\text{Epochs Required} = \frac{899,840}{2 \cdot \left(4 + \left\lfloor \frac{899,840-327,680}{65,536} \right\rfloor\right)}$$

Now, calculate the value:

$$\left\lfloor \frac{899,840 - 327,680}{65,536} \right\rfloor = \left\lfloor \frac{572,160}{65,536} \right\rfloor = 8$$

Now, calculate the final result:

$$\text{Epochs Required} = \frac{899,840}{2 \cdot (4+8)} = \frac{899,840}{2 \cdot 12} = \frac{899,840}{24} \approx 37,493.33$$

As of Dec. 31, 2023, there were 899,840 validators active in Ethereum's consensus layer ($N = 899,840$). As per our estimate, it would take the attacker approximately 37,493 epochs to introduce $\frac{N}{2}$ validators, assuming the attacker is the only one adding ETH to the consensus layer. That is roughly equivalent to 166 days the attacker would have to wait, which results in an OpEx of $929,266 USD. It's important to note that this duration and resulting cost would further increase if EIP-7514 [47] were to be adopted as it would implement a maximum churn limit of 8 validators per epoch.

6.3 Total Cost to Attack Ethereum

Given the dynamics of the churn limit, Ethereum's Total Cost to Attack is challenging to be represented as a time series because, unlike Bitcoin, a single attack would elapse many days. On the CapEx side, it can be simply defined as a function of the price of ETH and total amount that the attacker must stake. However, on the OpEx side, it would be a function of cloud computing costs over a longer time period depending on the count of validators active when the attack begins.

Applying to Dec. 31, 2023, with an ETH price of $2,279 USD, the total amount of ETH staked at 28.8M ETH, and a validator count of 899,840 validators, we estimate it would cost the attacker 34.39B USD to 34% attack the network. If the attack were to start December 31, 2023, it would take the attacker until June 14, 2024 to breach the 33% threshold.

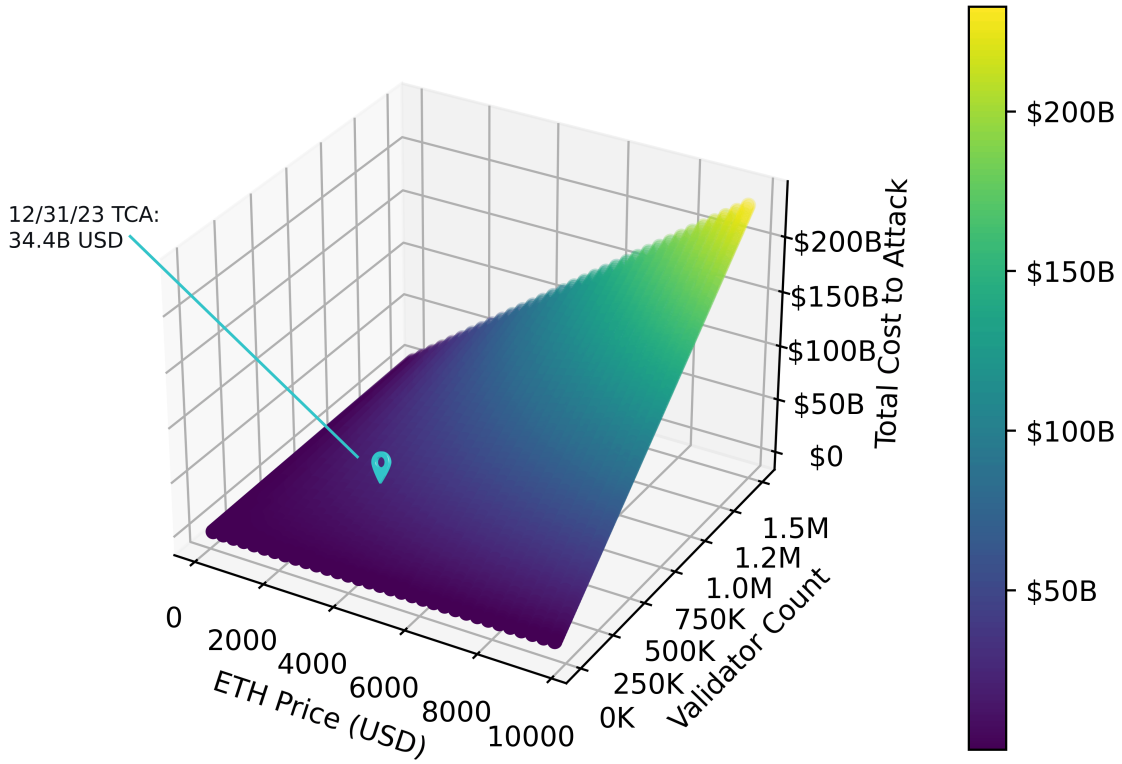

Figure 10: Total Cost to Attack (TCA) Ethereum

7 Qualifying Attacks

Now that we have quantified the cost to attack Bitcoin and Ethereum, we shall now evaluate the feasibility of such attacks, considering two primary motivations: monetization, where the attacker seeks to financially profit from the attack, and ideological, where the attacker intends to disrupt or destroy these networks for philosophical purposes.

7.1 Motivation: Monetization

The most obvious way to monetize a BFT breach is via a double spend attack. However, in order for a double spend attack to be profitable, the proceeds from the attack must obviously exceed its cost. Earlier, we presented scenarios where the cost to attack either Bitcoin or Ethereum could surpass $20B USD. That means that in order for an attack to be profitable, the attacker must find a victim to double spend at least that amount. So in order to make a $1B USD profit, the attacker would need to possess an additional balance of $21B USD. To put that figure into perspective, on Dec. 31st 2023, the entire reported spot volume of BTC was$5.19B USD. In the entire year of 2023, there were only 13 days where BTC's total spot volume exceeded $20B. It is highly implausible that a crypto exchange, or any third party

for that matter, would facilitate trading at this scale within 6 confirmations for BTC, or 12 confirmations for ETH and its tokens.

Another potential monetization approach could be via a Time Bandit Attack where the attacker reorgs the chain to extract MEV from the past. Given MEV is a phenomenon predominant in Ethereum, this strategy is mostly relevant to this network. In order to evaluate whether this could be a viable monetization strategy, let us look at an extreme scenario: what if the attacker was able to go back in time and extract all MEV profits as well as block rewards during the most profitable month in terms of MEV (and block rewards) in the entire history of Ethereum? That month, May of 2021, saw the launch of several NFT projects and record volumes in Decentralized Exchanges [48]. Even in the most favorable scenario imaginable, the cumulative block rewards and MEV as per Coin Metrics and MEV explore would total $674,891,881 and not be enough to cover the attacker's expenses [26].

As mentioned previously, reorgs are not the only way to monetize a breach in a network's BFT threshold. What if the attacker took over the network and stopped mining transactions unless users paid exorbitant fees? In this scenario, the attacker's ability to coerce users to pay for higher fees is a function of the elasticity of demand for block space. Historically, in times of congestion, EIP-1559's mechanism increases fees programmatically and is successful in disincentivizing chain usage [15]. This is evidence that block space tends to be elastic as users simply stop transacting when fees are too high. While there are unconditional buyers of blockspace that would oblige, the presence of alternative EVM-based blockspace would drastically hurt the attacker's ability to profit from fee market manipulation at the scale required to stage this attack.

In none of the hypothesized attacks presented here the attacker would be able to profit by attacking Bitcoin or Ethereum. Consider that even in the most profitable double spend scenario presented, where the attacker could potentially make $1B after spending $40B, that would account for a 2.5% rate of return. In the case of Ethereum, where the attack would take 6 months to perpetrate, the attacker's annualized rate of return would be 5%. To contextualize this figure, consider that a 5% yield is on par with current US treasury yields [49]. That also presupposes the attacker would be able to resell CapEx items (ETH, or ASICs) for at least the purchase price, which will probably not be the case after the attack. In light of the attack's costs relative to expected returns, along with all of the other practical reasons provided here, we find it highly unlikely for the monetization of the attack to be viable, let alone rational.

7.2 Motivation: Ideological

In the scenario where an attacker was not pursuing monetization but instead to disrupt or destroy these networks, a few additional factors must be considered. Given the resources required, the attacker would need to be a well-capitalized entity akin to a nation-state. Maintaining privacy around the attack would arguably be difficult in such a scenario. As with other forms of cyberattacks, tracing methods can be employed to gather intel on the attacker leading to potential deanonymization. The same is true for both 51% and 34% attacks, especially at this scale where the attacker is unable to leverage rental or lending marketplaces for ASICs, or crypto exchanges for ETH. Due to the global nature of cryptoassets there exists a risk of political backlash and potential unrest if the perpetrator were to be identified as a nation-state or corporation directly involved in an attack.

Arguably, the most plausible ideological attack motivator for a nation-state would be the perception of these networks as competitive to its own monetary system. As open source, nationless alternatives to systems like Central Bank Digital Currencies (CBDCs), nation states might become motivated to permanently disable these systems. However, this gets into a fundamental issue with regard to 51% attacks: it is very difficult to make them permanent. Even in a scenario where attackers simply mine empty blocks and perpetually attack, the network can fight back. Techniques such as block checkpointing and User Activated Soft Forks (USAFs) have been used as countermeasures to forcibly exclude the attacker's version of the chain in most previous instances of 51% attacks [3], [21], [50], [51].

An ideologically-driven attacker would also have to consider the threat of retaliation. The advent of slashing makes it so that networks like Ethereum can retaliate via a protocol-enforced deletion of all validators controlled by the attacker. We argue that, in the manufacturing scenario, the same could be true for Bitcoin in such extreme circumstances. As mentioned previously, MINE-MATCH enables the ASIC that produced a block to be identified due to patterns in a block's nonce that are intrinsic to that ASIC's chip design. If this type of validation were to be implemented at the consensus level, it could be used to slash the attacker's class of ASICs. Unlike Ethereum, however, this would entail some collateral damage as legitimate users of that ASIC model would also be slashed. Nevertheless, the network could plausibly perform a USAF to invalidate the model and render the attacker's stockpile of ASICs useless.

The impossibility of full destruction could perhaps be in and of itself a deterrent to ideological attacks in networks like Bitcoin and Ethereum. In the short term, the attacking party would undoubtedly inflict harm as conflicting branches may wreak havoc across the network. However, as witnessed with previous attacks, there exists a social consensus layer backing these networks that enables individual network participants to fight back [52]. Even the most resourceful attacker would be unable to beat either a protocol-enforced slashing, or a socially-enforced hardfork that ultimately invalidates the attack. This would be particularly detrimental to the attacker since, at every round of attack, resources are destroyed.

In evaluating the motivations of an ideologically-driven attacker, especially as it relates to perceived competition with initiatives such as CBDCs, it is likely that Bitcoin and Ethereum would be evaluated as one technology. As such, the mere existence of two distinct sources of secure blockspace, each with its own advantages and drawbacks, may act as a deterrent. In other words, one network makes the other less susceptible to being targeted by an attacker who views them as part of the same technology. A diversity of consensus mechanisms not only enhances the resiliency and optionality of secure block space but also underpins the development of blockchains as defensive technologies, ultimately enabling the creation of economic and social frameworks free from the influences of centralizing forces [53].

8 Implications for Declining Subsidies

We believe quantifying and qualifying how impractical such an attack would be on Bitcoin and Ethereum has interesting implications when it comes to the monetary policy of crypto assets. Both Bitcoin and Ethereum employ deflationary policies that, over the long run, push their respective supply curves downwards. In Ethereum, this was successfully achieved via the implementation of EIP-1559, which takes ETH out of circulation everytime users transact. In Bitcoin, this was baked into its protocol from genesis whereby new coins are issued at every block with the issuance rate declining every four years in an event called the "halving". That

issuance schedule functions as a subsidy to incentivize block production and was designed to decrease until a 21M BTC cap was reached—at which point the supply curve can only go down as coins are destroyed or lost.

Declining subsidies have been a particularly salient concern in the context of Bitcoin's long term security and its perceived susceptibility to attacks [54], [55]. The concern relates to Bitcoin's security depending on subsidies being replaced by user fees. These are fees users pay directly to miners to ensure their transactions are added to their block templates. Mechanistically, user fees are priced on the basis of a first-price auction in times of congestion where the highest bidder has the highest chance to be included in a block. Fluctuations in demand for block space make these fees highly cyclical in both Bitcoin and Ethereum contributing to the anxiety around whether fees alone could finance Bitcoin's security.

The implicit assumption that justifies such concerns is that Bitcoin fees correlate with security. The higher the fees collected by miners, the more secure the network is. Surprisingly, while this intuitively makes sense, we found this not to be a phenomenon observed historically. Other factors influence the behavior of miners and appear to challenge this assumption. In fact, when performing a linear regression to understand the relationship between Bitcoin fees aggregated monthly with that month's corresponding Tocal Cost-to-Attack (a proxy for security), we found low correlation between these data sets, with Peason correlation coefficients ranging from 0.361931 (k=5.68) to 0.384199 (Manufacturing S9s); suggesting a weak linear relationships between these two variables.

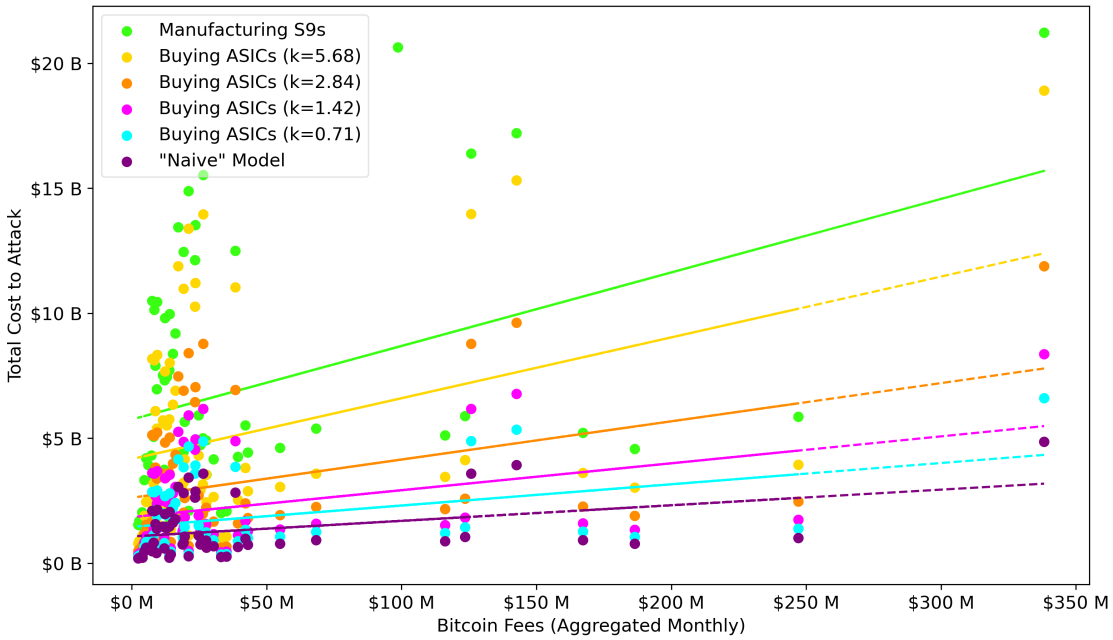

Figure 11: Linear Regression of Bitcoin Fees vs. Total Cost to Attack (TCA)

Another way to visualize this distribution is via a Kernel Density Estimate (KDE) plot. This makes the horizontal cluster more easily discernable. The shape of the cluster indicates how there are times where fees are low, but security as measured through TCA is high.

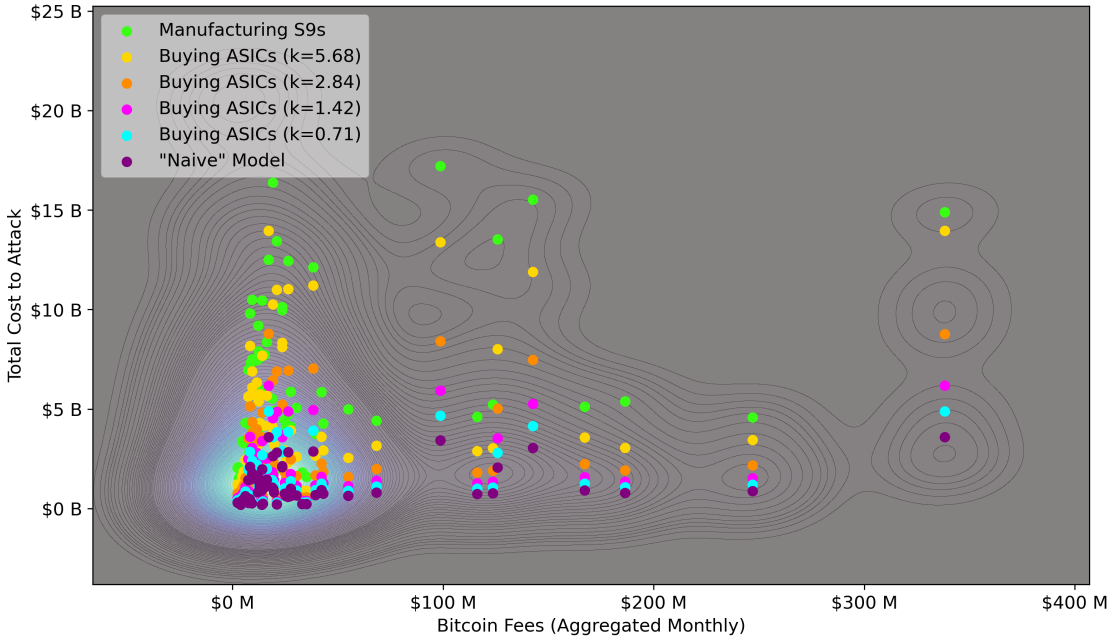

Figure 12: Kernel Density Estimate (KDE) of Bitcoin Fees vs. Total Cost to Attack (TCA)

In reality, there have been multiple instances in the network's history where fees are low and trending downwards but network security is high and trending upwards. We hypothesize this is due to the immense cyclicality of fees which, when coupled with the volatility of the price BTC itself, affects how miners make key decisions. Consider that the process of procuring, warehousing and deploying hashrate can be very lengthy. Because of that, industrial-scale miners must speculate and continuously deploy hashrate. That ensures they remain competitive in seemingly unpredictable high-fee cycles because that is when they see outsized returns. If they were to wait for a high-fee environment to acquire hashrate, by the time that hashrate is online the cycle might be over.

This historically-observed behavior shows how miners implicitly speculate on fee cycles, thereby making the relationship between fees and security more unpredictable.

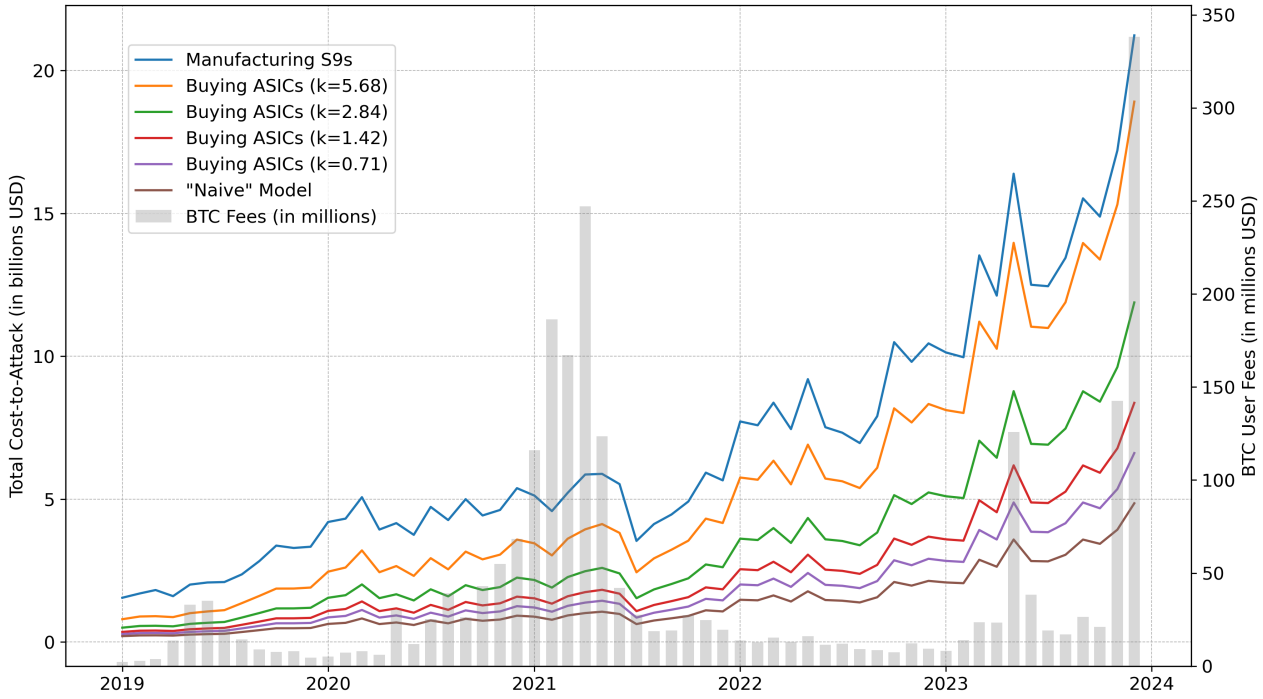

Figure 13: Timeseries: Bitcoin Fees (Monthly) vs. Total Cost to Attack (TCA)

The behavior of miner wallets onchain supports the hypothesis that miners are speculators [56]. Miners are one of the easiest types of entity to track on-chain given they are recipients of freshly minted coins with no previous transactions. One of the ways to track their behavior is by looking at the outflow of BTC from wallets one transaction away from mining pools. This serves as a proxy for selling behavior of individual miners, rather than mining pools. Remarkably, the outflows from miner wallets is highly correlated with market cycles, suggesting miners accumate BTC with the expectation of selling it in future bull markets.

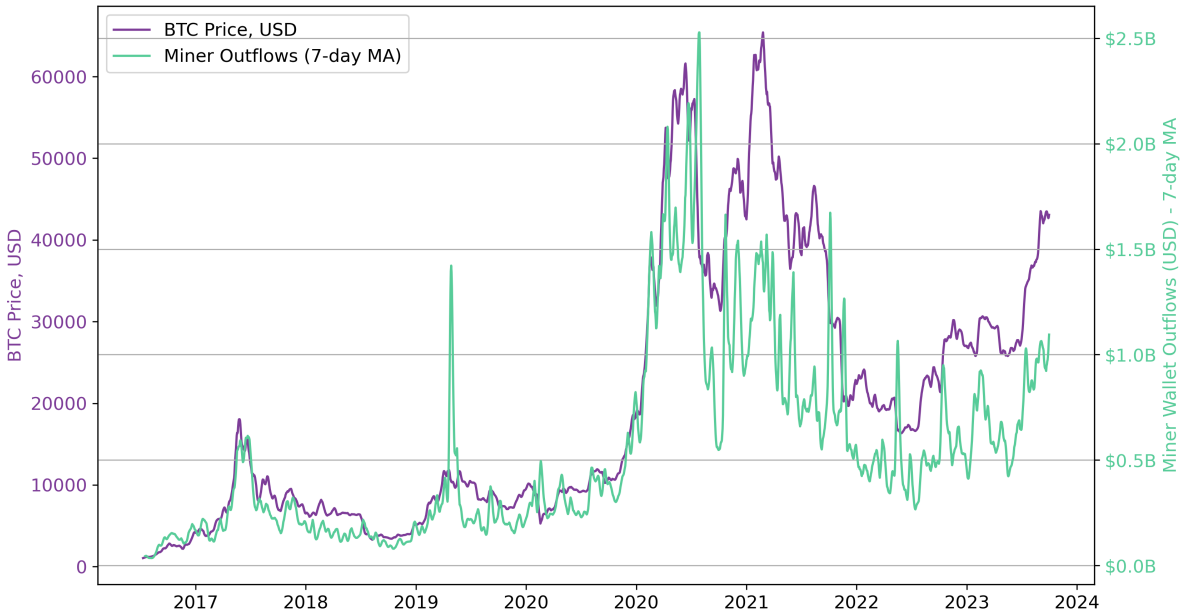

Figure 14: Bitcoin Miner Outflows (1-hop) vs. BTC Price (USD), Source: Coin Metrics

9 Discussion: The Remaining Attack Vectors

As demonstrated thus far, the security of Bitcoin and Ethereum has reached a point where the costs and associated risks of attacking these networks via 51% and 34% attacks vastly outweigh the benefits. This is indicative of Nash Equilibria, wherein attacks become unattractive compared to other behaviors, such as participating legitimately in block production, or refraining from attacking altogether. While this is a significant achievement that attests to the maturity of these networks, we would like to now shift the focus to more realistic attack vectors that warrant vigilance. It is important to note that the issues discussed herein are not unsolvable and should not dilute the likely achievement of Nash Equilibrium in Bitcoin and Ethereum at the BFT level. Nevertheless, we argue these are pressing issues that, if resolved, would significantly strengthen the security of these networks beyond BFT attacks.

9.1 Centralization of Block Templating

Block producers must decide which transactions will be in their blocks. This activity is called Block Templating and it is one of the most critical aspects of block production. From an economic perspective, the optimization of block templates for high-fee transactions is the most important driver of the block producer's revenue. From a security perspective, the entity or algorithm involved in templating a block is granted a substantial amount of power as it determines which transactions get settled. With the industrialization of block production in both Bitcoin and Ethereum, block templating became a specialized practice. This resulting specialization in block production has created unique attack vectors in both networks that we believe are more realistic than BFT attacks.

9.2 Bitcoin Mining Pools

As the name suggests, Bitcoin mining pools were designed to pool the resources of multiple miners as a way to increase their individual competitiveness. Mining pools today are enshrined with three key responsibilities: aggregate the hashrate of a group of miners, template the block everyone will be working on, and disburse funds to all constituents based on their contributions. While mining pools have historically charged a fee for this service, many are engaging in additional strategies to maximize their rewards. Such strategies have contributed to a level of centralization in Bitcoin mining pools and have created unique risks.

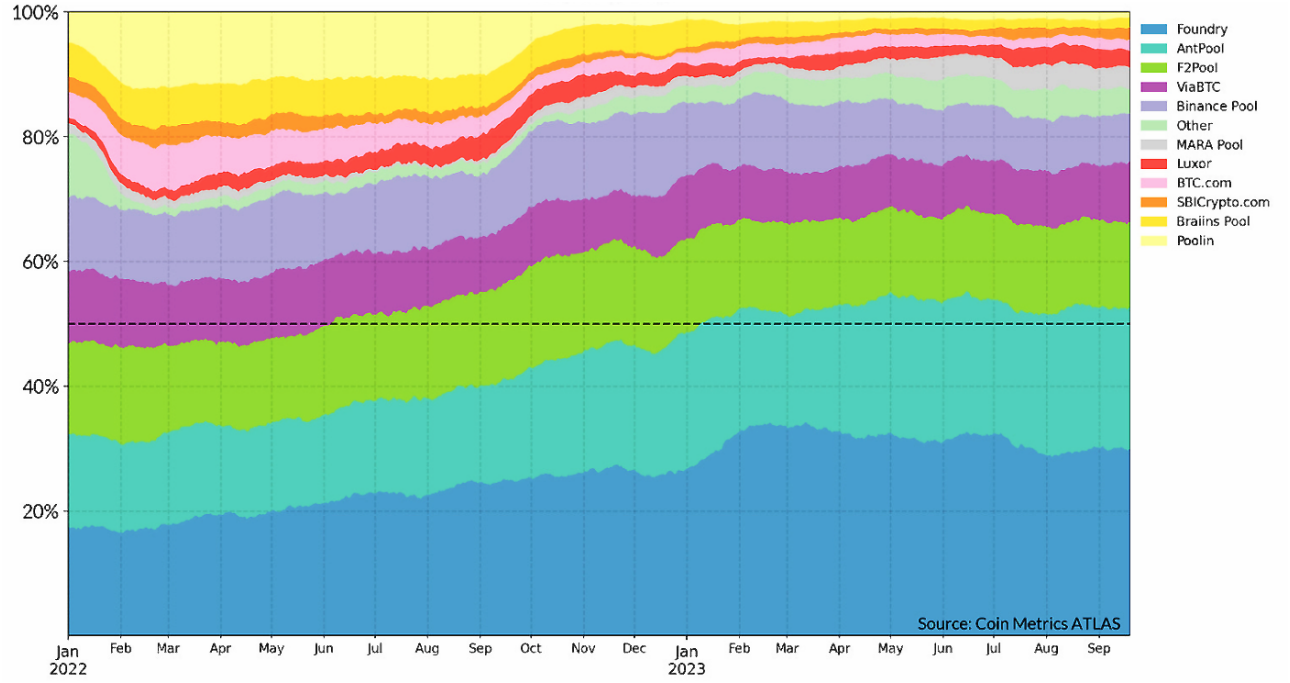

Figure 15: Bitcoin Mining Pool Dominance, Source: Coin Metrics ATLAS

Chief among the risks of centralization of mining pools is the implicit centralization of block templates. Mining pool constituents, the entities actually engaged in mining, are simply fed a block template to work on by the pools they participate in. Most mining pools operate under Stratum V1, a messaging protocol that allows for the coordination between the pool and all of its constituents [57]. Under this protocol, miners can technically see whether the pool is mining an empty block or if it will reorg the chain, but there is little evidence individual miners are checking for that. Importantly, miners cannot see the transactions they are working and that enables pools to engage in overt MEV strategies.

We believe discussions focused on the development, funding, and adoption of new mining protocol designs such as Stratum V2 are significantly more constructive than the more frequently-encountered rhetoric about 51% attacks. MEV is a phenomenon that started in Bitcoin in 2016 with products such as ViaBTC's transaction accelerator, a service that enables users to pay for network fees directly to the mining pool [58]. Today, MEV in Bitcoin has a much broader scope. Pools like F2Pool are increasingly engaging in creative strategies and operating what appears to be a proprietary MEV desk [59]. Embracing and decentralizing MEV could be key to reducing the risks of centralization at the block template level and meaningfully increase Bitcoin's security. That is especially true if MEV infrastructure in Bitcoin takes lessons from Ethereum and is designed in a way to prevent censorship.

9.3 Ethereum MEV and Liquid Staking Derivatives

Ethereum has the most developed and sophisticated MEV infrastructure of any crypto network at this time. As such, being the ecosystem where MEV was professionalized and matured, Ethereum is concurrently grappling with the negative externalities and risks that may be inherent byproducts of MEV. While not nearly as destructive as 34% attacks, there are ways such risks can materialize and negatively impact users. Namely, because templates are decentralized in Ethereum and sold to the highest bidder, it has become cheaper to perpetrate certain censorship attacks via MEV infrastructure.

Recall that under EIP-1559 there are two types of fees: base fees, which are destroyed; and priority fees which reward the block producer. With the decentralization of block template building as a byproduct of MEV, priority fees put a price on censoring transactions in Ethereum. To demonstrate this, in 2023 researchers with the Special Mechanisms Group (SMG) were able to produce an empty Ethereum block by simply bidding the highest priority fee which at the time was 0.05 ETH [60]. While this strategy would naturally cost a lot more in times of congestion, it demonstrates the potential dangers of selling block templates to the highest bidder without a level of accountability with regards to block composition.

Interestingly, just like in Bitcoin mining, there lacks a mechanism in Ethereum for block producers to verify if the templates they are using align with the more subjective principles espoused by the users of these networks. This was a major point of contention over the past two years, as many block template builders began censoring OFAC-sanctioned transactions [61]. Naturally, compliance is a consequence of specialization as many template builders aspire to become large businesses. Nevertheless, there needs to be an accountability mechanism for these entities so that, like mining pools, their power over block templates does not overreach.

Another trend reflective of the specialization of block production that warrants caution relates to the rise of Liquid Staking Derivatives (LSDs). Even though LSDs do not currently enable their holder to impact block templating (i.e. holders of the popular stETH LSD have no say on block composition), they add a degree of separation between custody and staking operations that must be monitored. Lido, for example, currently retains over 30% of staked ETH [62]; a figure very close to the network's BFT threshold. Participants of the LidoDAO select through a governance process who the actual block producers managing that stake will be and there are currently 36 entities involved [63]. Although no single entity managing Lido's validators has over 5% of block production power, this makes LidoDAO's governance a potential attack vector and increases the urgency to decentralize Lido constituents themselves via technologies like DVT [64].

10 Conclusion

In conclusion, this paper constitutes a new analytical framework to assess the cost and viability of potential attacks on Bitcoin and Ethereum. By constructing a model to estimate the costs associated with executing 51% and 34% attacks, we delve deep into the economic incentives that safeguard these networks against potential threats. This approach not only highlights the financial impracticality of such attacks but also brings to light the strategic challenges that would-be attackers would undertake, weighing the expected utility against the costs and risks involved.

Crucially, our analysis compares and contrasts the motivations behind such attacks, distinguishing between profit-driven and ideologically-driven adversaries. This is an important distinction, as it underscores the multifaceted nature of threats facing blockchains as they reach mass adoption, ranging from mere financial gain to more complex ideological objectives. Through this lens, we demonstrate how the security of Bitcoin and Ethereum has evolved to a point where the costs and risks associated with attacks far outweigh any potential benefits. This equilibrium, indicative of a Nash Equilibrium, suggests that adversarial actions become unattractive when compared to other strategies, such as honest participation in the network or abstention from attacking.

Furthermore, our findings challenge the conventional wisdom regarding the relationship between deflationary monetary policies, user-generated fees and the security of blockchain networks. By dissecting the dynamics of fee revenue and miner incentives, we reveal a nuanced picture of how security is maintained. Specifically, we argue that Bitcoin's security model transcends the simplistic correlation with immediate fee revenues, instead relying on miners' speculative investment in the network's future. This speculative behavior, likely driven by long-term price expectations, acts as a bulwark against attacks, effectively increasing the network's security through a proactive, continuous addition of hashing power.

We hope this paper solidifies the understanding of the economic underpinnings of blockchain security and sets a new benchmark for future research in this area. As Bitcoin and Ethereum continue to mature, our findings underscore the importance of a holistic approach to evaluating network security, one that considers both the technical and economic dimensions of blockchains as they exist today.

11 Acknowledgments

We would like to thank Karim Helmy, Nic Carter, Hasu, Antoine Le Calvez, Parker Merritt, Tanay Ved, and Alex Thorn, for conversations and feedback that helped shape this paper.

References

[1] E. Budish, *The economic limits of bitcoin and the blockchain*, Accessed on 12/13/2023, 2018. [Online]. Available: https://bfi.uchicago.edu/wp-content/uploads/2022/07/Economic-Limits-of-Bitcoin.pdf.

[2] Hasu, J. Prestwich, and B. Curtis, *A model for bitcoin's security and the declining block subsidy*, Accessed on 12/14/2023, 2019. [Online]. Available: https://uncommoncore.co/wp-content/uploads/2019/10/A-model-for-Bitcoins-security-and-the-declining-block-subsidy-v1.01.pdf.

[3] J. Lovejoy, "An empirical analysis of chain reorganizations and double-spend attacks on proof-of-work cryptocurrencies," Accessed on 12/15/2023, Ph.D. dissertation, Massachusetts Institute of Technology, 2019. [Online]. Available: https://dspace.mit.edu/bitstream/handle/1721.1/127476/1193019932-MIT.pdf?sequence=1&isAllowed=y.

[4] K. Helmy, L. Nuzzi, A. Mead, and K. Waters, *The signal & the nonce: Tracing asic fingerprints to reshape our understanding of bitcoin mining*, Accessed on 12/13/2023, 2020. [Online]. Available: https://coinmetrics.io/special-insights/bitcoin-nonce-analysis/.

[5] *Mine-match: Miner identification via nonce expectation maximization and traceable chip heuristics*, Accessed on 12/13/2023, 2020. [Online]. Available: https://labs.coinmetrics.io/.

[6] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982, Accessed on 12/13/2023. [Online]. Available: https://lamport.azurewebsites.net/pubs/byz.pdf.

[7] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, Accessed on 12/13/2023, USENIX Association, 1999, pp. 173–186. [Online]. Available: https://pmg.csail.mit.edu/papers/osdi99.pdf.

[8] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, Accessed on 12/13/2023, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[9] N. Carter, *It's the settlement assurances, stupid*, Accessed on 12/13/2023, 2020. [Online]. Available: https://medium.com/@nic__carter/its-the-settlement-assurances-stupid-5dcd1c3f4e41.

[10] S. Sayeed and H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Applied Sciences*, vol. 9, no. 9, p. 1788, 2019, Accessed on 12/13/2023. DOI: 10.3390/app9091788. [Online]. Available: https://www.mdpi.com/2076-3417/9/9/1788.

[11] G. Pırlea and I. Sergey, "Mechanising blockchain consensus," in *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, Accessed on 12/13/2023, Association for Computing Machinery, 2018, pp. 78–90. DOI: 10.1145/3167086. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3167086.

[12] *Bitcoin wiki: Value overflow incident*, Accessed on 12/13/2023, 2010. [Online]. Available: https://en.bitcoin.it/wiki/Value_overflow_incident.

[13] *Vulnerability details: Cve-2017-14451*, Accessed on 12/13/2023, 2017. [Online]. Available: https://www.cvedetails.com/cve/CVE-2017-14451/.

[14] T. Roughgarden, *Transaction fee mechanism design for the ethereum blockchain: An economic analysis of eip-1559*, Accessed on 12/13/2023, 2020. [Online]. Available: https://arxiv.org/abs/2012.00854.

[15] Y. Liu, Y. Lu, K. Nayak, F. Zhang, L. Zhang, and Y. Zhao, "Empirical analysis of eip-1559: Transaction fees, waiting times, and consensus security," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, Accessed on 12/13/2023, Association for Computing Machinery, 2022, pp. 2099–2113. DOI: 10.1145/3548606.3559341. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3548606.3559341.

[16] K. Kulkarni, T. Diamandis, and T. Chitra, *Towards a theory of maximal extractable value i: Constant function market makers*, Accessed on 12/13/2023, 2022. [Online]. Available: https://arxiv.org/abs/2207.11835.

[17] *Randao: A dao working as rng of ethereum*, Accessed on 12/13/2023, 2021. [Online]. Available: https://github.com/randao/randao.

[18] C. Pinzón and C. Rocha, "Double-spend attack models with time advantage for bitcoin," *Electronic Notes in Theoretical Computer Science*, vol. 329, pp. 79–96, 2017, Accessed on 12/13/2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157106611630113X.

[19] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," *IEEE Access*, vol. 7, pp. 28 712–28 725, 2019, Accessed on 12/13/2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8653269.

[20] B. Edgington, *Upgrading ethereum*, Accessed on 12/13/2023, 2022. [Online]. Available: https://eth2book.info/capella/part2/consensus/lmd_ghost/.

[21] L. Nuzzi, *The deepest reorg in blockchain history*, Accessed on 12/13/2023, 2022. [Online]. Available: https://www.lucasoncha.in/research/verge.

[22] I. Eyal and E. G. Sirer, *Majority is not enough: Bitcoin mining is vulnerable*, Accessed on 12/13/2023, 2014. [Online]. Available: https://arxiv.org/abs/1311.0243.

[23] S.-N. Li, C. Campajola, and C. J. Tessone, *Twisted by the pools: Detection of selfish anomalies in proof-of-work mining*, Accessed on 12/13/2023, 2022. [Online]. Available: https://arxiv.org/abs/2208.05748.

[24] M. Bahrani and S. M. Weinberg, *Undetectable selfish mining*, Accessed on 12/13/2023, 2023. [Online]. Available: https://arxiv.org/pdf/2309.06847.pdf.

[25] P. Daian, S. Goldfeder, T. Kell, *et al.*, *Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges*, Accessed on 12/13/2023, 2019. [Online]. Available: https://arxiv.org/abs/1904.05234.

[26] *Mev-explore v1*, Accessed on 12/13/2023, 2021. [Online]. Available: https://explore.flashbots.net/.

[27] *Ethereum consensus client spec*, Accessed on 12/13/2023, 2021. [Online]. Available: https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/validator.md#proposer-slashings.

[28] L. Kovalchuk, D. Kaidalov, A. Nastenko, O. Shevtsov, M. Rodinko, and R. Oliynykov, *Number of confirmation blocks for bitcoin and ghost consensus protocols on networks*, Accessed on 12/13/2023, 2020. [Online]. Available: http://tacs.ipt.kpi.ua/article/view/169018.

[29] E. Anceaume, A. Del Pozzo, T. Rieutord, and S. Tucci-Piergiovanni, *On finality in blockchains*, Accessed on 12/13/2023, 2020. [Online]. Available: https://arxiv.org/pdf/2012.10172.pdf.

[30] J. Neu, E. N. Tas, and D. Tse, *Short paper: Accountable safety implies finality*, Accessed on 12/13/2023, 2023. [Online]. Available: https://eprint.iacr.org/2023/1301.pdf.

[31] *Coinbase help center: Confirmations*, Accessed on 12/13/2023, 2021. [Online]. Available: https://help.coinbase.com/en/coinbase/getting-started/crypto-education/glossary/confirmations.

[32] *Kraken support: Cryptocurrencies deposit processing times*, Accessed on 12/13/2023, 2021. [Online]. Available: https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times.

[33] *Pow 51% attack cost: A collection of coins and the theoretical cost of a 51% attack on each network*, Accessed on 12/13/2023, 2021. [Online]. Available: https://www.crypto51.app/.

[34] *Nicehash rental marketplace*, Accessed on 12/13/2023, 2021. [Online]. Available: https://www.nicehash.com/my/marketplace/SHA256ASICBOOST.

[35] *Hashrateindex*, Accessed on 12/13/2023, 2021. [Online]. Available: https://hashrateindex.com/.

[36] B. Schmidt, "Ex-bitmain chip designer takes on crypto's mining goliath," *Bloomberg*, 2018, Accessed on 12/13/2023. [Online]. Available: https://www.bloomberg.com/news/articles/2018-10-08/ex-bitmain-chip-designer-takes-on-crypto-s-mining-goliath.

[37] W. Chu, Y.-W. Chang, Y.-C. Hu, and C.-C. Wang, "Thermal performance analysis and heat transfer enhancement study in an antminer mining machine," *ASME Journal of Thermal Science and Engineering Applications*, vol. 13, no. 2, p. 021 011, 2020, Accessed on 12/13/2023. DOI: 10.1115/1.4047383. [Online]. Available: https://asmedigitalcollection.asme.org/thermalscienceapplication/article-abstract/13/2/021011/1084159/Thermal-Performance-Analysis-and-Heat-Transfer?redirectedFrom=fulltext.

[38] J. Song, *Just how profitable is bitmain?* Accessed on 12/13/2023, 2018. [Online]. Available: https://jimmysong.medium.com/just-how-profitable-is-bitmain-a9df82c761a.

[39] *Bitmain s21 product page*, Accessed on 12/13/2023, 2021. [Online]. Available: https://m.bitmain.com/product/detail?pid=00020230927131349825 4Xvbhsva0713&locale=en&region=en.

[40] *World population review*, Accessed on 12/13/2023, 2024. [Online]. Available: https://worldpopulationreview.com/.

[41] K. Waters, L. Nuzzi, N. Maddrey, and M. Andrade, *Mapping out the merge*, Accessed on 12/13/2023, 2021. [Online]. Available: https://5264302.fs1.hubspotusercontent-na1.net/hubfs/5264302/coinmetrics-research_mapping-out-the-merge.pdf.

[42] *Coin metrics exchange supply metric*, Accessed on 12/13/2023, 2021. [Online]. Available: https://docs.coinmetrics.io/asset-metrics/exchange/splyexntv.

[43] *Beaconchain docs*, Accessed on 12/13/2023, 2021. [Online]. Available: https://kb.beaconcha.in/ethereum-staking/ethereum-2-keys.

[44] *Nethermind docs*, Accessed on 12/13/2023, 2021. [Online]. Available: https://docs.nethermind.io/validators/#aws.

[45] *Aws calculator*, Accessed on 12/13/2023, 2021. [Online]. Available: https://calculator.aws/.

[46] L. Collective, *Ethereum's activation and exit queues: How ethereum's activation and exit queues work*, Accessed on 12/13/2023, 2021. [Online]. Available: https://liquidcollective.io/eth.

[47] E. I. Proposals, *Eip-7514*, Accessed on 12/13/2023, 2023. [Online]. Available: https://eips.ethereum.org/EIPS/eip-7514.

[48] S. Nover, "Bored ape yacht club's metaverse nfts cost buyers $181 million in "gas" fees," *Quartz Magazine*, 2023, Accessed on 12/13/2023. [Online]. Available: https://qz.com/2161193/bored-ape-yacht-clubs-nfts-cost-181-million-in-gas-fees.

[49] *Us treasury interest rates*, Accessed on 12/13/2023, 2024. [Online]. Available: https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_yield_curve&field_tdr_date_value_month=202402.

[50] A. Zamyatin, N. Stifter, A. Judmayer, P. Schindler, E. Weippl, and W. J. Knottenbelt, *A wild velvet fork appears! inclusive blockchain protocol changes in practice*, Accessed on 12/13/2023, 2018. [Online]. Available: https://eprint.iacr.org/2018/087.pdf.

[51] L. Nuzzi, *Identifying the bsv 51% attack*, Accessed on 12/13/2023, 2023. [Online]. Available: https://www.lucasoncha.in/research/identifying-the-bsv-51-attack.

[52] J. K. Brekke, K. Beecroft, and F. Pick, "The dissensus protocol: Governing differences in online peer communities," *Frontiers in Human Dynamics*, 2023, Accessed on 12/13/2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fhumd.2021.641731/full.

[53] V. Buterin, *D/acc: Defensive (or decentralization, or differential) acceleration*, Accessed on 12/13/2023, 2023. [Online]. Available: https://vitalik.eth.limo/general/2023/11/27/techno_optimism.html#dacc.

[54] P. Ciaian, d'Artis Kancs, and M. Rajcaniova, "The economic dependency of bitcoin security," *Applied Economics*, 2023, Accessed on 12/13/2023. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/00036846.2021.1931003.

[55] P. Ciaian, d'Artis Kancs, and M. Rajcaniova, *Interdependencies between mining costs, mining rewards and blockchain security*, Accessed on 12/13/2023, 2023. [Online]. Available: https://arxiv.org/abs/2102.08107.

[56] C. Metrics, *Coin metrics miner metrics*, Accessed on 12/13/2023, 2023. [Online]. Available: https://docs.coinmetrics.io/asset-metrics/mining/flowminerout1hopallusd.

[57] R. Recabarren and B. Carbunar, *Hardening stratum, the bitcoin pool mining protocol*, Accessed on 12/13/2023, 2023. [Online]. Available: https://arxiv.org/abs/1703.06545.

[58] J. Messias, V. Pahari, B. Chandrasekaran, K. P. Gummadi, and P. Loiseau, *Dissecting bitcoin and ethereum transactions: On the lack of transaction contention and prioritization transparency in blockchains*, Accessed on 12/13/2023, 2023. [Online]. Available: https://arxiv.org/abs/2302.06962.

[59] P. Merritt and K. Waters, *State of the network's q3 2023 mining data special*, Accessed on 12/13/2023, 2023. [Online]. Available: https://coinmetrics.substack.com/p/state-of-the-network-issue-226.

[60] M. Peterson, *Ethereum transactions frozen for 12 seconds*, Accessed on 12/13/2023, 2023. [Online]. Available: https://blockworks.co/news/ethereum-transactions-frozen.

[61] S. Kessler, *Ethereum's 'censorship' problem is getting worse*, Accessed on 12/13/2023, 2023. [Online]. Available: https://www.coindesk.com/tech/2023/12/06/ethereums-censorship-problem-is-getting-worse/.

[62] R. Explorer, *Rated explorer: Lido dashboard*, Accessed on 12/13/2023, 2023. [Online]. Available: https://www.rated.network/?network=mainnet&view=pool&timeWindow=1d&page=1&poolType=all.

[63] R. Explorer, *Rated explorer: Ethereum validator ratings*, Accessed on 12/13/2023, 2023. [Online]. Available: https://www.rated.network/o/Lido?network=mainnet&timeWindow=30d&viewBy=operator&page=1&idType=pool.

[64] S. Kessler, *Lido tests of 'distributed validator technology' portend 2024 decentralization push*, Accessed on 12/13/2023, 2023. [Online]. Available: https://www.coindesk.com/tech/2023/12/20/lido-tests-of-distributed-validator-technology-portend-2024-decentralization-push/.