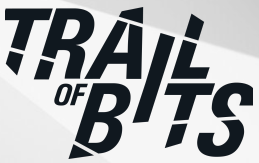


Are Blockchains Decentralized?

Unintended Centralities in Distributed Ledgers

June 2022

Prepared by:

Evan Sultanik
Alexander Remie
Felipe Manzano
Trent Brunson
Sam Moelius
Eric Kilmer
Mike Myers
Talley Amir
Sonya Schriner

About Trail of Bits

Founded in 2012 and headquartered in New York, Trail of Bits provides technical security assessment and advisory services to some of the world's most targeted organizations. We combine high-end security research with a real-world attacker mentality to reduce risk and fortify code. With 80+ employees around the globe, we've helped secure critical software elements that support billions of end users, including Kubernetes and the Linux kernel.

We maintain an exhaustive list of publications at https://github.com/trailofbits/publications, with links to papers, presentations, public audit reports, and podcast appearances.

In recent years, Trail of Bits consultants have showcased cutting-edge research through presentations at CanSecWest, HCSS, Devcon, Empire Hacking, GrrCon, LangSec, NorthSec, the O'Reilly Security Conference, PyCon, REcon, Security BSides, and SummerCon.

We specialize in software testing and code review projects, supporting client organizations in the technology, defense, and finance industries, as well as government entities. Notable clients include HashiCorp, Google, Microsoft, Western Digital, and Zoom.

Trail of Bits also operates a center of excellence for blockchain security. Notable projects include audits of Algorand, Bitcoin SV, Chainlink, Compound, Cosmos, Ethereum 2.0, MakerDAO, Matic, Polkadot, Solana, Uniswap, Web3, and Zcash.

To keep up to date with our latest news and announcements, please follow @trailofbits on Twitter and explore our public repositories at https://github.com/trailofbits. To engage us directly, visit our "Contact" page at https://www.trailofbits.com/contact, or email us at info@trailofbits.com.

Trail of Bits, Inc.
228 Park Ave S #80688
New York, NY 10003
https://www.trailofbits.com
info@trailofbits.com

Table of Contents

Executive Summary

Over the past year, Trail of Bits was engaged by the Defense Advanced Research Projects Agency (DARPA) to investigate the extent to which blockchains are truly decentralized. We focused primarily on the two most popular blockchains: Bitcoin and Ethereum. We also investigated proof-of-stake (PoS) blockchains and Byzantine fault tolerant consensus protocols in general. This report provides a high-level summary of results from the academic literature, as well as our novel research on software centrality and the topology of the Bitcoin consensus network. For an excellent academic survey with a deeper technical discussion, we recommend the work of Sai, et al.[1]

Blockchains Are Decentralized, Right?

Distributed ledger technology (DLT)—and, specifically, *blockchains*—are used in a variety of contexts, such as digital currency, decentralized finance, and even electronic voting. While there are many different types of DLT, each built with fundamentally different design decisions, the overarching value proposition of DLT and blockchains is that they can operate securely without any centralized control. The cryptographic primitives that enable blockchains are, by this point, quite robust, and it is often taken for granted that these primitives enable blockchains to be *immutable* (not susceptible to change). This report gives examples of how that immutability can be broken *not* by exploiting cryptographic vulnerabilities but instead by subverting the properties of a blockchain's implementations, networking, and consensus protocol. We show that a subset of participants can garner excessive, centralized control over the entire system.

Sources of Centralization

This report covers several ways in which control of a DLT can be centralized:

- **Authoritative centrality:** What is the minimum number of entities necessary to disrupt the system? This number is called the *Nakamoto coefficient*, and the closer this value is to one, the more centralized the system. This is also often referred to as "Governance Centrality".

- **Consensus centrality:** Similar to authoritative centrality, to what extent is the source of consensus (e.g., proof-of-work [PoW]) centralized? Does a single entity (like a mining pool) control an undue amount of the network's hashing power?

- **Motivational centrality:** How are participants disincentivized from acting maliciously (e.g., posting malformed or incorrect data)? To what extent are these

[1] Sai et al., "Taxonomy of centralization in public blockchain systems: A systematic literature review," *Information Processing & Management*, Volume 58 Issue 4, (July 2021).

incentives centrally controlled? How, if at all, can the rights of a malicious participant be revoked?

- **Topological centrality:** How resistant is the consensus network to disruption? Is there a subset of nodes that form a vital bridge in the network, without which the network would become bifurcated?

- **Network centrality:** Are the nodes sufficiently geographically dispersed such that they are uniformly distributed across the internet? What would happen if a malicious internet service provider (ISP) or nation-state decided to block or filter all DLT traffic?

- **Software centrality:** To what extent is the safety of the DLT dependent on the security of the software on which it runs? Any bug in the software (either inadvertent or intentional) could invalidate the invariants of the DLT, e.g., breaking immutability. If there is ambiguity in the DLT's specification, two independently developed software clients might disagree, causing a fork in the blockchain. An upstream vulnerability in a dependency shared by the two clients can similarly affect their operation.

Key Findings and Takeaways

The following are the key findings of our research. They are explained in more detail in the remainder of the report.

- The challenge with using a blockchain is that one has to either (a) accept its immutability and trust that its programmers did not introduce a bug, or (b) permit upgradeable contracts or off-chain code that share the same trust issues as a centralized approach.

- Every widely used blockchain has a privileged set of entities that can modify the semantics of the blockchain to potentially change past transactions.

- The number of entities sufficient to disrupt a blockchain is relatively low: four for Bitcoin, two for Ethereum, and less than a dozen for most PoS networks.

- The vast majority of Bitcoin nodes appear to not participate in mining and node operators face no explicit penalty for dishonesty.

- The standard protocol for coordination within blockchain mining pools, Stratum, is unencrypted and, effectively, unauthenticated.

- When nodes have an out-of-date or incorrect view of the network, this lowers the percentage of the hashrate necessary to execute a standard 51% attack. Moreover, only the nodes operated by mining pools need to be degraded to carry out such an

attack. For example, during the first half of 2021 the actual cost of a 51% attack on Bitcoin was closer to 49% of the hashrate.

- For a blockchain to be optimally distributed, there must be a so-called *Sybil cost*. There is currently no known way to implement Sybil costs in a permissionless blockchain like Bitcoin or Ethereum without employing a centralized trusted third party (TTP). Until a mechanism for enforcing Sybil costs without a TTP is discovered, it will be almost impossible for permissionless blockchains to achieve satisfactory decentralization.

- A dense, possibly non-scale-free, subnetwork of Bitcoin nodes appears to be largely responsible for reaching consensus and communicating with miners—the vast majority of nodes do not meaningfully contribute to the health of the network.

- Bitcoin traffic is unencrypted—any third party on the network route between nodes (e.g., ISPs, Wi-Fi access point operators, or governments) can observe and choose to drop any messages they wish.

- Of all Bitcoin traffic, 60% traverses just three ISPs.

- Tor is now the largest network provider in Bitcoin, routing traffic for about half of Bitcoin's nodes. Half of these nodes are routed through the Tor network, and the other half are reachable through .onion addresses. The next largest autonomous system (AS)—or network provider—is AS24940 from Germany, constituting only 10% of nodes. A malicious Tor exit node can modify or drop traffic similarly to an ISP.

- Of Bitcoin's nodes, 21% were running an old version of the Bitcoin Core client that is known to be vulnerable in June of 2021.

- The Ethereum ecosystem has a significant amount of code reuse: 90% of recently deployed Ethereum smart contracts are at least 56% similar to each other.

Contact Information

Administrative points of contact:

Dan Guido, CEO
dan@trailofbits.com

Trent Brunson, PhD, Director of Research
trent.brunson@trailofbits.com

Technical point of contact:

Evan Sultanik, PhD, Principal Investigator
evan.sultanik@trailofbits.com

Scrutinizing Blockchain Immutability

Every blockchain has a privileged set of entities that can modify the semantics of the blockchain to potentially change past transactions: namely, the authors and maintainers of the software. Many blockchains have a virtual machine (VM) built atop—or sometimes even integrated into—their consensus protocol. Bitcoin and its derivatives have a VM for interpreting transaction output scripts. Ethereum uses a VM for executing its smart contracts. Blockchains' VM semantics often evolve in response to both the demand for new features and the need for security mitigations. New VM opcodes are often added, and the costs of performing certain operations are regularly tweaked to prevent denial-of-service attacks.[2]

In some cases, the developers or maintainers of a blockchain intentionally modify its software to mutate the blockchain's state to revert or mitigate an attack—this was Ethereum's response to the 2016 DAO hack.[3] But in most other cases, changes to a blockchain are an unintentional or unexpected consequence of another change. For example, Ethereum's Constantinople hard fork reduced the gas costs of certain operations. However, some immutable contracts that were deployed before the hard fork relied on the old costs to prevent a certain class of attack called "reentrancy." Constantinople's semantic changes caused these once secure contracts to become vulnerable.[4] Fortunately, this issue was discovered manually, by chance, with just enough time before the fork for it to be delayed and later abandoned. In 2021, the Polkadot blockchain platform was temporarily crippled by node failures caused by an update to the Rust programming language compiler used to build the nodes.[5] In late August of 2021, a consensus issue related to changes in the most popular Ethereum client was exploited to cause a hard fork of the cryptocurrency.[6]

The data—and, more importantly, the code—deployed to a blockchain are not necessarily semantically immutable. Not only can the state of the blockchain be retroactively changed through modifications to the blockchain's software, but the semantics of individual transactions can change between when the transaction is initiated and when it is ultimately mined onto the blockchain thanks to software changes in the interim. Some blockchain platforms like Polkadot and Substrate also allow certain parameters and code to be updated through an on-chain governance process.

---

[2] Renlord Yang et al., "Empirically Analyzing Ethereum's Gas Mechanism," *IEEE EuroS&P*, 2019.
[3] David Siegel, "Understanding the DAO Attack," *CoinDesk*, June 25, 2016.
[4] Christine Kim and Nikhilesh De, "Ethereum's Constantinople Upgrade Faces Delay Due to Security Vulnerability," *CoinDesk*, January 15, 2019.
[5] Bastian Köcher, "A Polkadot Postmortem," *Polkadot* (blog), May 24, 2021.
[6] Turner Wright, "Bug in Ethereum Client Leads to Split — EVM-Compatible Chains at Risk," *Cointelegraph*, August 27, 2021.

The software itself does not necessarily need to change to affect the security properties of a DLT. For example, although Bitcoin is less than 15 years old, many of the foundational assumptions made when its protocol was designed have already become obsolete. When Bitcoin was originally conceived, Nakamoto assumed that each node in the consensus network would participate in mining. However, as the mining difficulty increases—thus decreasing the probability of getting a mining reward—"mining pools" (collectives that group both mining power and rewards) become increasingly popular as a means to garner a consistent profit. Today, the four most popular mining pools constitute over 51% of the hashrate of Bitcoin. Each mining pool operates its own, proprietary, centralized protocol and interacts with the public Bitcoin network only through a gateway node. In other words, there are really only a handful of nodes that participate in the consensus network on behalf of the majority of the network's hashrate. Controlling those nodes provides the means to, at a minimum, deny service to their constituent hashrate. This breaks the original assumption that all Bitcoin nodes will have a financial incentive (via mining) to remain honest. **If a node operator's self-interest is to be dishonest, then there is no explicit penalty for doing so**. Moreover, the number of *entities* necessary to execute a 51% attack on Bitcoin was reduced from 51% of the entire network (which we estimate at approximately 59,000 nodes) to only the four most popular mining pool nodes[7] (less than 0.004% of the network).

Finally, any blockchain that supports Turing-complete[8] on-chain execution (e.g., Ethereum, Hyperledger, and Tezos) cannot enforce semantic immutability. This is because such blockchains cannot prevent contracts from being upgradeable (a Turing Machine is capable of simulating any other Turing Machine,[9] allowing for upgradeability via interpreted inputs even if the on-chain code is immutable). For example, Alice can submit a transaction to a contract and, before the transaction is mined, the contract could be upgraded to have completely different semantics. The transaction would be executed against the new contract. Upgradeable contract patterns have become incredibly popular in Ethereum as they allow developers to circumvent immutability to patch bugs after deployment. But they also allow developers to patch in backdoors that would allow them to abscond with a contract's assets. **The challenge with using a blockchain is that one has to either (a) accept its immutability and trust that the programmers did not introduce a bug, or (b) permit upgradeable contracts or off-chain code that share the same trust issues as a centralized approach.**

[7] https://www.blockchain.com/charts/pools
[8] Such blockchains are technically linear bounded automata due to gas constraints.
[9] *Stanford Encyclopedia of Philosophy,* s.v. "Turing Machines," first published September 24, 2018.

The Nakamoto Coefficient

Various metrics have been proposed to measure the centrality or fairness of a DLT, including the Gini coefficient and Lorenz curve, both borrowed from economic theory. However, the minimum *Nakamoto coefficient* is perhaps the most intuitive. The Nakamoto coefficient is the number of entities sufficient to attack the system.[10] A completely centralized system will have a Nakamoto coefficient of one. The lower the Nakamoto coefficient, the more centralized the system.

It is well known that Bitcoin is *economically* centralized: in 2020, 4.5% of Bitcoin holders controlled 85% of the currency.[11] But what about Bitcoin's *systemic* or *authoritative* centralization? As we saw in the last section, **Bitcoin's Nakamoto coefficient is four**, because taking control of the four largest mining pools would provide a hashrate sufficient to execute a 51% attack. In January of 2021, **the Nakamoto coefficient for Ethereum was only two.**[12] As of April 2022, it is three.[13]

Even though these Nakamoto coefficients are relatively low, some might argue that exploiting them to attack a blockchain would be prohibitively expensive. While this may be true for individuals, **the actors incentivized to perpetrate these attacks include operators of competing currencies and nation-states who have the requisite resources**. Perverse incentives can exist with blockchains in the same way that the relative values of fiat currencies can be manipulated.

PoS protocols are becoming increasingly popular consensus mechanisms that address some of the shortcomings (e.g., expensive computation) of PoW blockchains like Bitcoin, Ethereum, and their derivatives. Instead of solving computationally hard problems like PoW miners do to mine blocks, most PoS networks instead require its block validators to stake a certain amount of cryptocurrency as collateral in the event that they act dishonestly—their mining power is proportional to their stake. Some PoS chains like Algorand distribute cryptocurrency as rewards for good governance.[14] PoS blockchains employ complex protocols to ensure that transactions are validated and to police the validators. Most PoS blockchain's consensus protocols (Avalanche's Snowflake, Solana's Tower BFT, etc.) break down if the validators associated with at least one-third of the staked assets are malicious, effectively pausing the network. Therefore, the Nakamoto coefficient of most PoS

[10] Balaji S. Srinivasan, "Quantifying Decentralization," *news.earn.com*, July 27, 2017.
[11] Sami Ben Mariem et al., "All that Glitters Is Not Bitcoin — Unveiling the Centralized Nature of the BTC (IP) Network," *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, (February 19, 2020).
[12] Qinwei Lin et al., "Measuring Decentralization in Bitcoin and Ethereum Using Multiple Metrics and Granularities," *arXiv:2101.10699v2 [cs.CR]*, (February 2, 2021).
[13] https://miningpoolstats.stream/ethereum
[14] Algorand Governance, s.v. "More Committing Commitments," accessed April 27, 2022.

blockchains is equal to the smallest number of validators that have collectively staked at least a third of all of the staked assets.

The following are the Nakamoto coefficients for popular PoS blockchains as of August 25, 2021:

Blockchain	Nakamoto Coefficient	Total # of Validators	Staked Value	Source
Avalanche	25	1,041	$11B	https://explorer.avax.network/validators
Solana	19	876	$37B	https://solana.com/validators
Eth2[15]	12	219,182	$22B	https://www.nansen.ai/
THORChain	11	38	$0.5B	https://thorchain.net/#/nodes
Terra	8	130	$12B	https://stake.id/#/
Cosmos	6	125	$4B	https://www.mintscan.io/cosmos/validators
BSC[16]	5	21	$7B	https://bscscan.com/validatorset
Fantom	3	46	$1B	https://ftmscan.com/validators
Polygon	2	100	$3B	https://wallet.matic.network/staking/

[15] The total number of validators is an upper bound. According to Nansen, the four biggest depositors have more than a third of the stake, and those depositors have 12 nodes.
[16] The number of validators necessary to reach one third of the stake is seven, but three are controlled by the same entity: Binance.

Consensus Centrality: Mining Pool Vulnerabilities

An increasing number of consensus protocol operations are being delegated to a small number of entities that typically run their own centralized software and protocols with little-to-no on-chain governance—in the case of PoW blockchains, these entities are the mining pools, and in the case of PoS blockchains, these entities are staked validators. In the previous section, we discussed how these entities present a significant target to disrupt the stability of a blockchain. In this section, we discuss how such entities' off-chain governance structures further increase the attack surface of a blockchain.

While there is evidence that risk-sharing entities such as mining pools and staked validators decrease the *economic* centralization of a blockchain, it is well known that they exist as *technological* single points of failure and are therefore rich targets for denial-of-service attacks.[17] **The safety of a blockchain depends on the security of the software and protocols of its off-chain governance or consensus mechanisms.**

Today, mining pool operators communicate with their participants using Stratum: an ad hoc JSON remote procedure call (RPC) protocol that organically evolved over the past decade with no official standardization. The protocol permits the mining pool operator to create "jobs" for each mining participant, each of which requires the participant to brute-force search through a unique subset of the search space of possible valid blocks.

The Stratum protocol is not encrypted. All jobs assigned to miners, all work results from miners, and even the initial authentication are transmitted in plaintext. The Stratum developers may have made this design decision because the Stratum protocol is implemented in the firmware of most hardware miners, which may not have the resources to implement SSL or TLS. Moreover, the Stratum developers may not have anticipated that attackers could exploit this design to authenticate as another user. It was later discovered that an eavesdropper such as a nation-state, ISP, or local network participant can use this transmitted information to estimate the hashrate and payouts of a miner in the pool. A malicious attacker-in-the-middle can actually manipulate Stratum messages to steal CPU cycles and payouts from mining pool participants.[18] These vulnerabilities have been known for years, and were initially addressed by adding forms of authentication to the Stratum protocol. However, none of the proposals to transition to a more secure protocol have been widely adopted.

Until 2018, authentication in the Stratum protocol did not even require a password. Attackers realized that they could deny service to mining participants by authenticating

[17] Lin William Cong, Zhiguo He, and Jiasun Li, "Decentralized Mining in Centralized Pools," *SSRN Electronic Journal* (January 2018).
[18] Ruben Recabarren and Bogdan Carbunar, "Hardening Stratum, the Bitcoin Pool Mining Protocol," *PETS* 3 (March 2017): 1–18.

with their usernames (which were enumerable from the mining pool website) and submitting invalid work.[19] After a miner submits a sufficient number of invalid blocks, mining pools would block the account of the participant, ignoring all further work and preventing future payouts. This was patched by requiring a password with authentication and using IP-based rather than account-based ban lists.

We have discovered that, today, all of the mining pools we tested either assign a hard-coded password for all accounts or simply do not validate the password provided during authentication. For example, all ViaBTC accounts appear to be assigned the password "123." Poolin seems not to validate authentication credentials at all. Slushpool explicitly instructs its users to ignore the password field as, "It is a legacy Stratum protocol parameter that has no use nowadays."[20] We discovered this by registering multiple accounts with the mining pools, and examining their server code, when available. These three mining pools alone account for roughly 25% of the Bitcoin hashrate.

The job of each miner is to find a nonce value that, when appended to the block header chosen by the mining pool, hashes to a value below a certain threshold set by the blockchain's current difficulty. A certain portion of the header is specific to the job/miner in order to prevent duplicate work across the jobs. The strategy by which mining pools choose both the base header for each job and the division of the search space between jobs (and, therefore, between individual miners) is not a part of the Stratum protocol; it is proprietary to the mining pool. ViaBTC is open source, so we can inspect how it works. ViaBTC creates a custom "coinbase" for each miner: the address to which rewards are deposited on success. This is what prevents a miner from absconding with a successfully mined block—the reward address, controlled by ViaBTC, is already baked into the header. ViaBTC also maintains a global, 32-bit job counter that it adds to the header, minimizing the search space overlap between jobs. The size of the search space for each job is 2^{96} bits out of 2^{256} bits, and it is unlikely that an attacker could overflow the job counter through repeated Stratum job requests, so it is still unlikely that jobs will have much overlap. However, **the mining pool server will continue to accept and perform computations to validate bogus work submitted by improperly authenticated miners, potentially leading to a denial of service.**

[19] Mohiuddin Ahmed et al., "A Poisoning Attack against Cryptocurrency Mining Pools," *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, eds. Joaquin Garcia-Alfaro et al. (Cham: Spring International Publishing, 2017), 140–154.
[20] Slushpool Bitcoin mining setup guide, s.v. "Which worker name/password should I choose?", accessed April 27, 2022.

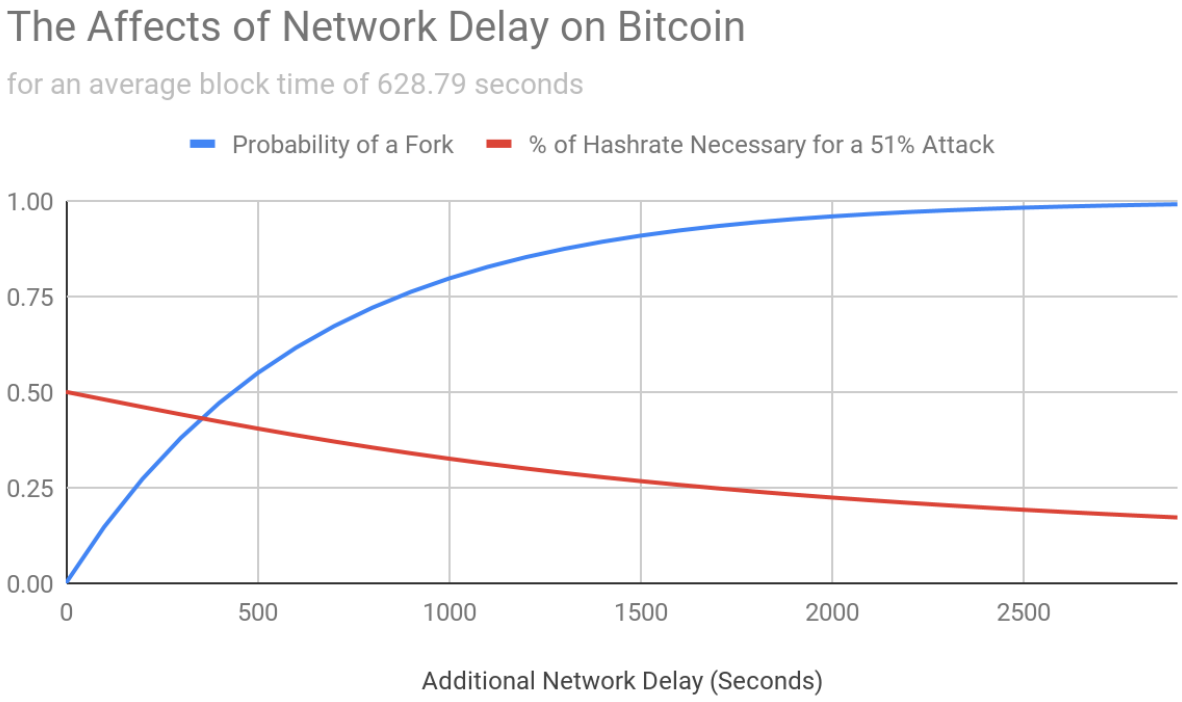
Sybil and Eclipse Attacks: The "Other" 51%

The discourse on attacks against PoW blockchains typically centers around the 51% attack: the very real threat that if a single entity controls at least 51% of the hashrate of the network, then that entity can modify the blockchain in otherwise prohibited ways.

It turns out that there are other forms of the 51% attack that affect all types of blockchains and distributed systems in general. What if the blockchain's consensus network were flooded with new, malicious nodes controlled by a single party? After all, deploying a new node requires only one inexpensive cloud server instance—no specialized mining hardware is necessary. This is called a *Sybil attack*. Such attacks can be used to affect the topology of the network in order to gain influence.

Sybil attacks can also be used to execute an *eclipse attack*: the denial of service to specific nodes in order to gain influence.[21] If one can cause nodes to have a sufficiently out-of-date or incorrect view of the network, this increases the probability of a blockchain fork: when two miners produce and broadcast valid but distinct blocks with the same parent block.[22] The longer the fork's branches become, the lower the percentage of the hashrate necessary for an attacker to execute a standard 51% attack.[23] This is because, eventually, one of the two branches will become the canonical head of the blockchain and the other branch will become a so-called "ommer" (previously called "uncle") blocks. Any transactions mined in ommer blocks will be invalidated, as if they had never been mined. The reason why forks reduce the cost of a standard 51% attack is because any hashrate expended toward extending a branch of the fork that will eventually become ommers is effectively wasted, reducing the effective global computational efficiency of the blockchain. Moreover, only the nodes directly connected to miners need to be degraded to carry out such an attack.[24]

[21] Atul Singh et al., "Defending against Eclipse Attacks on Overlay Networks," *EW 11: Proceedings of the 11th workshop on ACM SIGOPS European workshop* (September 2004).
[22] Christian Decker and Roger Wattenhofer, "Information Propagation in the Bitcoin Network," *IEEE P2P 2013 Proceedings* (2013).
[23] Dembo et al., "Everything is a Race and Nakamoto Always Wins," *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (November 2020)
[24] Ittay Eyal and Emin Gün Sirer, "Majority Is Not Enough: Bitcoin Mining Is Vulnerable," 2018.



The probability of a fork is calculated from Equation (3) of (Decker & Wattenhofer, 2013):

$$Pr[F \geq 1] = 1 - (1 - \lambda)^{\Delta},$$

where λ is the total mining rate (i.e., the inverse average block time) and Δ is the average network delay. The percentage of hashrate necessary to execute a standard 51% attack (also known as the "attack threshold") is a consequence of Equation (2) of (Dembo et al., 2020):

$$\beta < \frac{1-\beta}{1+(1-\beta)\lambda\Delta}$$

$$\Downarrow$$

$$\beta < \frac{\lambda\Delta+2}{2\lambda\Delta} - \frac{1}{2}\sqrt{\frac{\lambda^2\Delta^2+4}{\lambda^2\Delta^2}}, \text{ assuming } \lambda\Delta > 0.$$

From our calculations based on data collected between January and June 2021, the effective computational power of the Bitcoin network was only 98.68% of its theoretical maximum power, due to the natural latency of the network. In other words, miners were operating on out-of-date information 1.32% of the time, thereby wasting their time. This means that **the actual cost of a 51% attack on Bitcoin was closer to 49% of the hashrate.** Therefore, contrary to established lore, it does not actually take 51% of the network's hashing power to mount a successful 51% attack, even when all actors are

assumed honest. With the accidental or nefarious introduction of further latency, the hashrate needed can plummet. With just a few minutes of delay, the takeover threshold drops to 40%, and with less than an hour it can be as low as 20%. All this should be taken in the context that just four mining pools already control more than 51% of the hashing power.

In July 2021, Grundmann and Baumstark were able to observe a Sybil attack on the public Bitcoin nodes.[25] The authors neither concluded nor speculated on the purpose of the attack; however, the attack did have the effect of significantly reducing the connectivity of the public Bitcoin network. Our analysis shows that this Sybil attack could have enabled an eclipse attack.

A recent impossibility result for the decentralization of permissionless blockchains like Bitcoin and Ethereum was discovered by Kwon et al.[26] It indicates that for a blockchain to be optimally distributed, there must be a so-called *Sybil cost*. That is, the cost of a single participant operating multiple nodes must be greater than the cost of operating one node. Unfortunately, Kwon et al. conclude that **there is currently no known way to implement Sybil costs in a permissionless blockchain like Bitcoin or Ethereum *without* employing a centralized trusted third party (TTP)**. Until a mechanism for enforcing Sybil costs without a TTP is discovered, it will be almost impossible for permissionless blockchains to achieve satisfactory decentralization.

[25] Matthias Grundmann, Max Baumstark, and Hannes Hartenstein, "Estimating the Node Degree of Public Peers and Detecting Sybil Peers Based on Address Messages in the Bitcoin P2P Network," 2021.

[26] Yujin Kwon et al., "Impossibility of Full Decentralization in Permissionless Blockchains," *Proceedings of the 1st ACM Conference on Advances in Financial Technologies* (October 2019).

Distributed Organization and the Power Law

Casual observers often assume that DLTs' peer-to-peer networks are "scale-free".[27] Roughly, a network is scale-free if the fraction of nodes with degree k is k^{-c}, for some constant c. This is a reasonable assumption, since many other natural phenomena such as social networks self-organize in this way. Scale-free properties in peer-to-peer networks are desirable since they provide a good balance between minimizing propagation delays and network connections, allowing the network to reach consensus faster with fewer interconnections.[28] After all, the purpose of the network is to reach consensus on the current state of the blockchain and to disseminate new, unmined transactions to other nodes. The faster this information spreads through the network, the harder it is to exploit information delay by executing an eclipse attack as described in the last section.

Are popular blockchain networks *actually* scale-free? It turns out that there is very little empirical evidence for this. While some blockchains like Ethereum use peer discovery protocols that have theoretical guarantees on consistency,[29] Bitcoin and its derivatives use a custom protocol about which relatively little has been written. The Bitcoin protocol does not provide a means for directly observing the peers of a node, although a node's peers can be indirectly estimated under certain rare conditions.[30]

Bitcoin's network topology is dictated by its peer discovery and connection algorithm, which is a part of the client's implementation and not the protocol itself. Bitcoin Core—by far the most popular Bitcoin client implementation—has hard-coded constants for various parameters that affect peering and, therefore, the network topology. These constants are not officially documented anywhere else, yet drastically affect the topology of the consensus network. The only way to examine those constants (or even know they exist, for that matter) is to interrogate the source code. Therefore, **the only comprehensive reference for the behavior of Bitcoin nodes is the source code of its most popular client.**

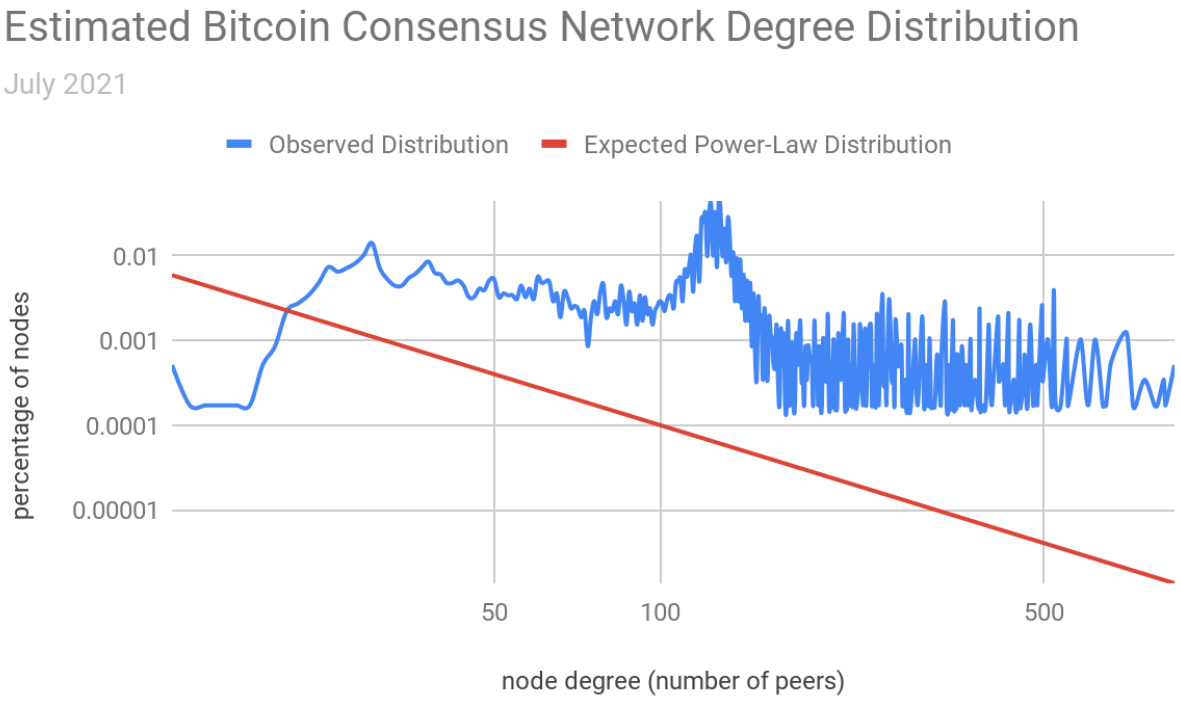
The cap on the number of known peer addresses that are shared with other peers is hard-coded to 23% or 1,000, whichever is smaller. Bitcoin Core does not enable *network address translation* (NAT) traversal or *Universal Plug and Play* (UPnP) by default, so if a

[27] Victoriano Izquierdo, "Centralized or Decentralized? Free Scale Networks!," *Medium*, August 19, 2017.
[28] Cohen and Havlin, "Scale-Free Networks Are Ultrasmall," *Physical Review Letters*, Volume 90 Issue 5, (February 7, 2003).
[29] Petar Maymounkov and David Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," 2002.
[30] Matthias Grundmann, Max Baumstark, and Hannes Hartenstein, "Estimating the Node Degree of Public Peers and Detecting Sybil Peers Based on Address Messages in the Bitcoin P2P Network," 2021.

Bitcoin node is run without a public IP address (e.g., on a home network or behind a firewall), it will not be able to receive incoming connections from other peers. These "non-public" Bitcoin nodes are able to make only outgoing connections, which are capped at eight. The "public" Bitcoin nodes that *do* accept incoming connections cap their peer count at 125. The Bitcoin client implementation also attempts to maximize the diversity of its peers by limiting the similarity of its peers' IP addresses.[31] Therefore, while the public nodes do interconnect with each other using a modified form of preferential attachment[32]—and therefore should have scale-free properties—the non-public nodes act as approximately regular-degree spokes around the hub of public nodes.

We know that the diameter of almost every random scale-free graph is very small:[33] $\log n \div \log \log n$, which for Bitcoin would place its diameter at five. The Bitcoin Core client has a hard-coded delay of two minutes before it gossips new verified blocks to a peer. Therefore, if Bitcoin were scale-free, we would expect an average block propagation delay of 10 minutes. However, we regularly observe block propagation delays of less than 10 minutes, suggesting that the graph is not in fact scale-free. Our crawls of the Bitcoin network suggest that the diameter is closer to four. This evidence supports our supposition that **a dense (possibly non-scale-free) subnetwork of public nodes is largely responsible for reaching consensus and communicating with miners.** This hypothesis is supported by empirical estimates of the degree distribution.

[31] Bitcoin Core will initiate at most only one peer connection to an IP address in each 16-bit CIDR block.
[32] Albert-Laszlo Barabási and Reka Albert, "Emergence of Scaling in Random Networks," *Science* 286, no. 509 (1999).
[33] Béla Bollobás and Oliver Riordan, "The Diameter of a Scale-Free Random Graph," *Combinatorica* 24, no. 1 (2004): 5–34.



There is a peak in the degree distribution at 125 peers, since this is the default cap for the Bitcoin Core client. Nodes with more peers are either running a different or modified Bitcoin client.

By crawling the Bitcoin network and querying nodes for known peers, we can estimate the number of public Bitcoin nodes (i.e., nodes actively accepting incoming connections). From crawling the Bitcoin network throughout 2021, we estimate that the public Bitcoin nodes constitute only 6–11% of the total number of nodes. Therefore, **the vast majority of Bitcoin nodes do not meaningfully contribute to the health of the Bitcoin network**. We have extended the Barabási–Albert random graph model to capture the behavior of Bitcoin peering. This model suggests that at the current size of the Bitcoin network, at least 10% of nodes must be public to ensure that new nodes are able to maximize their number of peers (and, therefore, maximize the health and connectivity of the network). As the total number of nodes increases, this bound approaches 40%.

Network Centrality

In the previous section, we investigated how a DLT's network of nodes can affect centralization. But what about the actual underlying network infrastructure? For at least the past five years, **60% of all Bitcoin traffic has traversed just three ISPs.**[34] As of July 2021, about half of all public Bitcoin nodes were operating from IP addresses in German, French, and US ASes, the top four of which are hosting providers (Hetzner, OVH, Digital Ocean, and Amazon AWS). The country hosting the most nodes is the United States (roughly one-third), followed by Germany (one-quarter), France (10%), The Netherlands (5%), and China (3%). Moreover, at the same time, approximately half of all Bitcoin traffic was routed through Tor.[35] This is yet another potential surface on which to execute an eclipse attack, since the ISPs and hosting providers have the ability to arbitrarily degrade or deny service to any node. Traditional Border Gateway Protocol (BGP) routing attacks have also been identified as threats.[36]

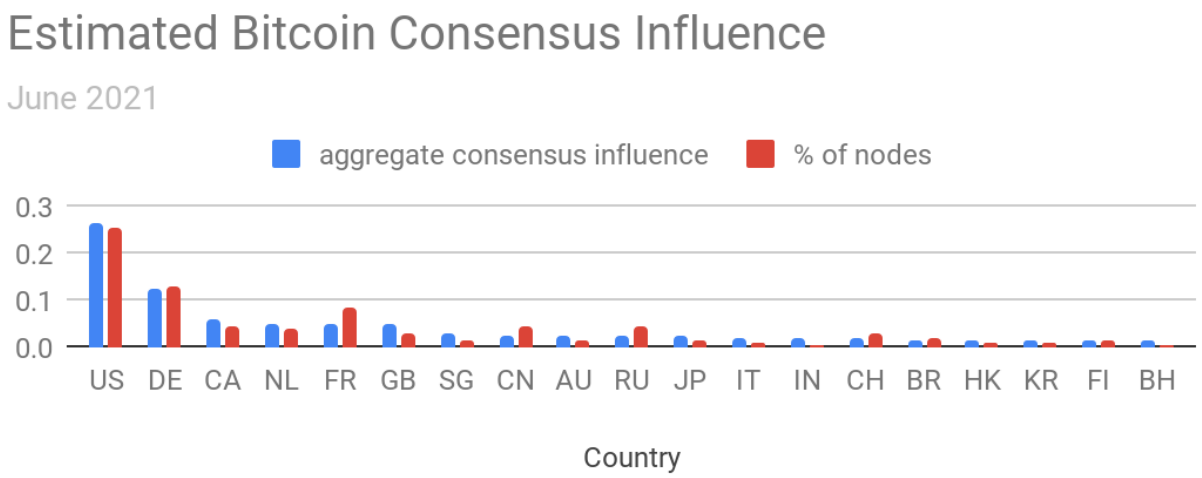
The underlying network infrastructure is particularly important for Bitcoin and its derivatives, since all Bitcoin protocol traffic is unencrypted. Unencrypted traffic is fine for transactional and block data, since they are cryptographically signed and, therefore, impervious to tampering. However, **any third party on the network route between nodes (e.g., ISPs, Wi-Fi access point operators, or governments) can observe and choose to drop any messages they wish**. Say Alice wants to transfer ₿1 to Bob. She creates a transaction for the transfer, digitally signs it, and submits it to a node for propagation throughout the network. The transaction is not yet confirmed; it is in a limbo called the *mempool.* Alice's node will gossip the transaction to its peers until the message eventually reaches a node associated with a miner (or, more likely, a mining pool). The miner can then choose to include the transaction in a block. Once a block with Alice's transaction is mined, it is passed back a node to be gossiped back through the rest of the network. At any point in this process, a malicious node, miner, or *intermediary on the network* can choose to forgo gossiping the transaction before it is mined. If a mining pool's nodes are not sufficiently connected to the dense subnetwork of public nodes described in the previous section, then this sort of attack is easier.

The Bitcoin protocol also allows nodes to be run as Tor hidden services. In fact, **Tor is now more popular than any other AS—or network provider—in Bitcoin, routing traffic for about 20% of Bitcoin nodes.** The next largest AS is AS24940 from Germany, constituting

[34] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever, "Hijacking Bitcoin: Routing Attacks on Cryptocurrencies," *IEEE Symposium on Security and Privacy* (2017).
[35] Osato Avan-Nomayo, "Bitcoin network node count sets new all-time high," *Cointelegraph*, July 15, 2021.
[36] Muoi Tran et al., "A Stealthier Partitioning Attack against Bitcoin Peer-to-Peer Network," *IEEE Symposium on Security and Privacy* (2020).

only 10% of nodes. This is concerning[37] because a malicious Tor exit node can modify or drop traffic similar to an ISP, as described above. Over the past year, a malicious actor (widely believed to be from Russia) used a Sybil attack to gain control of up to 40% of Tor exit nodes. The attacker used the nodes to rewrite Bitcoin traffic.[38]

We propose a new metric that captures the amount of influence a node has on the consensus of the entire network based on its topological position: *consensus influence*, equal to the node's *eigencentrality*.[39] A node's consensus influence is a function of the consensus influence of its peers; nodes with more influential peers are themselves more influential. The higher this value, the more influence a node has on consensus. Another property of this definition is that the higher a node's consensus influence, the more gossip protocol messages that will pass through it. This metric can be calculated using the principal eigenvector of the network's adjacency matrix. As expected, **the two countries with the highest percentage of non-Tor nodes, the United States and Germany, have the highest aggregate consensus influence in Bitcoin.**

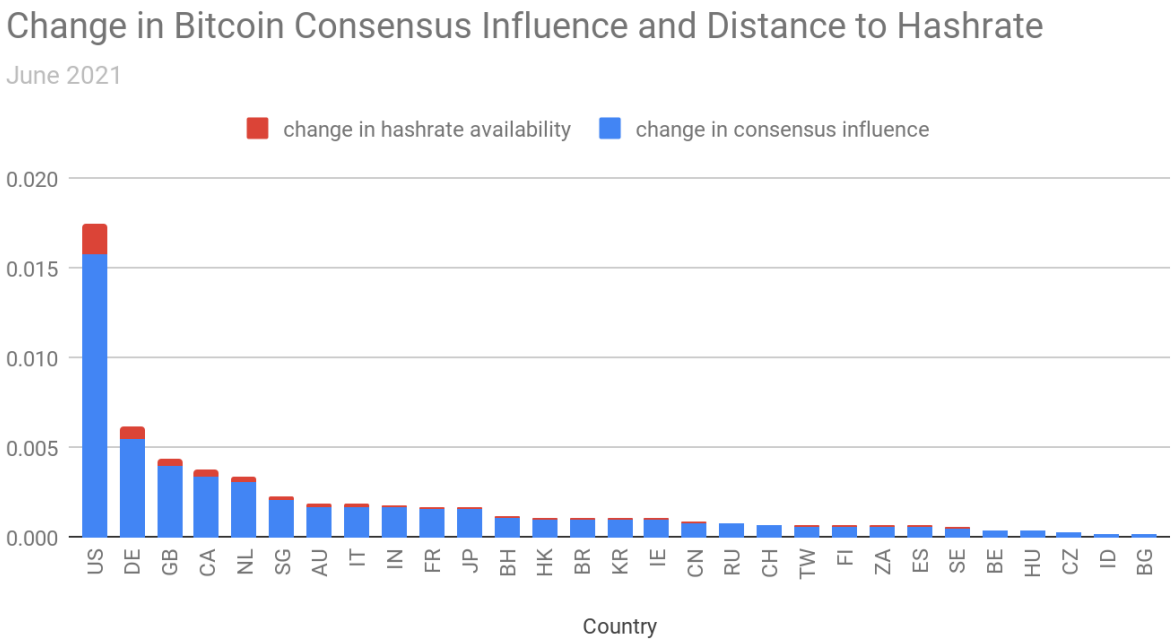


Consensus influence must be estimated for Bitcoin using a combination of crawl data and a probabilistic model of the topology since Bitcoin clients do not explicitly reveal their peers.

We would like to quantify the extent to which a country that unilaterally blocked all Bitcoin traffic could affect the system. We can calculate this effect on node consensus versus the effect on "hashrate availability", which we define as the estimated network delay between a node in the consensus network and all of the miners in the network, normalized by their hashrate. The lower the hashrate availability of a node, the quicker its messages will be

[37] Alex Biryukov and Ivan Pustogarov, "Bitcoin over Tor Isn't a Good Idea," *IEEE Symposium on Security and Privacy* (2015).
[38] Nusenu, "Tracking One Year of Malicious Tor Exit Relay Activities (Part II)," *Medium*, May 8, 2021.
[39] Mohammed J. Zaki and Wagner Meira, Jr. (2014). *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press. ISBN 9780521766333.

transmitted to and from the miners. We first estimate the global distribution aggregate consensus influence, as in the previous chart. Next, for each country, we remove that country and calculate the new distribution of consensus influence among the remaining countries. We quantify the change by comparing the distributions' relative entropy (Kullback–Leibler divergence). This is depicted as the blue bars in the following chart. We repeat this process calculating instead the change in hashrate availability, depicted as the red bars in the chart. Larger blue bars indicate countries whose removal would have the most significant effect on the resulting consensus network topology. Larger red bars indicate countries whose removal would have the most significant effect on the other countries' communications access to hashrate.



Software Centrality

As discussed earlier, it is vital that all DLT nodes operate on the same latest version of software, otherwise, consensus errors can occur and lead to a blockchain fork. Software differentials and vulnerabilities regularly cause consensus errors. For example, on August 24, 2021, a bug in an older version of the popular Ethereum client Geth was hastily patched.[40] However, participants in the Flexpool, BTC.com, and Binance mining pools continued to use older, unpatched versions of the software. On August 27, 2021, the inconsistent patching led to a consensus error that forked the Ethereum blockchain.[41] On October 25, 2021, a vulnerability in all prior versions of Geth was discovered that permitted a carefully crafted peer-to-peer message to inflict a denial-of-service attack on the receiving node.[42] From our crawls of the Bitcoin network, we observe that **21% of Bitcoin nodes are running an old version of the Bitcoin Core client that is known to be vulnerable.**

While software bugs can lead to consensus errors, we demonstrated that overt software changes can also modify the state of the blockchain. Therefore, the core developers and maintainers of blockchain software are a centralized point of trust in the system, susceptible to targeted attack. There are currently four active contributors with access to modify the Bitcoin Core codebase,[43] the compromise of any of whom would allow for arbitrary modification of the codebase. Recently, the lead developer of the $8 billion Polygon network, Jordi Baylina, was recently targeted in an attack with the Pegasus malware,[44] which could have been used to steal his wallet or deployment credentials.

The blockchain client implementation is not alone in its importance—the entire ecosystem of blockchain software poses a risk of consensus errors and differentials. For example, cryptocurrency traders must decide whether to use a non-custodial wallet (i.e., to manage and store their own credentials in a local digital wallet) versus escrowing their credentials in a centralized custodial exchange. The majority of users appear to do the latter. This choice is not simply about the convenience of delegating management to a third party; it is about whether one trusts a centralized third party versus one's own security hygiene and the developers of one's non-custodial wallet.

---

[40] Christine Kim, "Ethereum's Most Popular Software Client Issues Hotfix to High Severity Bug," *CoinDesk*, August 24, 2021.
[41] Joanna Ossinger, "Ethereum Weathers Bug that Underlines Possible Blockchain Risks," *Bloomberg*, August 30, 2021.
[42] Martin Holst Swende, "CVE-2021-41173: DoS via maliciously crafted P2P message," `ethereum/go-ethereum`, *GitHub*, October 25, 2021.
[43] Brandy Betz, "2 Prominent Bitcoin Core Contributors Step Away From Their Roles," *CoinDesk,* December 10, 2021.
[44] John Scott-Railton et al., "Extensive Mercenary Spyware Operation against Catalans Using Pegasus and Candiru," *The Citizen Lab*, April 18, 2022.

We generated software bills of materials (SBOMs) and dependency graphs for the major clients for Bitcoin, Bitcoin Cash, Bitcoin Gold, Ethereum, Zcash, Iota, Dash, Dogecoin, Monero, and Litecoin. We then compared two dependency graphs based on the clients' normalized edit distance.

	Bitcoin	Dash	Bitcoin Cash	Dogecoin	BTCGPU	Litecoin	Monero	Zcash	IOTA	Geth
Bitcoin	1.00	1.00	0.99	0.99	0.99	0.99	0.95	0.92	0.90	0.90
Dash	1.00	1.00	0.99	0.99	0.99	0.99	0.95	0.92	0.90	0.90
Bitcoin Cash	0.99	0.99	1.00	0.99	0.99	0.99	0.95	0.92	0.90	0.88
Dogecoin	0.99	0.99	0.99	1.00	1.00	1.00	0.95	0.93	0.90	0.89
BTCGPU	0.99	0.99	0.99	1.00	1.00	1.00	0.95	0.93	0.90	0.89
Litecoin	0.99	0.99	0.99	1.00	1.00	1.00	0.95	0.93	0.90	0.89
Monero	0.95	0.95	0.95	0.95	0.95	0.95	1.00	0.92	0.91	0.92
Zcash	0.92	0.92	0.92	0.93	0.93	0.93	0.92	1.00	0.94	0.88
IOTA	0.90	0.90	0.90	0.90	0.90	0.90	0.91	0.94	1.00	0.91
Geth	0.90	0.90	0.88	0.89	0.89	0.89	0.92	0.88	0.91	1.00

Our edit distance metric is calculated by comparing the relative depths of all shared dependencies in their dependency graphs. If the depth of a shared dependency is different between two dependency trees, then we say that they have an edit distance of the inverse of the minimum depth minus the inverse of the maximum depth. For all nodes that are in one dependency graph but not the other, the edit distance is the inverse of the depth of the node. We then normalize the total edit distance by the sum of the inverse depths of all dependencies in each graph. A value of 0.0 means that the graphs are completely different and a value of 1.0 means that the graphs are identical.

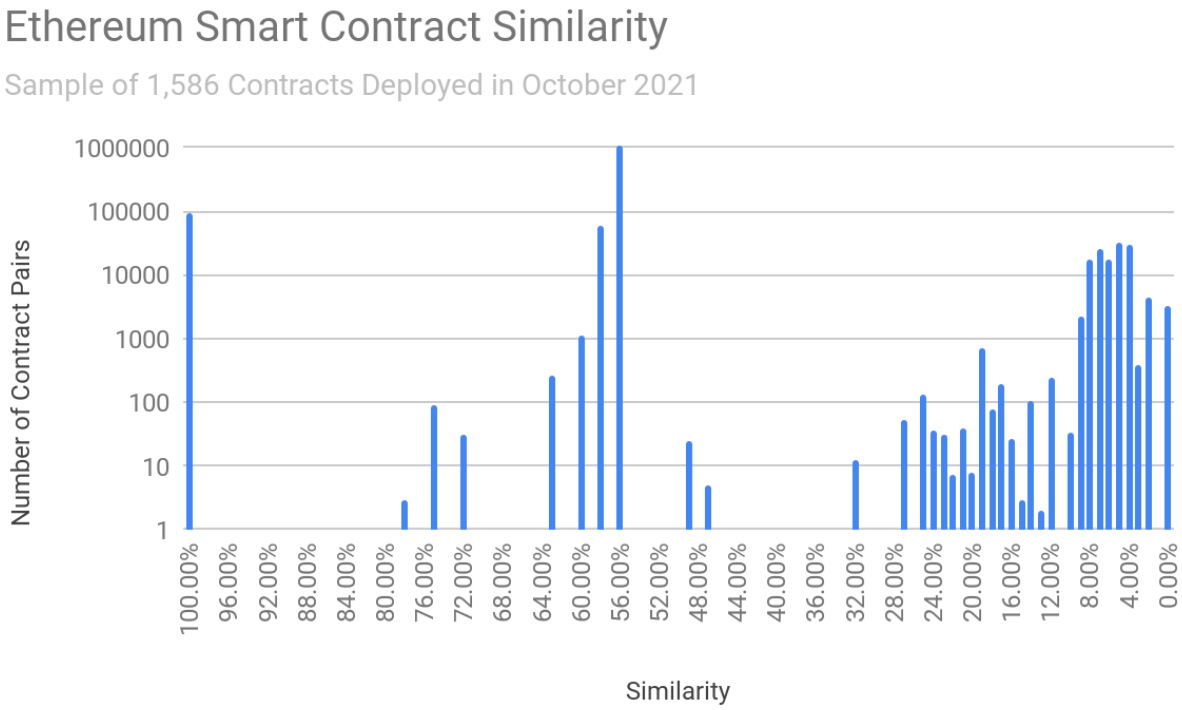
As expected, Bitcoin forks and derivatives remain nearly identical to Bitcoin. Surprisingly, Monero, Zcash, and Geth—which were all independently developed—are also very similar to Bitcoin.

As mining pools are increasingly necessary for PoW mining to be profitable, the centralization and security of their associated infrastructure are increasingly important. The most popular Bitcoin mining pool, AntPool, distributes client software to its miners in the form of black-box, closed-source Windows binaries. **To the best of our knowledge, there has never been a third-party security assessment of these tools.** ViaBTC, one of the top four Bitcoin mining pools, has open-sourced its client code. The system is complex, is written in C, and includes many historically difficult-to-implement components in a language like C. For example, it includes handwritten parsers that process external web requests. **Any remote code execution vulnerability in a mining pool client would allow**

an attacker to either deny service to the mining pool (i.e., reducing the overall hashrate) or redirect the hashrate toward a 51% attack.

On-chain software is also susceptible to code reuse and vulnerabilities. For example, the Ethereum smart contract ecosystem makes heavy use of code reuse and sharing to implement common features that are not natively available in the common language frameworks. Most contracts use the OpenZeppelin library for things like mathematical operations with overflow/underflow detection and standard token API implementations.

We sampled 1,586 smart contracts deployed to the Ethereum blockchain in October 2021, and compared their bytecode similarity, using Levenshtein distance as a metric. One would expect such a metric to *underestimate* the similarity between contracts, since it compares low-level bytecode that has already been transformed, organized, and optimized by the compiler, rather than the original high-level source code. This metric was chosen both to act as a lower bound on similarity and to enable comparison between contracts for which we do not have the original source code. **We discovered that 90% of the Ethereum smart contracts were at least 56% similar to each other.** About 7% were completely identical.



Ethereum contract bytecode contains embedded metadata such as hashes of the original source code as well as compilation configuration details. For example, this hash will vary if a single source code file is compiled twice with different indentations. These hashes *were*

not stripped from the binaries before performing the above comparison, nor were any constant operands (e.g., hard-coded contract addresses). This means that the true *semantic* similarity between the contracts could be much higher than pictured. This is because two codebases that vendor or copy/paste similar library code (e.g., OpenZeppelin or SafeMath, which are very popular) will be more similar if the hashes are ignored.

Conclusions

In this report, we identified several scenarios in which blockchain immutability is called into question not by exploiting cryptographic vulnerabilities but instead by subverting the properties of a blockchain's implementation, networking, or consensus protocol. A subset of a blockchain's participants can garner excessive, centralized control over the entire system. The majority of Bitcoin nodes have significant incentives to behave dishonestly, and in fact, there is no known way to create *any* permissionless blockchain that is impervious to malicious nodes *without* having a TTP. We provided updated data on the Nakamoto coefficient for numerous blockchains and proposed a new metric for blockchain centrality based on nodes' topological influence on consensus. A minority of network service providers—including Tor—are responsible for routing the majority of blockchain traffic. This is particularly concerning for Bitcoin because all protocol traffic is unencrypted and, therefore, susceptible to attacker-in-the-middle attacks. Finally, software diversity in blockchains is a difficult problem in terms of both upstream dependencies and patching.

Acknowledgements

This report was written by Trail of Bits based upon work supported by DARPA under Contract No. HR001120C0084 (Distribution Statement A, Approved for Public Release: Distribution Unlimited). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA. Many thanks to Molly White, Josselin Feist, and Artem Dinaburg for their helpful feedback.