Exploring Partitioning Attacks on the Bitcoin Network

Muhammad Saad, Victor Cook, *Member, IEEE*, Lan Nguyen, My T. Thai, *Fellow, IEEE*, and David Mohaisen, *Senior Member, IEEE*

***Abstract*—Bitcoin is the leading example of a blockchain application that facilitates peer-to-peer transactions without the need for a trusted third party. This paper considers possible attacks related to the decentralized network architecture of Bitcoin. We perform a data driven study of Bitcoin and present possible attacks based on spatial and temporal characteristics of its network. Towards that, we revisit the prior work, dedicated to the study of centralization of Bitcoin nodes over the Internet, through a fine-grained analysis of network distribution, and highlight the increasing centralization of the Bitcoin network over time. As a result, we show that Bitcoin is vulnerable to spatial, temporal, spatio-temporal, and logical partitioning attacks with an increased attack feasibility due to the network dynamics. We verify our observations through data-driven analyses and simulations, and discuss the implications of each attack on the Bitcoin network. We conclude with suggested countermeasures.**

***Index Terms*—Computer security, network security, distributed computing.**

I. Introduction

BITCOIN has been a lucrative target of attack for adversaries, who have been mainly targeting Bitcoin's exchanges and the Bitcoin peer-to-peer network. In this paper, we extend the security analysis of Bitcoin P2P network by exploring the partitioning attacks. In particular, through network data analysis (§IV), we uncover and exploit the increasing centralization of Bitcoin nodes over the Internet, the non-uniform consensus among peers, and the software diversity of Bitcoin clients to devise and optimize partitioning of the Bitcoin network. We outline spatial, temporal, spatio-temporal, and logical attacks, exploiting various aspects of Bitcoin dynamics. Some of those attacks are not new. For example, in 2014, an attacker from a malicious ISP hijacked IP prefixes of 19 Internet providers to isolate Bitcoin traffic and steal $83,000 USD worth of bitcoins [21], as an instance of the spatial attack. This attack has been formalized and examined in [1]. Our work shows that the network has become more vulnerable due to increasing centralization.

In 2017, 13 ASes hosted 30% Bitcoin nodes while 50 ASes hosted 50% Bitcoin nodes [1]. In our analysis, started in February 2018, we found that only 8 ASes host 30% of Bitcoin nodes and 24 ASes host 50% of Bitcoin nodes. At the organization-level, we found that only 13 organizations host 50% of the Bitcoin nodes. Among them, only two organizations host 65.7% of Bitcoin hashing rate, with the leading organization (*AliBaba*) having a 59.4% share of Bitcoin hashing rate. At the network level, we exploit the increasing centralization (§V-A) to show empirically that an adversary can easily partition the network *spatially* through BGP hijacking. At the AS level, we show a pattern of IP prefix distribution: in some cases, hijacking as little as 20 prefixes would give the adversary control over 80% of the Bitcoin nodes residing within this AS. At the organization-level, we uncover that multiple ISPs control more than one AS, amplifying the centralization effect, and facilitating new attacks.

Unique to our study, we exploit the non-uniform consensus among peers for optimized temporal attacks (§V-B). We observed that—due to latency—there is a lag in consensus and block propagation. Through our analysis, we found that even 5 minutes after the publication of a block, ≈62.7% of nodes in the network remain behind the latest block by one or two blocks. We show that such a behavior can be exploited to optimize an attack in which the adversary can feed false blocks to nodes and temporally partition the network. Considering the ethical ramifications of launching these attacks in practice, we instead use simulation-based models to validate our findings. Through simulations, we show that an attacker with ≈30% hash power can mislead nodes that are behind the main chain.

To optimize spatial and temporal attacks, we explore the spatio-temporal attack vector (§V-C). By observing that only 5 ASes hosted ≈30% of synchronized nodes, this attack considers them as more valuable targets, thus reducing the attacker's effort. Observing the presence of more than 200 Bitcoin software versions, demonstrating high software diversity, we outline a logical attack, in which an adversary manipulates the client behavior to partition the network (§V-D).

Little work has been done on measuring temporal behaviors in the Bitcoin network for attacks. Apostolaki *et al.* [1] performed a data analysis on Bitcoin to understand AS-level

Manuscript received November 26, 2020; revised March 10, 2021 and July 6, 2021; accepted August 10, 2021; approved by IEEE/ACM Transactions on Networking Editor N. Zhang. Date of publication August 24, 2021; date of current version February 17, 2022. This work was supported in part by the Air Force Material Command Award under Grant FA8750-16-0301, in part by the CyberFlorida Seed Grant (2021), and in part by the Global Research Laboratory Program of the National Research Foundation under Grant NRF-2016K1A1A2912757. An earlier version of this work has appeared in ICDCS 2019 [DOI: 10.1109/ICDCS.2019.00119]. *(Corresponding author: David Mohaisen.)*

Muhammad Saad, Victor Cook, and David Mohaisen are with the Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: mohaisen@ucf.edu).

Lan Nguyen and My T. Thai are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA.

Digital Object Identifier 10.1109/TNET.2021.3105604

centralization of nodes and miners, and presented the possibility of routing attacks. However, their work was limited to spatial attacks at vantage points on the Internet, which we demonstrate more effective due to network centralization.

Contributions and Roadmap. In summary, we make the following contributions. 1) Through data-driven analysis, we provide deeper insights into the Bitcoin network anatomy by outlining characteristics and distribution of full nodes. 2) Consolidating various insights from our measurements, we study four partitioning attacks on the Bitcoin network: the spatial, temporal, spatio-temporal, and logical partitioning attacks. Our longitudinal analyses capture the varying dynamics of the Bitcoin network over three years using which we examine the feasibility of each attack over time. 3) We discuss countermeasures to address those attacks.

In addition to the aforementioned contributions that also appeared in [34], this paper extends our analysis as follows: (1) We supplement our work in [34] with new dataset collected from 2018 to 2020. (2) Using that dataset, we conduct a longitudinal study to analyze the changing patterns in the spatial distribution of Bitcoin nodes and the network synchronization. Our results show that in 2019, the Bitcoin network was highly vulnerable to both spatial and spatio-temporal partitioning attacks since only 11 ASes hosted 50% nodes, and the average number of synchronized nodes among the top three ASes was the highest (≈1917). Moreover, we also observe that in 2020, the Bitcoin network has become more vulnerable to the temporal partitioning attack since the average network synchronization has significantly decreased to 64%. (3) Additionally, we also strengthen our prior analysis on the logical partitioning attack by conducting a vulnerability scan on Bitcoin full nodes. Our results show that ≈24% nodes are vulnerable to at least one software vulnerability. Among them, 70.57% are vulnerable to "CVE-2018-15919" which is an OpenSSH vulnerability that allows an adversary to detect the existence of users on the target system [13].

Through the rest of the paper, in §II, we outline the Bitcoin network model, and in §III, we outline the threat model. We provide our preliminary analysis in §IV. In §V, we discuss the partitioning attacks on Bitcoin network and in §VI, we explore the possible countermeasures for each attack. That is followed by related work and conclusion in §VII and §VIII, respectively.

II. The Bitcoin Network Model

The Bitcoin network consists of nodes connected in a peer-to-peer model. Upon joining the network, nodes connect to each other using public IP addresses, and use the gossip protocol to exchange network information such as transactions, blocks, and addresses. There are special nodes in the network, called *miners*, that extend the blockchain by creating new blocks [31].

Ideally, all the participating nodes in the network need to have an updated blockchain ledger, but the growing size of the chain makes it infeasible to be used on smart devices. To address this problem, Bitcoin also uses a concept of lightweight clients or SPV clients that run on a smart device and obtain the blockchain information by connection to the full nodes. Therefore, the current Bitcoin network is structured into full nodes that are active in the main network, and lightweight nodes that use services of full nodes. In Figure 1, we provide an illustration of this model. For more information regarding the full, we refer the reader to [19].

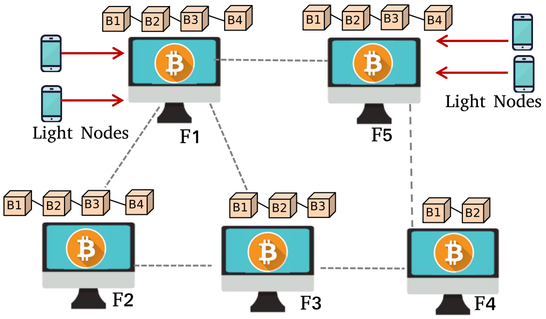


Fig. 1. Bitcoin network with full nodes and lightweight nodes. Lightweight nodes use the view that their associated full nodes provide. Full nodes F1, F2, and F5 have updated views while F3 and F4 are 1-2 blocks behind.

III. Threat Model

In this section, we outline the basics of partitioning attacks on Bitcoin and describe our threat model. Towards that, we revisit Apostolaki *et al.*'s work [1] (referred to as the "classical attack"), providing a baseline for partitioning attacks. We highlight new targeted attacks on the network, by introducing temporal and spatio-temporal attacks, which have not be identified before.

For the spatial partitioning, we assume the adversary to be an autonomous system (AS), an ISP organization, or a nation-state. An AS hosting a fewer Bitcoin nodes can launch a BGP attack on another AS that hosts more nodes. As a result, it can hijack the Bitcoin traffic, isolate the mining power, or simply harm the reputation of the target AS. For temporal attacks, we assume a malicious mining pool that attempts to fork the network and deprive an honest miner from block rewards. With soft forks, the adversary aims to create a temporary imbalance in system ramifications, such as transaction processing, and by hard forks it attempts to permanently split the network with diverging views. Additionally, due to the centralization of Bitcoin traffic and a shift in country-level policies towards Bitcoin, we do not exclude the possibility of a nation-state adversary.

Adversarial Capabilities. In the threat model, adversaries have unique capabilities. For example, a malicious AS or organization will have the ability to announce false routing information to other ASes and separate the target AS from neighboring nodes. This, in turn, can disrupt the exchange of transactions, blocks, and mining information, thereby affecting all the network nodes.

For temporal partitioning, the adversarial mining pool will have a consistent view of the network, which will allow it to identify nodes that are behind the blockchain. *Obtaining this information is not challenging since various Bitcoin crawlers are available and can be used to access the blockchain view of nodes* [9]. This can be exploited by the malicious mining pool to identify vulnerable nodes that are one or more blocks behind. A malicious miner, for instance, can mislead those nodes by propagating false information in the network. Doing so may create a partitioning in the network, where a group of nodes are misled into following a counterfeit blockchain.

For the spatio-temporal partitioning, we assume an adversary capable of announcing BGP prefixes and mining Bitcoin blocks. The adversary targets the synchronized nodes through spatial partitioning and the non-synchronized nodes through temporal partitioning attacks. Finally, for logical partitioning, we assume an adversary capable of exploiting bugs in Bitcoin or releasing a new Bitcoin version with improved functionalities.

Attack Objectives. The Bitcoin blockchain security strongly relies on the network's capability of preventing forks that cause inconsistency in consensus. Moreover, forks are also undesirable for the mining pools since they waste the work of honest miners. Through partitioning attacks, the adversary tries to violate the blockchain consistency and waste the effort of honest miners. In the following, we describe the adversary's objectives in each partitioning attack.

In spatial partitioning, the adversary aims to isolate a group of nodes in order to (1) prevent users from generating transactions or receiving blocks, and (2) reduce the network's hash rate (*i.e.,* if nodes belong to the mining pools). In temporal partitioning, the adversary subverts a group of nodes by feeding them the counterfeit blocks and creating forks in the network. In the spatio-temporal partitioning, the adversary ensures that forks persist for a long time, allowing the adversary to feasibly double-spend. Finally, in the logical partitioning, the adversary forces the vulnerable nodes to follow different rules from other nodes (*i.e.,* accept double-spent transactions).

IV. Preliminary Analysis

A. Data Collection

For our analysis, we crawled data from Bitnodes [9], which is a Bitcoin service supported by *Earn.com* [10]. Bitnodes maintains a persistent connection with all reachable nodes by running a full node that connects to the rest of the network. For each node, Bitnodes records useful information such as the latency, the uptime, and the latest block etc. From IP addresses, it determines the corresponding AS, organization, and location of nodes. We developed another crawler, atop Bitnodes, to collect Bitnodes data. We ran the crawler for two months and sampled the network snapshot at 10 minutes interval.

While the aforementioned dataset was used in our initial study [34], we continued our data collection until May 2018 to obtain more comprehensive results. Later, in August 2019, we resumed our data collection and continued till May 2020. As a result, for each year, we had five months of data from Bitnodes. The dataset included network snapshots sampled at 10 minutes interval. We used that dataset to conduct a longitudinal analysis of changing patterns in the Bitcoin network, including the distribution of nodes across ASes and the network synchronization. The results are provided in §V-A1, §V-B1, and §V-C1.

TABLE I

A View of Top Ten ASes and Organizations in Bitcoin on February 28th 2018. The Table Shows That Bitcoin is More Centralized With Respect to Organizations Than ASes. AS24940 Intercepts the Maximum Bitcoin Traffic

ASes	Nodes	Nodes %	Organizations	Nodes	Nodes %
AS24940	1,030	7.54%	Hetzner	1,030	7.54%
AS16276	697	5.11%	Amazon	756	5.54%
AS37963	640	4.69%	OVH SAS	700	5.13%
AS16509	609	4.47%	Hangzhou	640	4.69%
AS14061	460	3.37%	DigitalOcean	503	3.69%
AS7922	414	3.04%	Comcast	414	3.04%
AS4134	394	2.89%	Jin-rong Street	394	2.89%
TOR	319	2.34%	TOR	319	2.34%
AS51167	288	2.11%	Contabo	288	2.11%
AS45102	279	2.05%	Alibaba	279	2.05%

B. Methodology

First, we analyzed the distribution of nodes across ASes and organizations. The initial results gave us a holistic view of the network and its centralization, which we used to describe spatial partitioning attacks. Next, we analyzed the network synchronization by analyzing the blockchain view of each node. We recorded the latest block published by miners in the network and the most recent block that every node had. The difference between the two denoted how far behind the node was from the network. As shown in Fig. 1, nodes F3 and F4 are 1-2 blocks behind the main chain. We leveraged this information to outline temporal partitioning attacks that can be launched on Bitcoin network to isolate nodes based on their outdated view.

C. Measurements and Observations

Below, we discuss some key observations we made during the preliminary analysis on the Bitcoin network on February 28, 2018. The network snapshot showed that there were 13,635 full nodes in the network out which 11,382 (83.47%) nodes were up. Only 6,155 (45.14%) nodes had the most updated copy of the blockchain while 7,480 (54.86%) were 1 or more blocks behind. Among the full nodes, 12,737 (93.41%) had IPv4 address, while 579 (4.24%) had IPv6 address. The remaining 319 (2.33%) full nodes had onion addresses [29], meaning that they were using TOR services to run Bitcoin. During the two months data collection, the average number of nodes that were up was ≈10K.

V. Partitioning Attacks on Bitcoin

Based on our preliminary analysis, in this section, we present four partitioning attacks on the Bitcoin network.

A. Spatial Partitioning

In this section, we analyze the centralization of full nodes and mining pools across ASes and organizations. Towards that, we revisit the prior work to evaluate the classical attack, and demonstrate that over time, the Bitcoin network has further centralized and become more vulnerable to the attack.

Attack Objectives. The objective of spatial partitioning is to isolate Bitcoin nodes. The objective can be purely to isolate miners, and restricting their access to the network, or eclipsing an entire AS that hosts a large fraction of

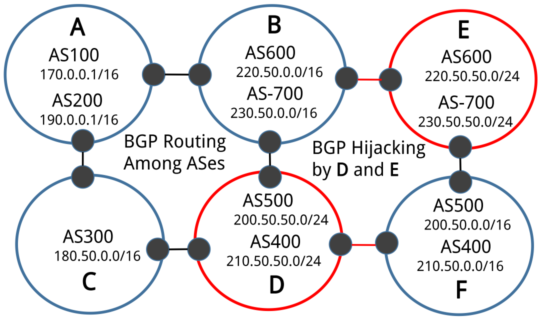


Fig. 2. Network topology consisting of organizations, ASes and full nodes. Organizations D and E can launch BGP attacks against F and B respectively.

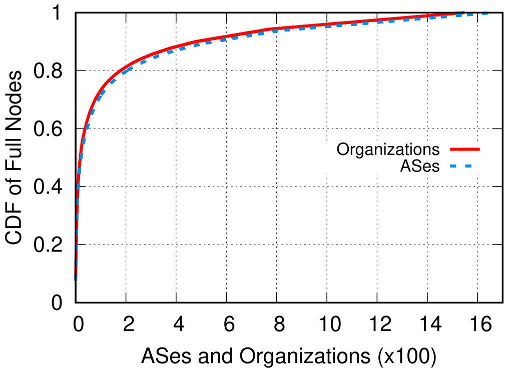


Fig. 3. Distribution of Bitcoin full nodes across ASes and organizations. Note that nodes are more centralized across organizations than ASes.

nodes. A mining pool might launch such an attack against its competitor to increase its chances to publish more blocks. A competing cryptocurrency can launch this attack to affect Bitcoin's reputation.

Attack Procedure. In Figure 2, we provide an illustration of a BGP attack, which can be launched by a malicious organization or an AS. In this attack, the malicious AS announces prefixes that belong to the victim AS. As shown Figure 2, organizations D and E can launch BGP attacks against organization F and B, respectively, by broadcasting more specific prefixes. Moreover, the attack can be made more targeted by announcing prefixes addressing only Bitcoin nodes. This attack relies on two major factors: the total number of ASes and organizations, and the total number of nodes hosted in each of them. In particular, if the total number of ASes and organizations hosting full nodes is large, the attack becomes costly. Similarly, if the number of nodes is concentrated within a few ASes, that makes a better target rather than attacking arbitrary ASes with fewer nodes. To evaluate that, we carried out two experiments to observe the total number of ASes hosting Bitcoin nodes and the distribution of nodes among those ASes.

Practical Considerations. Our results show that the full nodes in Bitcoin are highly centralized at the AS and organization level. Compared to [1], the network has become even more centralized, and more vulnerable to BGP hijacking and routing attacks. In particular, we observed that among the total of 84,903 ASes in the world [32], only 8 (0.0094%) ASes host 30% Bitcoin nodes. 24 (0.028%) ASes host 50% while 1,660 (1.95%) ASes host 100% Bitcoin nodes. This shows a significant difference in the number of ASes that host 50% and 100% full nodes. To understand that, we plot CDF of ASes that host the traffic of full nodes in Figure 3.

TABLE II

Top 5 Mining Pools per Hash Rate, ASes, and Organizations. 65.7% Mining Data Goes Through Only Three Organizations. Alibaba Has a View of at Least 60% of the Mining Data. We Exclude the Remaining 12 Mining Pools From the Study as Their Total Contribution to Hash Rate Is Minimal

Mining Pool	H. Rate %	ASes	Organizations
BTC.com	25%	AS37963 AS45102	Hangzhou Alibaba AliBaba (China)
Antpool	12.4%	AS45102	AliBaba (China)
ViaBTC	11.7%	AS45102	AliBaba (China)
BTC.TOP	10.3%	AS45102	AliBaba (China)
F2Pool	6.3%	AS45102 AS58563	AliBaba (China) Chinanet Hubei
12 others	34.3%	—	—

Similarly, we observed that the top 8 organizations intercepted 30% Bitcoin traffic and the top 13 organizations intercepted 50% traffic, collectively. We also noticed that each organization controlled one or more ASes, alluding to the possibility of a fine-grained partitioning attack.

In Table I, we show the top 10 ASes and organizations along with the percentage of total nodes that they host. We group TOR nodes and treat them as a single AS. AS24940 hosts 7.54% nodes and its corresponding organization *Hetzner Online* also hosts 7.54% nodes, meaning that the Bitcoin traffic routed by Hetzner Online entirely goes through AS24940. On the other hand, Amazon.com routes 5.54% of the traffic while AS16276 intercepts 5.11% traffic. This shows that Amazon.com owns another AS besides AS16276 that also routes traffic. This model can be observed in Figure 2.

Mining pools are another important part of Bitcoin, since they are responsible for extending the blockchain and maintaining its state. Mining pools consist of miners on the Internet communicating via a special mining protocol known as the "Stratum Mining Protocol" [8]. All miners compute PoW and send the result to the stratum server address specified by the mining pool. The stratum address is made public by the mining pool. As such, if the link to the stratum server is compromised, the mining pool gets disconnected and its aggregate hash rate decreases. To analyze the distribution of stratum servers, we carried out two experiments. First, we gathered information about major mining pools in Bitcoin and their hash rate from *Blockchain.info* [5]; results are reported in Table II. Next we selected the top 5 mining pools, which had an aggregate hash rate of 65% of the total in the Bitcoin network. We then collected the stratum address of the selected mining pools from their websites and traced the IP address corresponding to each stratum address [6]. We mapped each IP address to the AS hosting the stratum server. We found that 3 ASes had 65% of Bitcoin mining pool traffic while one organization "AliBaba" alone had more than 50% of the Bitcoin mining pool traffic. We report our results in Table II. In the light of our threat model, and given an adversary capable of BGP hijacking, policy enforcement at an organization level, or collusion, having an organization hosting more 50% of the mining power makes such an attack even more effective.

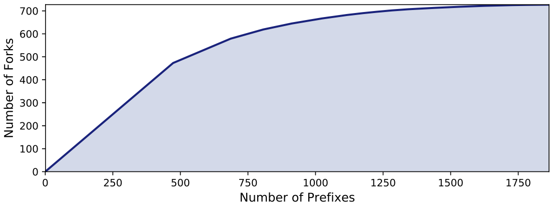


Fig. 4. The number of forks created by hijacking the minimum number of prefixes in the current network.

Analytics Framework. In order to comprehensively understand the dynamics of the spatial partitioning attacks, we developed an analytics framework to perform real-time analysis of the spatial partitioning attacks in the Bitcoin network. In our analytics framework, we collect the IP addresses of all the *reachable* Bitcoin nodes present in the network at any time. We then obtain the corresponding ASes that host those nodes along with their IP prefixes. For each AS, we divide the total number of nodes by the total number of prefixes to analyze the attack feasibility. We then define the attack feasibility as the adversary's ability to isolate the largest number of nodes by hijacking fewer prefixes. Through the complete isolation of nodes in an AS, the adversary creates a fork between the victim nodes and the rest of the network [1].

To illustrate the attack feasibility, consider three ASes (AS A, AS B, and AS C) that host Bitcoin nodes. Assume that AS A hosts 100 nodes in 100 unique prefixes, AS B hosts 50 nodes in 15 unique prefixes, and AS C hosts 50 nodes in 25 unique prefixes. In a naïve attack, AS A could be the most lucrative target for the adversary since it hosts more nodes than the other two ASes. However, attacking AS A can be costly since it would require the adversary to announce up to 100 prefixes in order to fully isolate all the nodes and create a blockchain fork. In contrast, if the adversary simply announces 15 prefixes of AS B and 25 prefixes of AS C, the adversary can isolate up to 100 nodes by hijacking only 40 prefixes. Moreover, by isolating nodes in two distinct ASes (AS B and AS C), the adversary can create up to two blockchain forks. [1] Through our proposed framework, we analyze the number of Bitcoin blockchain forks that can be created if the adversary follows the aforementioned strategy.

In Figure 4, we plot results obtained from our analytics framework. The y-axis in Figure 4 shows the total number of forks and the x-axis shows the number of cumulative prefixes the adversary needs to hijack in order to create the required number of forks. Our results show that an adversary can fork the Bitcoin blockchain 700 times by hijacking up to 1.8K BGP prefixes. Our analytics frameworks is available for evaluation in [35].

Implications. Spatial partitioning is detrimental to the Bitcoin network as it facilitates other major attacks including double-spending attacks, eclipse attacks, and the 51% attack. As shown in Table II, if an attacker hijacks 3 ASes, he can isolate more than 60% of the Bitcoin hash power. As Figure 4 shows that by hijacking 15 BGP prefixes, the attacker can cut 95% traffic of AS24940 that hosts 1,030 full nodes. By isolating the hash power, an attacker can cause delays in the block creation and the transaction confirmation.

TABLE III

LONGITUDINAL ANALYSIS OF NODE HOSTING PATTERNS ACROSS ASES FROM 2018 TO 2020. IN 2018, 3.23% NODES WERE USING TOR. IN 2019, THAT NUMBER DECREASED BELOW 1.50%, AND IN 2020 IT INCREASES TO 25.8%

2018		2019		2020	
ASes	**Nodes**	**ASes**	**Nodes**	**ASes**	**Nodes**
AS24940	9.57%	AS24940	12.15%	TOR	25.8%
AS16276	5.94%	AS14061	7.57%	AS24940	10.52%
AS16509	5.42%	AS16509	7.23%	AS16509	6.00%
AS14061	3.62%	AS16276	5.54%	AS14061	4.79%
TOR	3.23%	AS20473	4.92%	AS16276	4.40%
AS51167	2.67%	AS7922	2.66%	AS63949	2.73%
AS7922	2.58%	AS15169	2.17%	AS15169	1.99%
AS37963	2.56%	AS51167	2.08%	AS51167	1.83%
AS4837	2.42%	AS45102	1.89%	AS7922	1.63%
AS15169	1.82%	AS14618	1.50%	AS45102	1.07%

If the attacker is a mining pool with lower hash rate, it can launch the attack on competing mining pools and deprive them of their mining rewards. By isolating a majority of the network's hash power, the attacker can launch the 51% attack on Bitcoin which will grant him a permanent control over the blockchain. Furthermore, in peer-to-peer settings, nodes are responsible to relay blocks and transactions to each other. By hijacking a subset of nodes, the attacker can introduce a cascade effect in which propagation of blocks and transactions can be stalled; the attacker does not have to isolate all nodes by hijacking all BGP prefixes in an AS. Isolating a major subset of nodes can eclipse the entire AS.

1) Longitudinal Analysis of Spatial Partitioning: So far, our analysis of the spatial partitioning attack shows the Bitcoin network state in February 2018. As mentioned in §IV-A, we continued our data collection until 2020 to observe changes in the Bitcoin nodes' hosting patterns and their vulnerability to the BGP attacks. For each year, we report the average results in Table III and derive the following key conclusions.

The first key observation is that the number of Bitcoin nodes using Tor has fluctuated in the last three years. In 2018, the average number of nodes using the Tor network was 3.23%. In 2019, that number decreased below 1.50%, and in 2020, the number increased to 25.8%. This shows that in 2020, approximately a quarter of the total network started using Tor. This could be due to the fact that the threat of spatial partitioning has become well-known in the Bitcoin community [34]. Since Tor shields the public IP address, therefore, an adversary cannot precisely know the location of the node in an AS. This provides a certain degree of protection against the spatial partitioning attack. Therefore, the number of Tor nodes has increased over time.

The second key observation is that the dominant ASes in 2018, including AS24940, AS16276, AS16509, and AS14061, also remained dominant in 2019 and 2020. Although, the number of nodes that they host each year has changed, however, they still host a significant number of Bitcoin nodes. In 2018, AS24940 hosted 9.57% nodes, which then increased to 12.15% in 2019, and later decreased to 10.52% in 2020.

[1] We assume there is at least one mining node in each AS.

The third key observation is that the degree of centralization has changed over time. In 2018, 24 ASes hosted 50% nodes. That number dropped to 11 ASes in 2019, making the network highly vulnerable to spatial partitioning attacks. However, since 2020, the diversity in node hosting pattern has increased, and 49 ASes host 50% nodes. This is similar to the network state in 2017 [1] where 50 ASes hosted 50% nodes. One factor that has contributed to this diversity is that a significant number of nodes (25.8%) has switched to the Tor network which shields the information about their ASes.

Overall, we conclude that since 2018, the network has witnessed significant variations in terms of node hosting patterns across ASes. While on average, the dominance of a few ASes has persisted, the degree of centralization has changed over time. In 2019, the Bitcoin network was the most vulnerable to spatial partitioning attacks.

B. Temporal Partitioning

Temporal partitioning involves isolation of a group of nodes in the network that are some blocks behind the rest of the network. As shown in Figure 1, three nodes have the most updated copy of the blockchain, while nodes F3 and F4 are 1–2 blocks behind. These nodes might be behind the main chain due to a number of reasons, such as the network latency, a low bandwidth, software malfunctions, or a malicious peer. Therefore, these nodes have an outdated view of the blockchain and remain vulnerable to partitioning attacks.

Attack Objectives. The objective of the temporal partitioning is the isolation and subversion of nodes or a group of nodes within the network. Latency in updating the blockchain is a well known vulnerability of Bitcoin, which is confirmed in our data. Propagation delays are known to be key contributors towards the latency [14]. Propagation delays are influenced by the number of hops between nodes due to sparse peering, and the time required by software clients to verify and forward a block. Solutions have been proposed that cluster nodes to reduce latency [37], but the authors note this may increase the potential for partitioning attacks. This indicates a trade-off between spatial and temporal vulnerability. Also contributing to the node latency are communication failures and the behavior of nearby peers. The adversary would seek to disrupt communication and control peers where the attack is launched. It is inexpensive to setup new nodes on the Bitcoin network for this purpose. The adversary would want to separate and control nodes which are not up to date with the main network. Under normal operation, those nodes might eventually catch up with the network, but an adversary will prevent that from happening.

Attack Procedure. Analysis of Bitcoin nodes over a period of days shows several times a day when a significant fraction of nodes are not up-to-date. We report our findings in Figure 5. In Figure 5, the x-axis denotes a time-index for network observations (one observation every 10 minutes in Figure 5(a) and Figure 5(b), and one every minute in Figure 5(c)). The y-axis is stacked, meaning that curves are cumulative. The green part shows nodes that are up-to-date, the yellow part shows nodes that are 1 block behind, and the purple part shows nodes that are 2-4 blocks behind. The remaining colors and their descriptions are in the figure.

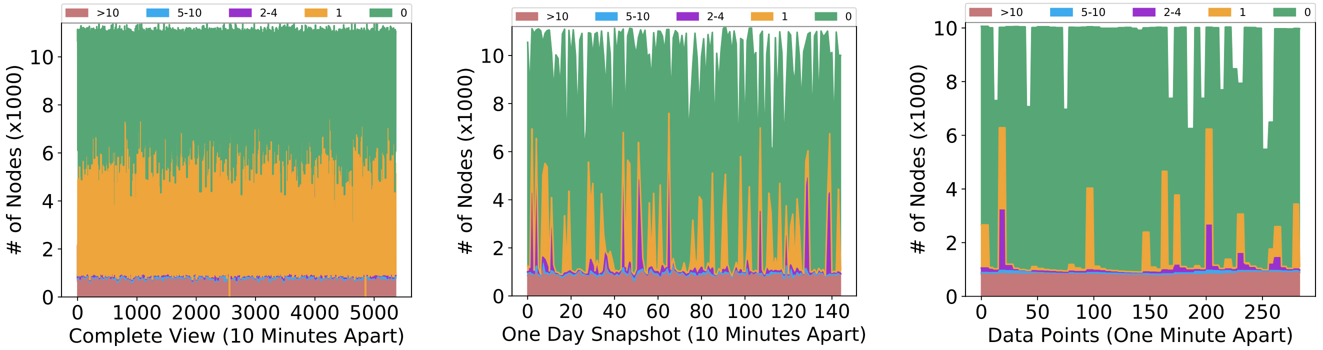
From Figure 5(a), we were able to make following observations. 1) Generally, a majority of nodes ($\approx 50\%$) remains synchronized on the blockchain state. These nodes do not lag behind in the main chain for a long duration. 2) 10% nodes are forever behind the main blockchain. They do not update their blockchain and as such, they have no benefit in the network. 3) 30-40% nodes in Bitcoin occasionally waver in terms of their view of the blockchain. Possibly due to network latency or consensus delay, they lag behind the recent block.

To further study the distribution of consensus in the network, we take a single day snapshot of the network to observe consensus pruning among all nodes. From the view of an attacker, with higher granularity, there is a better vantage point to attack a group of nodes. Focusing on a single day shown in Figure 5(b), we observed that some yellow and purple spikes are larger and wider than others. The height of a spike denotes the count of nodes that are behind the updated nodes, while the width indicates the length of time for which they remain behind the updated nodes.

From Figure 5(b), with a closer look at the network, we made the following observations. 1) Consensus pruning is not uniform across the network. 2) The most frequent delay among the blocks is 1 block indicated by yellow region, followed 2-4 blocks, indicated by the purple region. 3) On various occasions, yellow and purple spikes can reach up to 7,000 nodes; approximately 90% of the network can be partitioned if an attacker isolates them.

In Bitcoin, on average, a block is published after every 10 minutes. Once a block is published, ideally the network is expected to be synchronized within 10 minutes before the next block is computed. However, network synchronization is an artifact of time and fairness of the network. In the previous two experiments, we observed that with fine grained sampling, on a given day, the attacker can isolate a group of nodes which are behind the main chain. To further analyze this behavior, we performed another experiment that involved per-minute sampling of network. Our objective was to observe the distribution of consensus among peers immediately after broadcast of one block and before the broadcast of the next one. We plot the results obtained from the third experiment in Figure 5(c). It can be observed in the figure that there are vulnerable spots in the network in which up to 90% of the network is 1-4 blocks behind. As such, the non-uniform consensus pruning presented itself as an attack opportunity whereby an attacker can find a time window to isolate a group of targeted nodes. In Figure 5(c), the width of nodes that are behind show the attack time window while the height represents the number of vulnerable nodes.

Theoretical Analysis. This becomes an optimization problem to find the moment where a majority of nodes is behind for the longest attack window. The attacker's timing constraints include the time to calculate false blocks and establish connections to vulnerable nodes. Hence, to identify vulnerable nodes, we formulate the temporal attack as an optimization model: *Given a timestamp t and a timing constraint* $\mathtt{T}$*, find the maximum number of vulnerable nodes whose lagging time*



(a) General trend of the network

(b) One day snapshot

(c) Consensus between block propagation

Fig. 5. Bitcoin network synchronization. Y-axis denotes number of nodes in 1000. In each figure, green region denotes the up-to-date blocks. Yellow region denotes 1 block behind. Purple, blue, and magenta regions represent nodes that are 2–4, 5–10, and $\geq$ 10 blocks behind respectively. Figure 5(a) shows the overall network, Figure 5(b), shows a day (March 25) that offers greater attack opportunity, and Figure 5(c) shows consensus during 10 minutes.

TABLE IV

THE MAXIMUM NUMBER OF VULNERABLE NODES

T (minutes)	$\geq$ 1 block	$\geq$ 2 blocks	$\geq$ 5 blocks
5	6280(62.67%)	3206(31.99%)	966(9.68%)
10	1761(27.13%)	1189(11.87%)	955(9.53%)
15	1141(11.39%)	1083(10.81%)	952(12.00%)
20	1109(13.97%)	1023(15.76%)	947(11.93%)
25	1070(10.68%)	1013(15.61%)	942(9.40%)
30	1042(10.39%)	984(9.82%)	942(9.39%)

$\mathtt{L}(t)$ *is at least* $\mathtt{T}$. *Lagging time* $\mathtt{L}(t)$ of a node is defined as minimum timing for this node to catch up to the main blockchain if it lags behind at t. The objectives of this formulation are as follows. (1) By identifying maximum nodes that were lagging concurrently, attacker could isolate them and mislead them with false blocks. (2) By investigating all possible timestamps, an attacker could find an optimal time to attack those nodes.

We identify nodes whose historical behaviors show their vulnerability to temporal attacks, and record their results in Table IV. Note that, at any time, the total number of nodes in Bitcoin fluctuates between 8k–13k. For any time window, we are interested in finding the maximum percentage of vulnerable nodes for that window.

With this information, we perform a theoretical analysis on the timing threshold $\mathtt{T}$ that is suitable for the attacker to isolate a targeted set of m nodes. We assume the attacker wants to isolate m nodes which requires the attacker to create connections to these nodes and feed them its own version of block. We model the required timing for this process as an exponential distribution by rate λ. In 2015, the Bitcoin community switched from a traditional gossip-style protocol known as *trickle spreading* to *diffusion spreading*, in which the information propagates with independent exponential delays. This method of modeling Bitcoin connections has been used in prior work as well, by Fanti and Viswanath [18]. Using that, the timing of the attacker to connect to a node is:

$$f(t) = \lambda e^{-\lambda t}, \quad F(t) = 1 - e^{-\lambda t} \tag{1}$$

where $f(\cdot), F(\cdot)$ are probability density and cumulative distribution functions. Given timing assigned to isolate m nodes is $\mathcal{T} = (t_1, \dots t_m)$. The probability that an attacker isolates m nodes under $\mathcal{T}$, derived from Cauchy inequality theorem is:

$$\rho(\mathcal{T}) = \prod_{i=1}^{m}(1 - e^{-\lambda t_i}) \leq \left(1 - \frac{\sum_{i=1}^{m} e^{-\lambda t_i}}{m}\right)^m \tag{2}$$

Theorem 1 (Cauchy Theorem): Let $x_1, x_2, \dots x_n$ are n non-negative numbers, then:

$$\prod_{i=1}^{n} x_i \leq \left(\frac{\sum_{i=1}^{n} x_i}{n}\right)^n \leq \frac{\sum_{i=1}^{n} x_i^n}{n} \tag{3}$$

Both equalities occur if and only if $x_1 = x_2 = \dots = x_n$

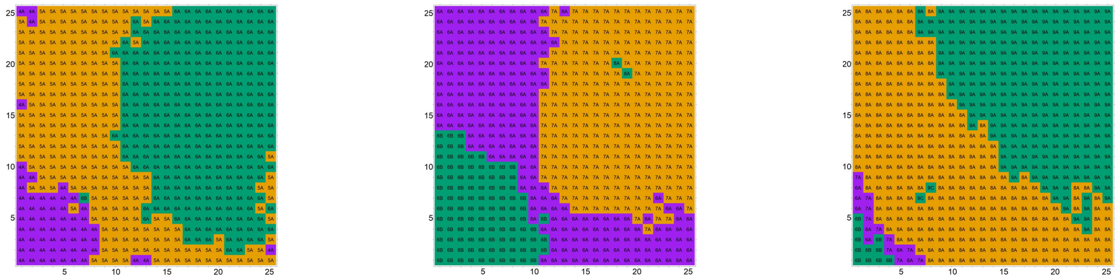
Now, consider a timing constraint $\mathtt{T}$, in which the attacker wants to isolate all m nodes. This means that the timing assignment $\mathcal{T}$ should satisfy $\sum_{i=1}^{m} t_i \leq \mathtt{T}$. So:

$$\rho(\mathcal{T}) \leq (1 - e^{-\frac{\lambda}{m}\mathtt{T}})^m \tag{4}$$

With timing constraint $\mathtt{T}$, the attacker will have at most $\binom{\mathtt{T}}{m}$ choices for timing assignment. By union bound, the probability p to isolate m nodes within $\mathtt{T}$ is bounded by:

$$p \leq \mathtt{b}(m, \mathtt{T}) = \binom{\mathtt{T}}{m}(1 - e^{-\frac{\lambda}{m}\mathtt{T}})^m \tag{5}$$

Given m, $\mathtt{b}()$ is monotonically increasing by $\mathtt{T}$. Therefore, given a successful probability p, we can infer a lower bound of $\mathtt{T}$ by binary bisection. We experiment with the relationship among values of m, $\mathtt{T}$, and λ. We set the targeted successful rate of attacker p as 0.8, and test it with various values of λ. The results are recorded in table V. Column labels show different values of m nodes that the attacker aims to isolate, and row labels show values of λ. Values in each cell denote the bound of $\mathtt{T}$ such that within this bound, the attacker can isolate m nodes under delay rate λ with probability of at least 0.8. For example, with $\lambda = 0.8$ and $m = 500$, it would take only 589 seconds (approximately 10 minutes) to isolate all m nodes with probability at least 0.8. 500 is much smaller than number of vulnerable nodes in 10 minutes timing constraint (from table Table V, there can be 1,761 vulnerable nodes within $\mathtt{T} = 10$ minutes). Our results indicate that even after 10 minutes, a group of nodes can still be isolated through temporal partitioning. In a prior measurement study conducted in 2013 [14], the authors observed that all Bitcoin



(a) Time Step 151 (b) Time Step 201 (c) Time Step 251

Fig. 6. Simulation of temporal attack. Figure 6(a) shows fork B emerging at node [7,7]. Compare the color distribution to the peaks of Figure 5(c) above. Two blocks later in Figure 6(b) fork B has control of 1/6 of the nodes. In Figure 6(c) the longer chain A overwhelms fork B but has lost synchronization so cannot prevent emergence of a new fork C.

TABLE V

MINIMUM TIMING CONSTRAINT T (SECONDS) TO ISOLATE m NODES UNDER THE GIVEN RATE λ

λ \ m	100	300	500	800	1000	1200	1500
0.4	142	424	705	1127	1610	2313	3517
0.5	133	397	661	1057	1320	1851	2814
0.6	127	379	630	1007	1258	1545	2345
0.7	122	365	607	970	1213	1455	2010

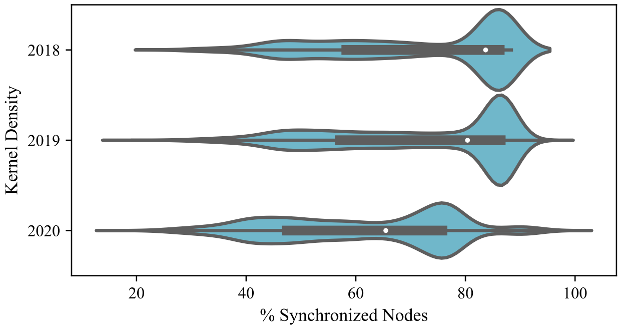


Fig. 7. The Bitcoin network synchronization in 2018, 2019, and 2020. The synchronization is represented as the kernel density estimation in which the distribution shape denotes the synchronization pattern. In 2020, network synchronization has significantly decreased.

nodes receive blocks within 11 seconds. In other words, the time window for the temporal partitioning attack was limited to 11 seconds only. Our measurements show that the time window has significantly increased, making the network more vulnerable to the temporal partitioning attack.

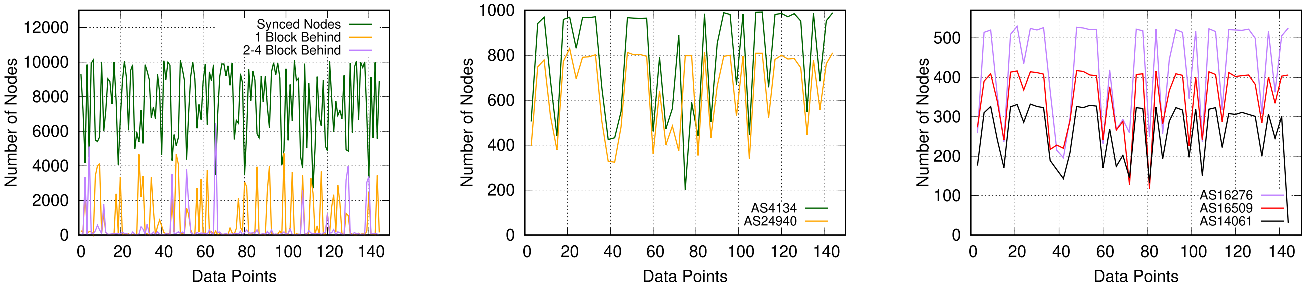
Simulation and Attack Validation. To validate the insights obtained from our data and theoretical analyses, we developed a simulator to evaluate the temporal partitioning attacks. The simulator was developed with the following objectives: (1) it must accurately model the network behavior as observed in our measurements, and (2) it must be lightweight and easily configurable across multiple platforms with varying computational capabilities. To meet these objectives, we first modeled the network synchronization pattern in our simulations by using data from Figure 5. Second, we reduced the computation complexity by using MD5 hash function instead of SHA256. Since MD5 is faster than SHA256 [25], we saved the processing power to simulate a longer block race and observed forks caused by the temporal partitioning attack.

The simulator was tested in base simulation scenarios, such as zero and perfect communication among nodes. Each simulated node maintains a 64-bit MD5 hash linked chain of values updated to its current fork. The default number of Bitcoin peers is 8, which is used in our simulation. Studies have shown that peers are distributed, and can be associated with any AS [17]. Our experimental data confirmed this distribution. Following this, the peers were evenly distributed in terms of communication errors and latency. Peer communication failure rate was represented by a model parameter, typically around 10 percent failures. The latency was represented by the number of communication time steps per simulation block. Each time step represented one peer-to-peer communication attempt for each node. The simulator code and results are available in [36], and the simulator has been updated with a SHA256-based implementation.

Figure 6 shows a sample of the simulation results, where the attacker has 30% of the network hash rate. Once a portion of the network is isolated, it can be sustained with successive forks, since the isolated nodes naturally assume that block delays are due to network issues. As such, they do not know that new blocks are taking more time to calculate due to the lower hash rate of the attacker. Meanwhile, the main chain loses some of its hash rate and is therefore, less capable of responding. Note that the cost of launching a temporal attack is much less than the spatial attack, provided that the attacker has the consistent view of the network as shown in Figure 5.

Implications. Even a short term fork can cause sufficient disruption to invalidate transactions and blocks. Such an attack wastes the effort of honest miners and reduces the *effective* network hash rate. Our simulations show that, in the current network, an attacker with 30% hash rate can fork the chain and target other miners. If the fork persists for more than six blocks, the adversary can launch a double-spend attack [30]. However, maintaining a fork for six blocks is challenging if the adversary only relies on block propagation delay. In §V-C, we will show how the fork can be sustained for six or more blocks through spatio-temporal partitioning attack.

1) Longitudinal Analysis of Temporal Partitioning: Similar to the analysis in §V-A1, we conducted the longitudinal study of Bitcoin network synchronization using our new dataset §IV-A. For this experiment, we sampled the percentage of nodes with an up-to-date blockchain in each network



(a) One day snapshot (b) Top 1-2 synced nodes ASes (c) Top 3-5 synced nodes ASes

Fig. 8. Spatial and temporal distribution of nodes for the day defined in Figure 5(b). For the synced nodes in Figure 8(a), we outline their distribution across top five ASes in Figure 8(b) and Figure 8(c).

snapshot. We then calculated the kernel density estimation to monitor the network synchronization pattern in each year. We report results in Figure 7 and make the following key conclusions.

Our results show that the Bitcoin network synchronization has deteriorated over time. In 2018, on average, 72.25% nodes had an up-to-date blockchain with a median percentage value of 83.68%. In 2019, 72.01% nodes had an up-to-date blockchain with a median percentage value of 80.38%, showing a marginal decrease in network synchronization. However, in 2020, we observed a sharp decline in the number of synchronized nodes. On average, 61.91% nodes had an up-to-date blockchain with a median percentage value of 65.47%. This shows that in 2020, the Bitcoin network has become increasingly vulnerable to the temporal partitioning attacks and malicious miners can easily subvert a group of nodes into following a counterfeit blockchain.

Although, we do not know the root cause of deteriorating network synchronization, however, it could be due to the increasing number of nodes using Tor. Tor circuits add delay in the information propagation [15], and in Bitcoin, that information includes blocks relayed among peers. Since the number of Tor nodes has increased from 1.5% to 25.8%, therefore it could be a reason for decreasing synchronization. As discussed in §V-A1, Tor provides a certain degree of protection against spatial partitioning attacks. However, in light of our new results, it seems to be affecting network synchronization, thereby increasing the risk of temporal partitioning attacks. Therefore, using Tor provides a trade-off in countering the spatial or temporal partitioning attacks.

C. Spatio-Temporal Partitioning

In this section, we analyze how an attacker can make use of spatial and temporal distribution of nodes over time to find vulnerable spots in the network, through which he can effectively isolate a group of nodes. From our data analysis, we found the feasibility and cost of this attack compared to spatial and temporal partitioning. Saptio-temporal analysis also provides insights into the general behavior of nodes within an AS or an organization. Therefore, it is intuitive to investigate the attributes of the overall topology of Bitcoin network in relation to the ASes and organizations.

Attack Objectives. In this attack, the aim of the adversary is to split the network based on the network's vulnerability to both the spatial and temporal partitioning. As shown in Figure 5(a) and Figure 5(b), the purple and yellow nodes are vulnerable to temporal attacks. However, the attacker cannot launch the same attack on nodes lying in the green region (synced nodes), since they are up-to-date and will reject a false block. These nodes can still be partitioned based on the BGP attack presented in spatial partitioning. A combined effect of both attacks will be an optimized and targeted attack that will affect the entire Bitcoin network.

It is worth mentioning that for a BGP attack on nodes within the green region, the attacker does not have to isolate all target nodes. Since these up-to-date nodes are connected with each other, therefore, an attack on a subset of nodes can have a cascade effect, thereby compromising all other nodes.

Attack Procedure and Validation. For a successful attack, the attacker needs information about ASes and organizations of the synced nodes as well as nodes that are behind. The feasibility of this attack depends on the adversarial capabilities of the attacker. To analyze that, we elaborate the network behavior from Figure 5(b) in Figure 8(a). The green line shows the nodes that are synced, while yellow and purple lines show nodes that are 1 block and 2–4 blocks behind, respectively.

Per our threat model, if the attacker is an AS, it will prefer to hijack BGP prefixes to damage Bitcoin. As such, it will prefer maximum nodes in the green region and minimum nodes in yellow and purple region, to maximize the attack severity. If the attacker is a mining pool, then it will launch a temporal attack, and will prefer minimum nodes in green region and maximum nodes in other regions. However, if the attacker is a cloud service provider that has both routing and mining capabilities, then it can launch both spatial and temporal attacks. Therefore, the key aspect of spatio-temporal attack is that it is adjustable to the attacker's capabilities.

Although multiple attack scenarios and case studies can be drawn for spatio-temporal partitioning but in the interest of space, we illustrate one case study. From our simulations, we observed that the temporal partitioning forks the network at a faster rate than spatial attacks. Therefore, we assume a case in which cloud provider waits for minimum number of synced nodes, and launches a spatio-temporal attack. As seen in Figure 8(a), at two instances, the number of synced nodes falls as low as 3,000, while the number of nodes that are 2–4 blocks behind go as high as 6,000 nodes. This can serve as an ideal attack opportunity to launch the spatio-temporal

TABLE VI
TOP 5 ASES THAT HOSTED ALL THE SYNCHRONIZED NODES IN FIGURE 5(b) FOR 24 HOURS

AS	Organization	Nodes	Percentage
AS4134	No.31, Jin-rong	993	9.57%
AS24940	Hetzner Online	830	7.98%
AS16276	OVH SAS	530	5.22%
AS16509	Amazon.com	417	4.19%
AS14061	DigitalOcean	332	3.23%

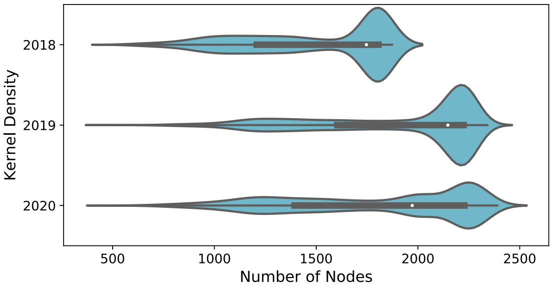


Fig. 9. Distribution of the synchronized nodes among the top three ASes in 2018, 2019, and 2020, respectively. Note that in 2019, the top three ASes hosted the maximum number of synchronized nodes (≈1917 on average).

attack. To isolate synced nodes, the attacker needs to have information about their ASes. To analyze that, we gathered information about synced nodes and their corresponding ASes and organizations. In Table VI, we enlist the top 5 ASes and organizations that hosted most synced nodes in Figure 8(a). We observed that 28% of synced nodes are hosted within the top 5 ASes. We plot their hosting pattern over a full day in Figure 8(b) and Figure 8(c). The cloud provider can spatially attack synced nodes by hijacking five ASes and temporally attack the remaining nodes.

Implications. Spatio-temporal attack is an optimized and targeted attack that provides multiple attack opportunities to a strong adversary to take down the network with minimal effort. As demonstrated by our results in Figure 8, at a given time, more than 50% of nodes can be behind the main blockchain and vulnerable to temporal attacks. Moreover, at the same time, the remaining synced nodes can be attacked by hijacking BGP prefixes of their hosting ASes and organizations. As a result, the adversary can split the network and orchestrate a mining race between two or more groups of miners and mine on the chain with a higher hash rate. The partitioning among the miners will allow the adversary to easily maintain the fork for six or more consecutive blocks. Moreover, since the adversary mines on the branch with a higher hash rate, the branch eventually becomes longer, allowing the adversary to successfully double-spend.

In our simulations §V-B, we modeled a block race for an adversary with 30% hash rate. However, we observed that with 30% hash rate, the adversary could only maintain the fork for a few blocks. In the spatio-temporal partitioning attack, the adversary can maintain the fork for a longer duration with the same hash rate and successfully double-spend.

1) Longitudinal Analysis of Spatio-Temporal Attacks: Similar to our analysis in §V-A1 and §V-B1, in this section, we analyze the changes in the spatio-temporal attack avenues over the last three years.

Note that to achieve the maximum impact of the spatio-temporal attack, the adversary must launch (1) the spatial attack against ASes that host the maximum number of synchronized nodes, and (2) the temporal attack against nodes that are behind the blockchain by one or more blocks. As a result, there will be a long-term partitioning (*i.e.,* delayed fork recovery) since the isolated synchronized nodes will not be able to help the victims of the temporal partitioning attacks. To analyze that, we selected the top three ASes that hosted the maximum number of Bitcoin nodes in each year. Among them, we counted the number of nodes that had an up-to-date blockchain. In Figure 9, we plot the kernel density estimation of the results obtained from each year. Our results show that since 2018, the concentration of synchronized nodes across top ASes has increased. In 2018, the mean and median number of synchronized nodes in the top three ASes were ≈1511 and ≈1747, respectively. In 2019, those numbers increased to ≈1917 and ≈2147, respectively. Finally, in 2020, the mean and median values were ≈1806 and ≈1972, respectively.

The spatio-temporal partitioning attack relies on (1) the distribution of synchronized nodes across the dominant ASes, and (2) the number of non-synchronized nodes in the rest of the network. Figure 9 shows that the distribution of synchronized nodes across dominant ASes has increased in 2019 and 2020. As a result, compared to 2018, the adversary can isolate more synchronized nodes by hijacking the same number of prefixes. Moreover, Figure 7 shows that the number of non-synchronized nodes has also increased in the last two years, making the temporal partitioning attack more optimal. Due to an increase in the feasibility of spatial partitioning attack on the synchronized nodes and the temporal partitioning attack on the non-synchronized nodes, we conclude that Bitcoin has become more vulnerable to the spatio-temporal partitioning attack.

D. Logical Partitioning

The Bitcoin network is actuated by communication among peers, each of which is a full node running software that conforms to a protocol. The protocol is defined by an open source software project, Bitcoin Core, initially published by Satoshi Nakamoto on January 9, 2009 [7]. Since 2009, there have been over 50 updates to Bitcoin Core, with each version provisioning new features or patching vulnerabilities in the previous version. It is therefore advisable for users to switch to the latest version for better security.

Table VII shows the top five Bitcoin Core clients used by the full nodes in 2018, 2019, and 2020. We note that less than 37% nodes use the latest Bitcoin Core version in each year. In 2018, 36.2% nodes used the latest version (0.16.0). In contrast, in 2019 and 2020, 15.8% and 16.1% nodes used the latest versions (0.18.1 and 0.19.0), respectively. A slow adoption of the latest version (demonstrated in Table VII) allows the adversary to exploit vulnerabilities in the previous versions (*i.e.,* CVE-2013-4627 [12]) and target the vulnerable nodes.

Attack Objectives. In the logical partitioning attack, the adversary's objective is to either to exploit the existing

TABLE VII

TOP 5 BITCOIN CORE VERSIONS USED IN 2018, 2019, AND 2020. IN 2019, THE LATEST VERSION WAS 0.18.1, USED BY 15.8% NODES (RANKED SECOND). IN 2020, THE LATEST VERSION WAS 0.19.0, USED BY 16.1% NODES (RANKED THIRD)

	Version	Usage		Version	Usage		Version	Usage
2018	0.16.0	36.2%	2019	0.18.0	35.4%	2020	0.18.0	26.3%
	0.15.1	27.5%		0.18.1	15.8%		0.18.1	24.4%
	0.15.0.1	5.0%		0.17.1	13.8%		0.19.0	16.1%
	0.14.2	4.6%		0.13.2	4.3%		0.17.1	7.8%
	0.15.0	2.5%		0.16.3	3.9%		0.16.3	2.7%

TABLE VIII

TOP 10 MOST COMMON VULNERABILITIES FOUND AMONG VULNERABLE BITCOIN NODES. THE FIRST COLUMN PRESENTS THE VULNERABILITY TYPE ACCORDING TO THE "COMMON VULNERABILITY EXPOSURES" (CVE) SYSTEM, AND THE SECOND COLUMN SHOWS THEIR DISTRIBUTION AMONG THE VULNERABLE NODES

Index	CVE	Distribution (%)
1	CVE-2018-15919	70.57%
2	CVE-2017-15906	4.82%
3	CVE-2016-10708	3.59%
4	CVE-2010-5107	2.62%
5	CVE-2010-4478	2.45%
6	CVE-2010-4755	2.45%
7	CVE-2012-0814	2.45%
8	CVE-2011-5000	2.45%
9	CVE-2007-4752	2.00%

vulnerabilities or release a new Core version to gain the confidence of full nodes. The newer version can have improved functionalities that can attract users and hidden vulnerabilities that can put them at risk.

Attack Procedure. Given the diversity in the usage of the Bitcoin Core, an adversary can exploit the vulnerable Bitcoin Core versions to isolate a group of nodes from the network. For instance, in 2018, a vulnerability (CVE-2018-17144) allowed the attackers to remotely shut down a Bitcoin node by sending a double-spend transaction. When the vulnerability was reported, all Bitcoin nodes were vulnerable to the attack. If the adversary shuts down a group of nodes owned by the mining pools, he can reduce the network hash rate.

In a more subtle scenario, the adversary can release a new Bitcoin Core client offering better performance and features (*i.e.,* enhanced GUI for wallets). Concurrently, the adversary can also add vulnerabilities in the client which can be exploited to remotely shutdown the node or force the node to follow different blockchain rules (*i.e.,* accept double-spend transactions). If a group of nodes starts following different consensus rules, those nodes will be partitioned from the rest of the network. Since the Bitcoin network is not controlled by any central authority, users are free to choose any new Bitcoin Core client. The adversary can exploit this *permissionless* nature of the network to advertise the enhanced features in his Bitcoin Core client and incentivize users to install it on their full nodes. As more users switch to the malicious software client, the network becomes more vulnerable to the logical partitioning attack.

Implications. Logical partitioning can be used to optimize attacks and take advantage of nodes in the crippled network. With each node valued at $o(10^7)$ USD, incentives exist to distribute and support software modifications, especially if not obviously malicious. Logical partitioning proceeds along several tracks: Bitcoin Core heterogenity and improvement proposals, independent developer versions, and publicly announced hard forks, such as Bitcoin Cash. These collide with spatial and temporal dimensions to create and optimize opportunities for other network attacks.

Logical Attacks: A Closer Look. Recently, we performed a new experiment to explore new attack avenues for the logical partitioning attack. We collected IP addresses of Bitcoin nodes and conducted a vulnerability scan on port 22. Our objective was to analyze software vulnerabilities in the full nodes that run on the cloud. As shown in §V-A, several full nodes are hosted on Amazon which is a cloud service provider. Typically, users access their cloud machines through SSH, and therefore, if there are vulnerabilities in their SSH protocol implementation, an attacker can exploit them access their Bitcoin wallets.

Our experiment showed that 24% of all Bitcoin nodes are vulnerable to at least one vulnerability. Among those vulnerable nodes, we found 25 unique vulnerabilities with "CVE-2018-15919" present among 70.57% nodes. "CVE-2018-15919" is an SSH vulnerability that allows an adversary to detect the existence of users on a target system [13]. We also observed that 4.82% nodes were vulnerable to "CVE-2017-15906" which allows an adversary to create zero-length files on the target machine. In Table VIII, we show the top 10 vulnerabilities found in the vulnerable Bitcoin nodes. More details about vulnerabilities in Table VIII can be found in [11]. From this analysis, we conclude that an insecure implementation of the SSH protocol on a Bitcoin node can lead to attacks.

AS Switching. It is possible that nodes that are hosted on cloud may change their IP address or even switch to another cloud operator. By changing the cloud operator, those nodes may move to a new AS. From the network standpoint, it is difficult to monitor such changes since a node's unique identity cannot be tied to its IP address. However, we were able to analyze that by mapping the node's SSH public key to its IP address. During the vulnerability scan, we collected the SSH public keys of the Bitcoin nodes and sampled the number of IP addresses that mapped to the same public key. Our assumption was that while the node might change the IP address, its public key will not change. Therefore, the mapping between the key and the IP address can be used to detect the nodes that changed their IP addresses.

From our dataset, we observed that 1,242 keys mapped to two or more IP addresses confirming that Bitcoin nodes indeed change their IP addresses. Among them, 155 keys mapped to more than two IP addresses, and the maximum number of addresses associated with a single key was 17. In Figure 10, we plot the number of nodes whose keys mapped to two or more IP addresses.

The next task was to determine if a node also changed its AS by switching to a different cloud operator. To analyze that, we sampled all the IP addresses associated with the same key and obtained the corresponding AS of those IP addresses. Our results showed that 37 nodes switched their ASes, with two nodes switching between four ASes. Upon a closer inspection,

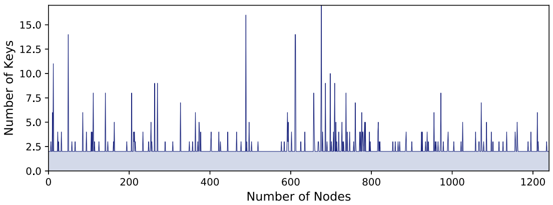


Fig. 10. The number of Bitcoin nodes whose SSH keys mapped to two or more IP addresses.

we observed that 12 nodes switched to AS6939 which is owned by an ISP called Hurricane Electric. We believe that instead of actually switching the AS, those nodes were using the IPv6 tunneling service provided by Hurricane Electric.

If an ISP only provides an IPv4 connection to a Bitcoin node, that node can only connect to IPv4 nodes. As such, to connect with IPv6 nodes, the node must either obtain an IPv6 address or use tunneling. Hurricane Electric is among the few ISPs that provide free IPv6 tunneling over the existing IPv4 connections. Using their service, an IPv4 node can easily connect to the IPv6 node. Therefore, it is possible that those 12 nodes were tunneling through AS6939.

E. Analysis Summary and Key Takeaways

Our analysis shows that the Bitcoin network can partitioned due to (1) the biased distribution of node across ASes, (2) weak network synchronization over a newly published block, and (3) vulnerable software implementations. Moreover, our supplementary analysis shows that in the last two years, the risk of spatial partitioning attacks has reduced since there is an increasing diversity in the node hosting patterns. On the other hand, the risk of temporal and spatio-temporal partitioning attacks has significantly increased since network synchronization has decreased and the distribution of the synchronized nodes among the top ASes has increased.

A key takeaway of our work is to explore the root causes for deteriorating network synchronization. In our preliminary work [34], we observed moments where an adversary could launch temporal partitioning attacks and create short-term forks in the network. However, our recent results cause a greater security concern. Decreasing network synchronization means that nodes are unable to receive new blocks on time. This could be due to increasing block propagation delay or weak network outdegree. Although, in §V-B1, we postulate that this delay could be due to the increasing number of Tor nodes. However, our analysis is not yet conclusive and requires further investigation. Note that decreasing network synchronization also increases the risk of majority attacks with less than 51% hash rate [14]. Since the current network condition is highly favorable for such an attack, it is therefore critical to identify the root causes for weak network synchronization.

VI. Countermeasures

To prevent spatial partitioning, mining pools should spread stratum servers across ASes to resist their centralization and raise the attack cost, since the attacker will have to hijack more BGP prefixes to isolate the targeted pool. Moreover, large Bitcoin exchanges, such as Coinbase and Bitstamp, should also host their full nodes across multiple ASes to prevent spatial attacks. In Bitcoin, spatial partitioning is a result of BGP hijacking. To counter that, Zhang *et al.* [40] propose reactive and proactive defense strategies that are based on the idea of "bogus route purging and valid route promotion" that can prevent BGP attacks on ASes across the Internet.

Temporal partitioning results from malicious peer behavior towards nodes that are behind the main chain. Although nodes can be behind due to various factors, the absence of a trusted central authority, makes them unaware of their condition. To counter the temporal partitioning attack without using a trusted party, we propose a simple and effective scheme called *BlockAware*, which is inspired by the *stale tip* detection mechanism in Bitcoin Core. In *BlockAware*, a node compares the timestamp of its latest block t_l and the current time t_c. Since the block time in Bitcoin is fixed at 600 seconds, if $t_c - t_l$ exceeds 600 seconds, the node can assume that it is behind the blockchain and vulnerable to the temporal partitioning attack. The node can then try new outgoing connections in different ASes to receive the block. Compared to the other existing approaches [16], *BlockAware* can be easily deployed in Bitcoin Core without significantly modifying the Bitcoin protocol.

Vulnerability to logical partitioning is due to the open network protocol. A central authority to regulate client participation would violate decentralization, a fundamental principle of Bitcoin. To remain the favored client, Bitcoin Core must continue to provide the best results for those who, typically without direct compensation, accept the responsibility of running a full node. In Bitcoin ecosystem, it would be reassuring for more than 36% nodes to run the most up-to-date version of Bitcoin Core. However, as diversity has long been known to enhance network security [27], we do not advocate enforcement mechanisms. Therefore, logical partitioning attacks remain a vulnerability to be considered.

VII. Related Work

Spatial Partitioning. The classical partitioning work is due to Apostolaki *et al.* [1] pointing out Bitcoin network centralization with respect to ASes, and highlighting the possibility of routing attacks with BGP prefixes. Some notable works related to spatial partitioning attacks include eclipse attacks [23], Bitcoin transaction graph analysis [33], and extracting intelligence from Bitcoin [22].

Blockchain Forks. Temporal and spatio-temporal partitioning result in a fork, forcing the affected nodes into following a different blockchain. As such, forks have been widely studied from the standpoint of regular nodes and miners. Decker and Wattenhofer [14] studied forks in the Bitcoin network and concluded that propagation delay is the major factor for them. The results in our experiments have validated their theory since delay is a major factor that causes some blocks to stay behind the main chain. Kwon *et al.* [24] introduced a new fork known as the Fork After Withholding (FAW), which guarantees more rewards than block withholding attacks. Eyal *et al.* [16] proposed a Byzantine fault tolerant protocol that addresses forks. Gervais demonstrated that double-spending is possible due to block tampering [20].

Consensus in Distributed Systems. Bano *et al.* [3] surveyed blockchain consensus protocols along with their strengths and limitations. In a similar vein, Mattila [28] analyzed blockchain consensus protocols and provided use cases for each scheme. Sun *et al.* [39] performed vulnerability analysis on distributed systems and proposed a trust evaluation framework to improve throughput and identify malicious peer behavior.

Related Attacks. Other notable attacks on blockchain applications include DDoS attacks, DNS attacks, selfish mining, the 51% attack, and blockchain ingestion [4], [38]. Li *et al.* [26], surveyed the security aspects of the blockchain by studying attacks on popular blockchain applications including Bitcoin, Ethereum, and Monero. Atzei *et al.* [2] performed analysis on vulnerabilities of smart contracts in Ethereum.

VIII. Conclusion

In this paper, we conduct a data-driven study to present four forms of partitioning attacks on the Bitcoin network namely spatial, temporal, spatio-temporal, and logical partitioning attacks. Our attacks are based on the biased distribution of nodes across ASes, non-uniform consensus over the blockchain, and diversity in the Bitcoin Core software usage. We validate our attacks with simulations and discuss the implication of each attack.

References

[1] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 375–392, doi: 10.1109/SP.2017.29.

[2] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts SoK," in *Proc. 6th Int. Conf. Princ. Secur. Trust*, vol. 10204, 2017, pp. 164–186. [Online]. Available: https://tinyurl.com/yd832abs.

[3] S. Bano *et al.*, "Consensus in the age of blockchains," 2017, *arXiv:1711.03936*. [Online]. Available: https://arxiv.org/abs/1711.03936

[4] K. Baqer, D. Y. Huang, D. McCoy, and N. Weaver, "Stressing out: Bitcoin stress testing," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Christ Church, Barbados: Springer, 2016, pp. 3–18. [Online]. Available: https://damonmccoy.com/papers/bitcoin16-final22.pdf

[5] Blockchain. (2018). *Hashrate Distribution*. [Online]. Available: https://blockchain.info/pools

[6] BTC. (2018). *BTC.Com Stratum Address*. [Online]. Available: https://pool.btc.com/helpCenter?id=miner

[7] B. Community. (2018). *Bitcoin Core Version History*. [Online]. Available: https://bitcoin.org/en/version-history

[8] B. Community. (2018). *Stratum Mining Protocol*. [Online]. Available: https://en.bitcoin.it/wiki/Stratum_mining_protocol

[9] B. Community. (2018). *Bitnodes: Global Bitcoin Nodes Distribution*. [Online]. Available: https://bitnodes.earn.com/

[10] E. Community. (2018). *Earn: Earn Money by Answering Messages and Completing Tasks*. [Online]. Available: https://earn.com

[11] N. Community. *National Vulnerability Database*. Accessed: Oct. 25, 2020. [Online]. Available: https://tinyurl.com/y9guktjx

[12] CVE. (2018). *Vulnerability Details: Cve-2017-9230*. [Online]. Available: https://www.cvedetails.com/cve/CVE-2017-9230/

[13] CVEDetails. *Vulnerability Details: Cve-2018-15919*. Accessed: Oct. 25, 2020. [Online]. Available: https://www.cvedetails.com/cve/CVE-2018-15919/

[14] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proc. IEEE Peer-to-Peer*, Sep. 2013, pp. 1–10, doi: 10.1109/P2P.2013.6688704.

[15] P. Dhungel, M. Steiner, I. Rimac, V. Hilt, and K. W. Ross, "Waiting for anonymity: Understanding delays in the tor overlay," in *Proc. IEEE 10th Int. Conf. Peer-to-Peer Comput. (P2P)*, Aug. 2010, pp. 1–4, doi: 10.1109/P2P.2010.5569995.

[16] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse. (Mar. 2016). *Bitcoin-NG: A Scalable Blockchain Protocol*. [Online]. Available: https://tinyurl.com/y7gxcdgr

[17] M. Fadhil, G. Owenson, and M. Adda, "Locality based approach to improve propagation delay on the bitcoin peer-to-peer network," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 556–559, doi: 10.23919/INM.2017.7987328.

[18] G. C. Fanti and P. Viswanath, "Deanonymization in the bitcoin P2P network," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1364–1373. [Online]. Available: https://tinyurl.com/y72zgvtk

[19] A. Gervais, S. Capkun, G. O. Karame, and D. Gruber, "On the privacy provisions of Bloom filters in lightweight bitcoin clients," in *Proc. Comput. Secur. Appl. Conf.*, C. N. P. Jr., A. Hahn, K. R. B. Butler, and M. Sherr, Eds., 2014, pp. 326–335, doi: 10.1145/2664243.2664267.

[20] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 692–705, doi: 10.1145/2810103.2813655.

[21] A. Greenberg. *Hacker Redirects Traffic From 19 Internet Providers to Steal Bitcoins*. Jun. 2017. [Online]. Available: https://www.wired.com/2014/08/isp-bitcoin-theft/

[22] E. Heilman, L. AlShenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "TumbleBit: An untrusted bitcoin-compatible anonymous payment hub," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2017. [Online]. Available: https://bit.ly/3btu2s8

[23] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *Proc. USENIX Secur. Symp.*, 2015, pp. 129–144. [Online]. Available: https://bit.ly/3vgJEXM

[24] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (FAW) attacks on bitcoin," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 195–209, doi: 10.1145/3133956.3134019.

[25] L. Latinov. (2020). *MD5, SHA-1, SHA-256 and SHA-512 Speed Performance—Automation Rhapsody*. [Online]. Available: https://bit.ly/3kXEF9J

[26] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," 2018, *arXiv:1802.06993*. [Online]. Available: http://arxiv.org/abs/1802.06993

[27] B. Littlewood and L. Strigini, "Redundancy and diversity in security," in *Proc. Eur. Symp. Res. Comput. Secur.*, vol. 2004, pp. 423–438, doi: 10.1007/978-3-540-30108-0_26.

[28] J. Mattila, "The blockchain phenomenon—The disruptive potential of distributed consensus architectures," Univ. California, Berkley, Berkley, CA, USA, Tech. Rep. 38, 2016.

[29] A. Mohaisen and K. Ren, "Leakage of .onion at the DNS root: Measurements, causes, and countermeasures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 3059–3072, Oct. 2017, doi: 10.1109/TNET.2017.2717965.

[30] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Advances in Cryptology*. Paris, France: Springer, 2017, pp. 643–673, doi: 10.1007/978-3-319-56614-6_22.

[31] B. Reward. (2018). *Bitcoin Block Reward Halving Countdown*. [Online]. Available: https://www.bitcoinblockhalf.com/

[32] (2018). *Autonomous Systems in the World*. [Online]. Available: https://tinyurl.com/yaz73jnb

[33] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2013, pp. 6–24. [Online]. Available: https://bit.ly/3c7NGc8

[34] M. Saad, V. Cook, L. Nguyen, M. T. Thai, and A. Mohaisen, "Partitioning attacks on bitcoin: Colliding space, time, and logic," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1175–1187, doi: 10.1109/ICDCS.2019.00119.

[35] SaadCM19. *Beingmsaad/-Partitioning-Attacks-Ton*. Accessed: Oct. 25, 2020. [Online]. Available: https://github.com/beingmsaad/-Partitioning-Attacks-ToN.git

[36] SaadCM19. *Beingmsaad/-Partitioning-Attacks-Ton*. Accessed: Oct. 25, 2020. [Online]. Available: https://github.com/beingmsaad/-Partitioning-Attacks-ToN.git

[37] M. F. Sallal, G. Owenson, and M. Adda, "Proximity awareness approach to enhance propagation delay on the bitcoin peer-to-peer network," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, doi: 10.1109/ICDCS.2017.53.

[38] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security*. Springer, 2016, doi: 10.1007/978-3-662-54970-4_30.

[39] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, "A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks," in *Proc. Int. Conf. Comput. Commun.*, 2006, pp. 1–13. [Online]. Available: https://tinyurl.com/5efwzumk

[40] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao, "Practical defenses against BGP prefix hijacking," in *Proc. ACM Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, New York, NY, USA, Dec. 2007, p. 3, doi: 10.1145/1364654.1364658.