Partitioning Attacks on Bitcoin: Colliding Space, Time, and Logic

Muhammad Saad*, Victor Cook*, Lan Nguyen†, My T. Thai†, and Aziz Mohaisen*
*University of Central Florida † University of Florida
{saad.ucf, victor.cook}@knights.ucf.edu, lan.nguyen@ufl.edu, mythai@cise.ufl.edu, mohaisen@cs.ucf.edu

***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 by simulating attack scenarios and the implications of each attack on the Bitcoin network. We conclude with suggested countermeasures.**

I. Introduction

Blockchain is a new paradigm for distributed computing, with Bitcoin being its most popular application [41], [48]. Due to its high market share of over $110 billion USD [10], Bitcoin has been a lucrative target of attack for adversaries, who have been mainly targeting Bitcoin's exchanges, the blockchain fabric, and nodes involved in Bitcoin's network.

In this paper, we analyze the peer-to-peer model of cryptocurrencies and associated security. 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 [29], as an instance of the spatial attack. This attack has been formalized and examined in [3]. 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 [3]. In our analysis, started on February 28, 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, causing a "hard fork" by controlling a limited number of ASes. 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 more than 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 attack avenues.

Unique to our study, we exploit the non-uniform consensus among peers for optimized temporal attacks (§V-B). We observed that—due to latency and malicious peer behavior—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.* [3] performed a data analysis on Bitcoin to understand AS-level 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 by outlining characteristics, distribution, location, and performance of full nodes. 2) Embracing various characteristics of the network, we propose several directions of attacks and validate them through data analysis and simulations. We outline, demonstrate, model, optimize, and evaluate spatial, temporal, spatio-temporal, and logical attacks. 3) We discuss possible countermeasures to address those attacks. Through the rest of the paper, in §II, we outline the Bitcoin network model, and in §III, we outline the threat model and adversarial capabilities. 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.

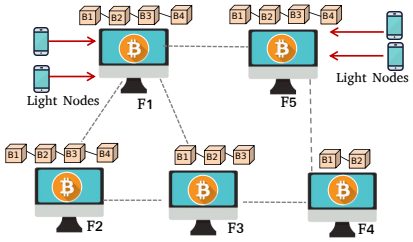


Fig. 1. Bitcoin illustration with full nodes and lightweight nodes. Lightweight nodes only have the view that their associated full nodes provide. Full nodes F1, F2, and F5 have updated views while F3 and F4 are 1-2 blocks behind.

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 are responsible for extending the blockchain by creating new blocks [44].

Ideally, all the participating nodes in the network need to have an updated copy of the blockchain, but the growing size of the chain makes it infeasible to be used on smart devices. For example, the current blockchain size in Bitcoin is approximately 150GB [52], and if a user wants to use Bitcoin's services on his smart phone, he might not be able to download the complete blockchain and become part of the network. To address this problem, third party services such as *Blockchain.info* [33] provide an easy access to such clients by downloading Blockchain and providing access to smart device users. *Blockchain.info* maintains an active node in Bitcoin that keeps track of all transactions and blocks, and replicates the network view to all of its customers. 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 nodes and the lightweight nodes, we refer the reader to [26].

III. Threat Model

In this section, we outline the basics of partitioning attacks on Bitcoin and describe our threat model. Through data-driven analyses, we establish the *modus operandi* of the Bitcoin network, and describe capabilities needed by the adversary to partition the network spatially and temporally. Towards that, we revisit Apostolaki *et al.*'s work [3] (referred to as the "classical attack"), providing a baseline for partitioning attacks. We highlight new targeted attacks on the network, by introducing temporal, spatio-temporal, and logical partitioning 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. Finally, for logical attacks, we assume the adversary to be a software developer capable of exploiting bugs in the Bitcoin software client. 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. As such, a nation-state can partition the network by blocking the flow of traffic through its ASes and organizations. Countries such as Bolivia, Kyrgyzstan, and Nepal have permanently banned Bitcoin and its exchanges [54]. If China, for example, decides to ban Bitcoin, it will have a significant impact on the health of the Bitcoin network since 60% of the mining traffic goes through China (as shown in Table IV).

Adversarial View. We assume that the adversary has a consistent view of the network similar to the one available to us for conducting our analysis. The adversary will have access to the following information. 1) The top ASes and organizations that host a maximum number of nodes and their distribution over time. 2) The temporal spread of block information among all nodes in the network upon block broadcast. 3) The vulnerable nodes in the network based on their location, uptime, latency, consensus time, and neighboring peers. 4) The vulnerable network entities (ASes and organizations) based on their public information such as BGP prefixes, neighboring ASes, location, and routing information.

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 full nodes, lightweight nodes, and mining pools.

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 in Bitcoin* [15]. This can be exploited by the malicious mining pool to identify vulnerable nodes that are 1–5 blocks behind. A malicious miner, for instance, can mislead those nodes by propagating false information in the network. Doing so may create a temporary or even a permanent partitioning in the network, where a group of nodes are misled into following a counterfeit blockchain.

IV. Preliminary Analysis

A. Data Collection

For our analysis, we crawled data from Bitnodes [15], which is a Bitcoin service supported by *Earn.com* [16]. Bitnodes maintains a persistent connection with all reachable nodes by running a full node that connects to the rest of the network. After connecting with all nodes, Bitnodes uses inventory message (*inv*) and data messages (*getdata, getblock, gettransaction*) to get recent blocks and transactions from each node (for more information regarding these protocol messages, we refer the reader to the Bitcoin protocol documentation [13]). For each node, Bitnodes records the response time to calculate 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 a node.

We used the information provided by Bitnodes to develop another crawler, atop Bitnodes, to acquire data and store it in

our local database. We ran the crawler on our campus server for two months, and our complete dataset spans two months of Bitcoin network information with an aggregate size of 80GB. In summary, we were able to collect the Bitcoin network information sampled at every 10 minutes to analyze consensus distribution after each published block, and at every 1 minute to observe consensus pruning in the network in-between the publication of two successive blocks.

B. Methodology

In our initial experiments, first we cross-validated the information provided by Bitnodes. We mapped the crawled IP addresses to a commercial-grade geo-mapping dataset obtained from Digital Envoy (DE) [42]. The DE dataset mapping of Bitnodes IP addresses validated the information in our dataset regarding ASes and organizations. After establishing data reliability, we performed a series of experiments to analyze the configuration of the network, and 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 consensus distribution among nodes, based on their view of the blockchain. 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 Figure 1, nodes F3 and F4 are 1-2 blocks behind the main chain. Therefore, they provide an outdated view of network to their lightweight nodes. This information can be used by the attacker to lure them into a counterfeit network by feeding them bogus blocks or a different blockchain. We leveraged this information to outline temporal partitioning attacks that can be launched on Bitcoin network to isolate nodes based on their outdated view. Our results showed that dynamics of Bitcoin network are not consistent over time and there are vulnerable spots for an attacker who can connect to a group of nodes and partition them.

Experiments and Simulations. We modelled and simulated partitioning attacks on Bitcoin based on the data, the network view, and adversarial capabilities. Our simulations accurately reproduced the vulnerable state of the network that was observed in our data analysis. By causing non-targeted communication errors, forks were created that resembled those occurring naturally when the network is not synchronized. Bitcoin forks have been observed up to a height of 13, and can enable double-spending [35]. As in the real network, the simulator resolved forks within two or three block intervals, with all nodes joining the longest chain. The simulation showed that partitioning attacks can create and exploit such forks using targeted communication disruption, holding them open long enough to achieve attack objectives.

C. Measurements and Observations

Below, we discuss some key observations we made during the preliminary analysis on the Bitcoin network on February 28, 2018. We show the number of full nodes in the network and their distribution with respect to IP addresses, link speed, latency, and block index.

The network snapshot showed that there were 13,635 full nodes in the Bitcoin network. This shows that the size of the actual network is small compared to SPV clients, considering that *Blockchain.info* alone hosts 2.3–5 million users [32]. At the time of data collection, 11,382 (83.47%) nodes were up while 2,253 (16.52%) nodes were down. 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. We also make use of peer information maintained by Bitnodes to characterize certain properties of nodes, including the latency index, the uptime index, and the block index. Each of these indicators can be used to profile the given node in the network.

TABLE I
OVERVIEW NODE CHARACTERISTICS OBSERVED ON FEB 28, 2018. NOTE THAT THE IPV4 AND IPV6 NODES ARE SIMILAR IN LINK SPEED (MBPS), LATENCY AND UPTIME INDEX, WHILE TOR NODES HAVE MUCH HIGHER LINK SPEED AND LOW LATENCY.

		Link Speed		**Latency Index**		**Uptime Index**	
Type	Count	μ	σ	μ	σ	μ	σ
IPv4	12,737	25.04	258.80	0.70	0.45	0.68	0.44
IPv6	579	23.06	245.36	0.86	0.35	0.67	0.42
TOR	319	432.67	1046.5	0.24	0.25	0.76	0.37

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, meaning that they were using TOR services to run Bitcoin. The average link speed of the IPv4 and IPv6 was 25.04 Mbps and 23.06 Mbps, respectively. Their latency index, block index, and uptime index were also similar to one another. On the other hand, TOR nodes had a high average links speed of 432.67 Mbps; approximately 17 times higher than the average link speed of IPv4 and IPv6 nodes, respectively. Consequently, they also had low latency and higher uptime index. We report our findings from preliminary analysis in Table I.

V. PARTITIONING ATTACKS ON BITCOIN

Based on our preliminary analysis, we propose four types of partitioning attacks that can be launched on the Bitcoin network. The fundamental premise of each attack is related to the spatial positioning of nodes, the topological symmetry of the network, the temporal consensus over the blockchain state, or the client side software used by nodes to run Bitcoin. We define these attacks as spatial, temporal, spatio-temporal, and logical partitioning attacks, respectively.

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.

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 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

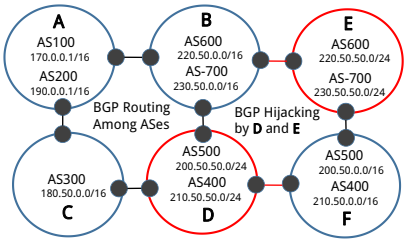


Fig. 2. Network topology consisting of organizations, ASes and full nodes. Organizations D and E can launch BGP attacks against F and B respectively.

TABLE II
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	# of Nodes	Total Nodes %	Organizations	# of Nodes	Total Nodes %
AS24940	1,030	7.54%	Hetzner Online GmbH	1,030	7.54%
AS16276	697	5.11%	Amazon.com, Inc	756	5.54%
AS37963	640	4.69%	OVH SAS	700	5.13%
AS16509	609	4.47%	Hangzhou Alibaba	640	4.69%
AS14061	460	3.37%	DigitalOcean, LLC	503	3.69%
AS7922	414	3.04%	Comcast Communication	414	3.04%
AS4134	394	2.89%	No.31, Jin-rong Street	394	2.89%
TOR	319	2.34%	TOR	319	2.34%
AS51167	288	2.11%	Contabo GmbH	288	2.11%
AS45102	279	2.05%	Alibaba (China)	279	2.05%

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 [3], 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 [45], 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.

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 II, 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.

As outlined in Figure 3, 50% of the Bitcoin network is hosted by 21 organizations and 24 ASes, respectively. Moreover, 30% of the traffic is hosted by 8 organizations and ASes, respectively. Prior work [3] done in 2017, showed that

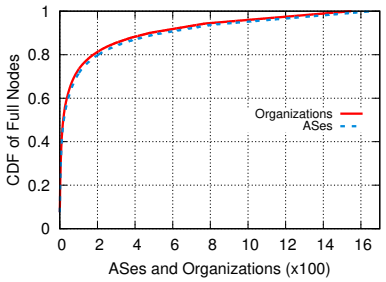


Fig. 3. CDF of the Bitcoin full nodes in ASes and organizations.

TABLE III
DISTRIBUTION OF BITCOIN FULL NODES OVER TIME.

	2017	2018	Change %
ASes with 50% nodes	50	24	52%
ASes with 30% nodes	13	8	38%

50% of the network was hosted by 50 ASes and 30% of the network was hosted by 13 ASes. To understand the change in the network, let N_1 be the number of nodes comprising $p\%$ of the network in 2017. Let N_2 be nodes comprising the same $p\%$ of the traffic in 2018. We define the change in the centralization of the network as $C = \frac{(N1-N2)\times 100}{N1}$, and provide the results of change in Table III. Notice that over one year, 50% nodes have been centralized by a factor of 52%. The prior work did not look into the distribution of network with respect to organizations, so we do not have a baseline for comparison. Although, it can be observed from our data and plots, that full nodes are more concentrated at the organization level.

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" [14]. 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* [8]; results are reported in Table IV. 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 [9], [2], [22]. 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 IV. 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.

Attack Validation. In this section, we will validate our observations and hypothesis regarding BGP hijacking on Bitcoin ASes and organizations. BGP routing attacks on Internet happen frequently. In 2008, a service provider from Pakistan hijacked Youtube traffic by announcing more specific BGP prefixes than the ones announced by Youtube [28]. Similarly,

TABLE IV
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%	—	—

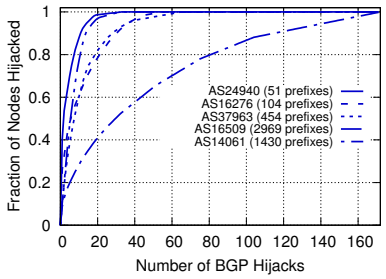


Fig. 4. CDF of top 5 ASes vulnerable to BGP attacks. The key shows total BGP prefixes announced by AS. For 8 ASes, 80% nodes can be isolated by hijacking 20 BGP prefixes.

in 2014, a Canadian ISP hijacked prefixes of 19 organizations hosting Bitcoin traffic including Amazon, OVH, Digital Ocean, LeaseWeb, and Alibaba [29]. In 2017 alone, 14,000 BGP attacks were launched against major ASes [46].

To validate the attack and its impact, we selected the top 5 ASes from Table II, and enumerated the IP addresses of full nodes hosted by these ASes. Next, we grouped the IP addresses based on the BGP prefixes announced by each AS. We then calculated the number of BGP prefixes required to isolate a percentage of full nodes hosted by the AS. As a result, a group of full nodes sharing the same BGP prefix can all be compromised if the BGP prefix is hijacked. We report our findings in Figure 4, where we show that except for AS16509, 95% of full nodes in all other ASes are vulnerable, once fewer than 40 BGP prefixes are hijacked. AS24940, which hosts 1,030 nodes can be compromised by hijacking only 15 BGP prefixes, while it takes more than 140 BGP prefixes to compromise AS16509, which hosts 609 nodes. Taking the number of isolated nodes as an advantage and the number of prefixes to be hijacked as an effort, AS24940 will be more costly with smaller advantage than AS16509.

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 IV, 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.

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.

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. In Figure 5, we provide an abstraction of the temporal attack that exploits the vulnerable nodes, and introduces a soft fork in the network.

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 [19]. 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 [49], [23], 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 6. In Figure 6, the x-axis denotes a time-index for network observations (one observation every 10 minutes in Figure 6(a) and Figure 6(b), and one every minute in Figure 6(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 6(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 most 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

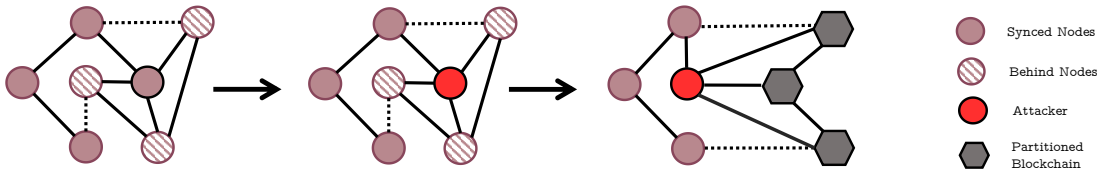
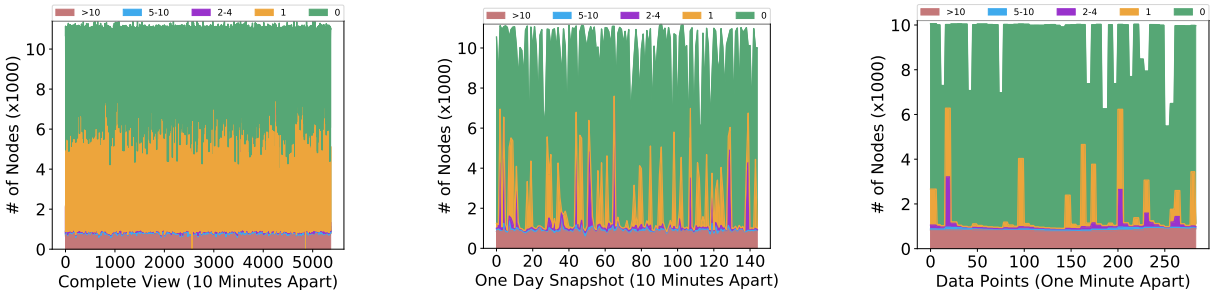


Fig. 5. An illustration of the temporal attack. The attacker establishes connections with nodes and identifies vulnerable nodes that have an outdated view. Vulnerable nodes have have not been provided new blocks by surrounding peers, which shows their weak relationship/connectivity. We annotate this weak relationship with dotted lines. The attacker feeds his copy of blocks to vulnerable nodes, thereby partitioning the network into two conflicting chains.



(a) General trend of the network (b) One day snapshot (c) Consensus between block propagation

Fig. 6. Temporal consensus in Bitcoin network. 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 6(a) shows the overall network, Figure 6(b), shows a day (March 25) that offers greater attack opportunity, and Figure 6(c) shows consensus pruning during 10 minutes.

attacker, with higher granularity, there is a better vantage point to attack a group of nodes. Focusing on a single day shown in Figure 6(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 6(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 6(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 6(c), the width of nodes that are behind show the attack time window while the height represents the number of vulnerable nodes.

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* $\mathtt{L}(t)$ *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.

TABLE V
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%)
40	1040(10.37%)	984(9.82%)	940(9.38%)
70	1036(10.34%)	976(9.74%)	929(9.27%)
200	908(9.08%)	887(8.82%)	821(8.16%)

We identify nodes whose historical behaviors show their vulnerability to temporal attacks, and record their results in Table V. 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. As such, the normalization parameter, represented by the total number of nodes at that time, may change, which results in an increasing percentage for a decreasing number of nodes in Table V. For instance, for 6,280 nodes, the total number of nodes was 10,020, which is about 62.67%. On the other hand, for 908 nodes, the total number of nodes was 10,000 which approximates to 9.08%. We tested with a variety of timing constraints T and present the results that best suit the attacker. The first column shows different T values, the second/third/forth columns show the

maximum number of nodes that lag behind main chain for at least 1/2/5 blocks respectively. The decreasing of maximum number of nodes, along with the increasing of timing constraint, shows the fact that the longer time it takes to implement an attack, the fewer choice of vulnerable nodes is available. We noticed that there were moments in which a majority of nodes in the network ($\geq 50\%$) was at least 1 block behind for more than 5 minutes, and up to 20% nodes lagged behind the main chain for more than 15 minutes.

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 *et al.* [24]. Using that, the timing of the attacker to connect to a node is:

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

where $f(\cdot), F(\cdot)$ are probability density and cumulative distribution functions. Given timing assigned to isolate m nodes is $\mathcal{T} = (t_1, ...t_m)$. 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 $x_1, x_2, ...x_n$ 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 $x_1 = x_2 = ... = x_n$

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 $\mathcal{T}$. 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 VI. 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 VI, there can be 1,761 vulnerable nodes within $\mathtt{T} = 10$ minutes). Therefore, we conclude that Bitcoin is highly vulnerable to temporal attacks.

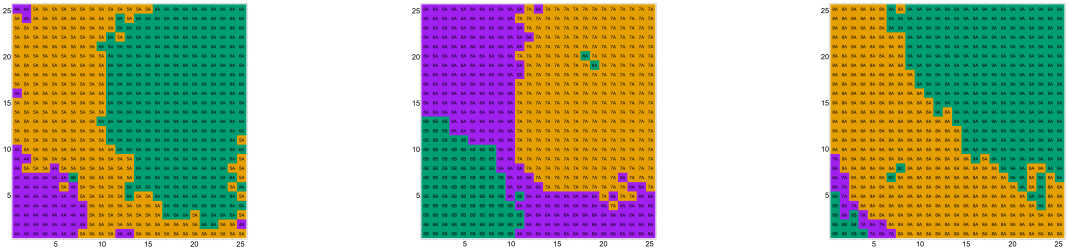
TABLE VI
MINIMUM TIMING CONSTRAINT $\mathtt{T}$ (SECONDS) TO ISOLATE m NODES UNDER THE GIVEN RATE λ.

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
0.8	119	354	589	942	1177	1412	1765
0.9	116	346	575	920	1149	1379	1723

Simulation and Attack Validation. To validate the insights obtained from our data and theoretical analysis, we developed a simulation model in R to test temporal attacks. The simulator was tested in base simulation scenarios, such as zero and perfect communication among nodes. As an internal error check, and to make the simulation more realistic, each simulated node maintains a 64-bit MD5 hash linked chain of values updated to its current fork. By adjusting parameters, the simulation was capable of accurately representing the state of the network as we observed in our dataset.

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 [23]. Our experimental data confirmed this distribution. Following this, the peers were evenly distributed in terms of communication errors and latency. Peer communication failure rate is represented by a model parameter, typically around 10 percent failures. The latency is represented by the number of communication time steps per simulation block. This is scaled according to the simulation size. Each time step represents one peer-to-peer communication attempt for each node.

The simulation was used to model information flow through the network under different attack scenarios. A network of 10,000 nodes can be simulated using a square grid of size 100. We ran simulations using the entire network. For clarity a grid of size 25 (1/16 of the active nodes) is shown in the figures. This grid ran faster, is easier to read, and well simulated experimental results. Using different scaled network simulations we discovered that the upper limit of Decker and Wattenhofer's node propagation delay T_{delay} can be expressed as a ratio of the block interval divided by the network diameter. Taking the inverse of this ratio we arrive at a non-dimensional parameter, the *span ratio* representing how many times information can travel from one side of the Bitcoin network to the other during the block interval. Assuming a square grid, network diameter is proportional to the square root of the number of nodes. A given span ratio R_{span} with the Bitcoin block interval T_{block} thus yields a maximum propagation delay to maintain the state of a network of N nodes: $T_{delay} = T_{block}/(R_{span} * N^{0.5})$. As the Bitcoin network grows, a smaller propagation delay is required to synchronize peers. Specifically, T_{delay} is inversely related to the square root of the number of nodes. The maximum value of R_{span} in simulation was 2.0, corresponding to a 3 second interval per peer communication in the actual network of 10,000 nodes. With reasonable values for the communication failure, such a small time step resulted in a network that was fully updated between blocks. Therefore, $R_{span} = 2.0$ is a good target for blockchain synchronization.



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

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

Figure 7 shows a sample of results obtained from simulation, 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 6.

Implications. Even a short term fork in the network would cause sufficient disruption to invalidate transactions. Such an attack is likely to result in significant loss to network stakeholders. Quantifying the impact of adverse events on Bitcoin has been inconclusive [25][20], and is dependent upon user perception [43]. However, once the targeted nodes are isolated, as shown in Figure 5, the soft fork will create a temporary partition in the network. The isolated nodes will be following a counterfeit blockchain with different transactions from the main chain. Therefore, when nodes recover from the fork, the attacker's blocks will be rejected, and all transactions belonging to legitimate users in those blocks will also be reversed. This will require a major update on the set of all UTXO's at each node, and a system-wide check on the transactions being reversed. Standing out in our analysis is the observation that Bitcoin has a level of asymmetric vulnerability. With a market capitalization of $o(10^{11})$ USD and network configuration of $o(10^4)$ nodes, each full node is worth $o(10^7)$ USD. However, the cost of disrupting the network is far less than the value being impacted, which makes Bitcoin an economically attractive target for temporal 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.

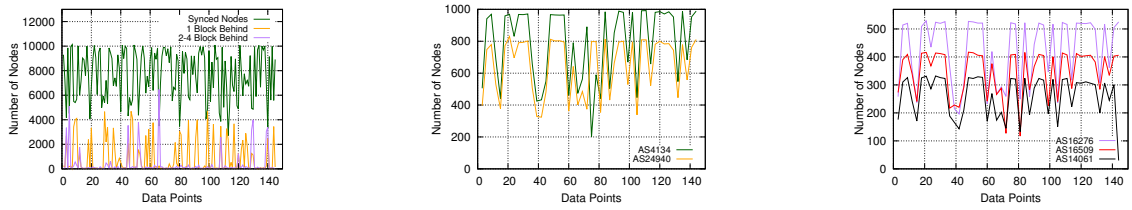
Attack Objectives. In this attack, the aim of the adversary is to split the network based on the network's vulnerability to both the spatial and temporal partitioning. As shown in Figure 6(a) and Figure 6(b), the purple and yellow nodes are vulnerable to temporal attacks. However, the attacker cannot 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 will need information about the 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 6(b) in Figure 8(a). The green line indicates the number of 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 capabilities of an attacker.

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 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 VII, we enlist the top 5 ASes and organizations that hosted most synced nodes in Figure 8(a).



(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 6(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). On average, AS4134 hosts most of the nodes.

TABLE VII
TOP 5 ASES THAT HOSTED ALL THE SYNCHRONIZED NODES IN FIGURE 6(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%

TABLE VIII
TOP 5 SOFTWARE VERSIONS USED BY BITCOIN FULL NODES ALONG WITH THEIR RELEASE DATE, LAG FROM THE DATE OF COLLECTION IN DAYS, AND PERCENTAGE OF USERS.

Index	Version	Release Date	Lag	Users %
1	B. Core v0.16.0	02-26-2018	59	36.28%
2	B. Core v0.15.1	11-11-2017	166	27.52%
3	B. Core v0.15.0.1	09-19-2017	219	5.01%
4	B. Core v0.14.2	06-17-2017	313	4.67%
5	B. Core v0.15.0	04-22-2017	369	2.05%

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. The attacker can select a suitable trade-off between the lagging nodes and synced nodes, based on the attacker's capabilities, and disrupt the network. For a successful attack on synced nodes, the attacker may just have to isolate a small number of nodes that relay blocks to each other, and due to the cascade effect, remaining nodes will eventually collapse. As such, if the number of full nodes is small in a cryptocurrency such as Bitcoin Cash or Litecoin, the attacker can compromise the entire cryptocurrency by affecting the flow of valuable data including transactions and blocks.

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 [12]. Since 2009, there have been over 40 updates to Bitcoin Core, with the latest, v0.16.0 released in February 2018. New versions build upon previous ones with improved security, performance and functionality. Since the Bitcoin network is open to any client that satisfies the network protocol, peers can run modified software. Optional features such as SegWit [1] are implemented in this way, compatible with Bitcoin Core.

Table VIII shows the distribution of Bitcoin software at the time of our data collection, along with their release date and percentage of users. We observed that 288 Bitcoin software variants are used by full nodes. The latest version of Bitcoin Core, 0.16.0, is used by only 36% of the nodes while 27% use version 0.15.1. The remaining 37% of the network uses 286 different software clients.

Attack Objectives. The objective of the attacker would be to gain the confidence of full nodes. Changes may be subtle and not perceived as threats. Diverse incentives may be employed for adoption. In our scenario, the attacker's influence over the software would be sufficient to optimize and magnify the effects of the attack.

Attack Procedure. Peer "democracy" in software selection has served well, but is vulnerable to attacks. Over time, a modified software variant might gain popularity by offering better performance and features. One example is Falcon, a custom Bitcoin client run by 10 nodes. Falcon provides faster connectivity and minimum delay during transaction propagation [55]. Falcon is not malicious, but it demonstrates the independence of peers to run a client that is not part of Bitcoin Core. A hypothetical client that economizes the cost of running a full node might gain general acceptance, while at the same time reducing the cost of controlling a significant portion of the network.

In a more subtle scenario, a malicious entity with cooperating peers could modify the Bitcoin Core software after download. The modifications may be surreptitious or proclaimed to be enhancements. Nodes influenced by the attacker would seem normal, but would be used to facilitate an attack. A simple example of permissible client modification would be to increase the number of peer connections [11], and help the spread of malicious blocks.

In either case, the software would provide a platform to enhance the partitioning attack. During the attack, modified clients could steal bitcoins from connected wallets, isolate peers from the network, propagate false information in the network, and cause DoS attacks on neighboring peers. To further analyze vulnerabilities associated with Bitcoin software clients, we mapped known client versions to the National Vulnerability Database (NVD). From NVD, we obtained 36 reported vulnerabilities along with the vulnerability ID, the publishing date, and the CVSS severity. For instance, a vulnerability with ID CVE-2018-17144, shows that Bitcoin clients are vulnerable to a remote denial-of-service attack via

duplicate inputs. This vulnerability can be found in all client versions, which puts the entire network at risk. Some other notable vulnerabilities reported in NVD are CVE-2017-9230, CVE-2013-5700, and CVE-2013-4627 [18]. For more details, we refer the reader to [17].

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.

VI. Countermeasures

To prevent spatial partitioning, mining pools should spread stratum servers across various ASes. This can resist the centralization of stratum servers and raise the attack cost, since the attacker will have to hijack more BGP prefixes to isolate the targeted pool. Furthermore, 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 an artifact of BGP hijacking and to counter that, Zhang *et al.* [55] 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 that, we propose a simple yet effective scheme, called *BlockAware*, which uses the expected block time to notify the node about its blockchain view with respect to the network. 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, a difference between the two values exceeding 600 seconds ($t_c - t_l > 600$) indicates a node has not received the latest block. In such a situation, the node can try to connect to other nodes, and query them for the latest block. As part of our ongoing work, we are prototyping *BlockAware* over Bitcoin Core to defend against the temporal attacks.

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 [39], we do not advocate enforcement mechanisms. Therefore, logical partitioning attacks remain a vulnerability to be considered.

VII. Related Work

Spatial Partitioning. The classic study on partitioning attacks was carried out by Apostolaki *et al.* [3] based on the centralization of Bitcoin network with respect to ASes, and highlighting the possibility of routing attacks with BGP prefixes. Some notable work on the attack surface includes eclipse attacks [30], double-spending [34], Bitcoin transaction graph analysis [47], anonymity in Bitcoin peer-to-peer model [36], and extracting intelligence from Bitcoin [51], [31].

Blockchain Forks. Temporal and spatio-temporal partitioning on the blockchain result in a fork that leads the affected nodes into following a different blockchain. As such, forks have been widely studied in the community from the standpoint of regular nodes and miners. Decker and Wattenhofer [19] studied the occurrence of forks in the Bitcoin network. They concluded that propagation delay is the major factor that might result in a fork. The results in our experiments have validated their theory since delay is the major factor that causes some blocks to stay behind the main chain. Kwon *et al.* [37] introduced a new form of blockchain fork known as the Fork After Withholding (FAW) attack which guarantees more rewards than block withholding attacks. Eyal *et al.* [21] proposed a Byzantine fault tolerant blockchain protocol that addresses the problems of forks. Gervais demonstrated that double-spending is possible due to block tampering [27].

Consensus in Distributed Systems. In a blockchain, consensus about the state of the system is achieved with a consensus protocol. Bano *et al.* [5] surveyed blockchain consensus protocols along with their strengths and limitations. In a similar vein, Juri Mattila [40] analyzed blockchain consensus protocols and provided use cases for each scheme. Sun *et al.* [53] 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 [50], [7], [6]. Li *et al.* [38], surveyed the security aspects of the blockchain by studying attacks on popular blockchain applications including Bitcoin, Ethereum, and Monero. Atzei *et al.* [4] performed analysis on vulnerabilities of smart contracts in Ethereum.

VIII. Conclusion

We examine various partitioning attacks on blockchain-based cryptocurrencies. We demonstrate that the Bitcoin network is becoming increasingly centralized at the AS-level, making it more vulnerable to spatial partitioning. Data collection and analysis demonstrate that consensus pruning of the Bitcoin network is non-uniform, presenting optimizable opportunities for an attacker to fork the network by segregating vulnerable nodes. We study four forms of partitioning attack: spatial, temporal, spatio-temporal, and logical. We validate our attacks with simulations and discuss the implication of each attack. Finally, we present possible countermeasures to those attacks. To the best of our knowledge, this is the first study conducted to analyze the attack surface of Bitcoin covering spatial, temporal, and logical dimensions.

Acknowledgement. This work is supported by Air Force Material Command award FA8750-16-0301, Global Research Lab program of the National Research Foundation NRF-2016K1A1A2912757, and NSF grant CNS-1814614.

References

[1] N. Acheson, "What is segwit?" 2018, https://tinyurl.com/y7d94hbu.
[2] Antpool, "Antpool stratum address," https://www.antpool.com/, 2018.

[3] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *IEEE Symposium on Security and Privacy, SP San Jose, USA*, May 2017, pp. 375–392, https://doi.org/10.1109/SP.2017.29.

[4] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts sok," in *Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204*, 2017, https://tinyurl.com/yd832abs.

[5] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "SoK: Consensus in the Age of Blockchains," 2017, https://arxiv.org/abs/1711.03936.

[6] K. Baqer, D. Y. Huang, D. McCoy, and N. Weaver, "Stressing out: Bitcoin stress testing," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 3–18, http://damonmccoy.com/papers/bitcoin16-final22.pdf.

[7] M. Bastiaan, "Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin," 2015, http://fmt.cs.utwente.nl/files/sprojects/268.pdf.

[8] Blockchain, "Hashrate distribution," 2018, https://blockchain.info/pools.

[9] BTC, "Btc.com stratum address," 2018, https://pool.btc.com/helpCenter?id=miner.

[10] CoinMarketCap, "Cryptocurrency market capitalizations — coinmarketcap," 2018, https://coinmarketcap.com/.

[11] B. Community, "Modify number of bitcoin peers," 2013, https://goo.gl/FggMtn.

[12] ——, "Bitcoin core version history," 2018, https://bitcoin.org/en/version-history.

[13] ——, "Bitcoin developer reference," 2018, https://bitcoin.org/en/bitcoin-for-developersn.com.

[14] ——, "Stratum mining protocol," 2018, https://en.bitcoin.it/wiki/Stratum_mining_protocol.

[15] ——, "Bitnodes: Global bitcoin nodes distribution," 2018, https://bitnodes.earn.com/.

[16] E. Community, "Earn: Earn money by answering messages and completing tasks," 2018, https://earn.com.

[17] N. Community, "National vulnerability database," https://tinyurl.com/y9guktjx.

[18] CVE, "Vulnerability details : Cve-2017-9230," 2018, https://www.cvedetails.com/cve/CVE-2017-9230/.

[19] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proceedings of 13th International Conference on Peer-to-Peer Computing, IEEE P2P, Trento, Italy*, Sep 2013, pp. 1–10, https://doi.org/10.1109/P2P.2013.6688704.

[20] J. Donier and J.-P. Bouchaud, "Why do markets crash? bitcoin data offers unprecedented insights," vol. 10, 03 2015.

[21] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-ng: A scalable blockchain protocol," Mar 2016, https://tinyurl.com/y7gxcdgr.

[22] F2Pool, "F2pool stratum address," 2018, https://www.f2pool.com/help.

[23] M. Fadhil, G. Owenson, and M. Adda, "Locality based approach to improve propagation delay on the bitcoin peer-to-peer network," in *Symposium on Integrated Network and Service Management (IM)*, May 2017, https://doi.org/10.23919/INM.2017.7987328.

[24] G. C. Fanti and P. Viswanath, "Deanonymization in the bitcoin P2P network," in *Annual Conference on Neural Information Processing Systems 2017 Long Beach, CA, USA*, Dec 2017, pp. 1364–1373, https://tinyurl.com/y72zgvtk.

[25] A. Feder, N. Gandal, J. T. Hamrick, and T. Moore, "The impact of DDoS and other security shocks on bitcoin currency exchanges: evidence from mt. gox," *J Cyber Secur*, vol. 3, no. 2, pp. 137–144, Jun. 2017.

[26] A. Gervais, S. Capkun, G. O. Karame, and D. Gruber, "On the privacy provisions of bloom filters in lightweight bitcoin clients," in *Computer Security Applications Conference, ACSAC, New Orleans, LA, USA*, C. N. P. Jr., A. Hahn, K. R. B. Butler, and M. Sherr, Eds. ACM, Dec 2014, pp. 326–335. [Online]. Available: https://doi.org/10.1145/2664243.2664267

[27] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proceedings of the 22nd ACM Conference on Computer and Communications Security (CCS)*, Denver, Colorado, Oct. 2015, pp. 692–705, https://doi.org/10.1145/2810103.2813655.

[28] S. Goldberg, "Why is it taking so long to secure internet routing?" *Commun. ACM*, vol. 57, no. 10, pp. 56–63, Sep. 2014, http://doi.acm.org/10.1145/2659899.

[29] A. Greenberg, "Hacker redirects traffic from 19 internet providers to steal bitcoins," Jun 2017, https://www.wired.com/2014/08/isp-bitcoin-theft/.

[30] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *Proceedings of the 24th USENIX Security Symposium (Security)*, Washington, DC, Aug. 2015, https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman.

[31] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "Tumblebit: An untrusted bitcoin-compatible anonymous payment hub," in *Network and Distributed System Security Symposium, NDSS San Diego, USA*, Feb 2017, http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/ndss201701-3HeilmanPaper.pdf.

[32] G. Hileman and M. Rauchs, "Global cryptocurrency benchmarking study," 2017, https://tinyurl.com/lnx44cf.

[33] B. Info, "Bitcoin block explorer," 2018, https://blockchain.info/.

[34] G. Karame, E. Androulaki, and S. Capkun, "Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin." *IACR Cryptology ePrint Archive*, vol. 2012, no. 248, 2012, http://eprint.iacr.org/2012/248.

[35] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Capkun, "Misbehavior in bitcoin: A study of double-spending and accountability," *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 1, pp. 2:1–2:32, 2015, http://doi.acm.org/10.1145/2732196.

[36] P. Koshy, D. Koshy, and P. D. McDaniel, "An analysis of anonymity in bitcoin using P2P network traffic," in *International Conference on Financial Cryptography and Data Security FC, Christ Church, Barbados*, Mar 2014, pp. 469–485, https://doi.org/10.1007/978-3-662-45472-5_30.

[37] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin," in *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*, Dallas, TX, Oct.–Nov. 2017, https://doi.org/10.1145/3133956.3134019.

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

[39] B. Littlewood and L. Strigini, "Redundancy and diversity in security," in *Proceedings of the 9th European Symposium on Research in Computer Security (ESORICS)*, Sophia Antipolis, France, Sep. 2004, https://doi.org/10.1007/978-3-540-30108-0_26.

[40] J. Mattila, "The blockchain phenomenon–the disruptive potential of distributed consensus architectures," Tech. Rep., 2016.

[41] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, https://bitcoin.org/bitcoin.pdf.

[42] NetAcuity, "Netacuity and netacuity edge ip location technology," Feb 2014, http://www.digitalelement.com/.

[43] M. Polasik, A. I. Piotrowska, T. P. Wisniewski, R. Kotkowski, and G. Lightfoot, "Price fluctuations and the use of bitcoin: An empirical inquiry," *International Journal of Electronic Commerce*, vol. 20, no. 1, pp. 9–49, Sep. 2015, https://doi.org/10.1080/10864415.2016.1061413.

[44] B. Reward, "Bitcoin block reward halving countdown," 2018, http://www.bitcoinblockhalf.com/.

[45] RIR, "Autonomous systems in the world," 2018, https://tinyurl.com/yaz73jnb.

[46] A. Robachevsky, "14,000 incidents: A 2017 routing security year in review," Jan 2018, https://goo.gl/MtiVus.

[47] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International Conference on Financial Cryptography and Data Security FC, Okinawa, Japan*, Apr 2013, https://doi.org/10.1007/978-3-642-39884-1_2.

[48] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "P2P mixing and unlinkable bitcoin transactions," in *Proceedings of the 2017 Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb.–Mar. 2017, https://tinyurl.com/y99reaqs.

[49] M. F. Sallal, G. Owenson, and M. Adda, "Proximity awareness approach to enhance propagation delay on the bitcoin peer-to-peer network," in *International Conference on Distributed Computing Systems ICDCS, Atlanta, USA*, Jun 2017, pp. 2411–2416, https://doi.org/10.1109/ICDCS.2017.53.

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

[51] M. Spagnuolo, F. Maggi, and S. Zanero, "Bitiodine: Extracting intelligence from the bitcoin network," in *International Conference Financial Cryptography and Data Security, Christ Church, Barbados*, Mar 2014, pp. 457–468, https://doi.org/10.1007/978-3-662-45472-5_29.

[52] Statista, "Bitcoin blockchain size 2010-2017 — statistic," 2018, https://tinyurl.com/y8ys8evp.

[53] 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 *25th IEEE International Conference on Computer Communications INFOCOM, Joint Conference of the IEEE Computer and Communications Societies, Barcelona, Spain*, Apr 2006, https://doi.org/10.1109/INFOCOM.2006.154.

[54] S. Williams, "Bitcoin banned countries," 2017, https://tinyurl.com/y8r5gdhl.

[55] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao, "Practical defenses against BGP prefix hijacking," in *Proceedings of the 2007 ACM Conference on Emerging Network Experimentand Technology CoNEXT , New York, USA*, Dec 2007, p. 3, http://doi.acm.org/10.1145/1364654.1364658.