Three Birds with One Stone: Efficient Partitioning Attacks on Interdependent Cryptocurrency Networks

Muhammad Saad
PayPal
muhsaad@paypal.com

David Mohaisen
University of Central Florida
mohaisen@ucf.edu

***Abstract*—The biased distribution of cryptocurrency nodes across Autonomous Systems (ASes) increases the risk of spatial partitioning attacks, allowing an adversary to isolate nodes by hijacking AS prefixes. Prior works on spatial partitioning attacks have mainly focused on the Bitcoin network, showing that the prominent cryptocurrency network can be paralyzed by disrupting the physical topology through BGP hijacks.**

Despite the persisting threat of BGP hijacks, Bitcoin and other cryptocurrencies have not been frequently targeted, likely due to their shielded overlay topology, which limits the exposure of physical network anomalies. In this paper, we present a new perspective by examining the security of cryptocurrency networks, considering shared network resources (*network interdependence*). We conduct measurements extending beyond the Bitcoin network and analyze commonalities in Bitcoin, Ethereum, and Ripple node hosting patterns. We observe that all three networks are highly centralized, predominantly sharing the common ASes. We also note that among the three cryptocurrencies, Ripple does not shield its overlay topology, which can be exploited to learn about the physical network anomalies. The observed network anomalies present practical attack strategies that can be launched to target all three cryptocurrencies simultaneously.[1] We supplement our analysis by surveying recent BGP attacks on high-profile ASes and recognizing a need for application-level countermeasures. We propose attack countermeasures that reduce the risk of spatial partitioning, notwithstanding the increasing centralization of nodes and network interdependence.

***Index Terms*—Partitioning Attacks, Distributed Systems, Cryptocurrencies**

1. Introduction

The cryptocurrency market is currently dominated by Bitcoin and Ethereum, with a combined market capitalization of over $500 Billion at the time of writing this paper [9]. A significant market value makes these cryptocurrencies a lucrative target for attacks, including the spatial partitioning attack in which the adversary exploits the biased distribution of nodes across Autonomous Systems (ASes) to isolate them by launching BGP attacks [2], [18], [35].[2] Spatial partitioning attacks are more profitable and effective if the cryptocurrency network resources (*i.e.,* nodes) are clustered across ASes, allowing the adversary to partition them by hijacking fewer prefixes [2]. Such attacks are subversive for a cryptocurrency since they can result in (1) transaction confirmation delay [37], (2) block propagation delay [15], (3) blockchain forks [30], and (4) decreased mining power [2].

The size of a cryptocurrency network is typically determined by the number of *reachable* full nodes that maintain a blockchain ledger [8]. In the last nine years, the Bitcoin network size has increased from ≈3.5K nodes in 2012 to over ≈9K nodes in 2021. Running a full node in a home network can be costly, often requiring more than 350 GB of storage for the blockchain and a persistent Internet connection to exchange transactions. To alleviate such costs, users nowadays prefer to host their full nodes on cloud services. Despite being cost-effective, cloud hosting invariably contributes to the increasing centralization of nodes across ASes that host the cloud infrastructure [23]. The recent growth of the cryptocurrency market has also led to the expansion of cryptocurrency networks, with more full nodes joining Bitcoin and Ethereum [8]. Assuming that those new nodes also follow the cloud hosting pattern of the existing nodes, it is reasonable to expect that Bitcoin and Ethereum are now more vulnerable to the spatial partitioning attacks, requiring a re-evaluation of their current network distribution.

Prior works on Bitcoin and Ethereum network distribution and security assumed an adversary that targets each network independently [37], [27], [23], [32], [29]. These works did not consider that Bitcoin and Ethereum nodes can exhibit *network interdependence* by sharing the cloud infrastructure hosted within the same set of ASes. Network interdependence can amplify the effect of the spatial partitioning attack by allowing an adversary to target multiple cryptocurrency networks simultaneously [28]. Considering the recent growth of cryptocurrency networks [8] and realistically assuming network interdependence among

This work was supported by NRF under grant 2016K1A1A2912757 and the College of Engineering and Computer Science (CECS) at UCF.

[1] "Practical" signifies an attack where an adversary exploits the routing path anomalies to target other ASes on which it has a low dependency.

[2] In prior works [2], [37], BGP attacks on the Bitcoin network are also called "routing attacks" or "spatial partitioning attacks." For simplicity, in this paper, we use the term 'spatial partitioning attack"

the cryptocurrency nodes, three research questions can be formulated around the spatial partitioning attacks. (1) What is the current distribution of cryptocurrency nodes across the physical network of ASes? (2) To what extent do major cryptocurrencies exhibit network interdependence? (3) How network interdependence changes the spatial partitioning attack model? In this study, we comprehensively answer these questions in §5.

It can be argued that despite being a well-known threat to cryptocurrencies, spatial partitioning attacks are not frequently observed in the wild. As a result, the best outcome of network interdependence analysis is a theoretical postulation of a wide attack surface jointly formed by multiple cryptocurrency networks. Therefore, despite being a persistent threat to cryptocurrency networks, the classical model for spatial partitioning attacks may not provide practical opportunities for an adversary. In this paper, we present a new perspective by studying practical attacks that can be launched on high-profile ASes hosting Bitcoin and Ethereum nodes. For the practical attack construction, we first discuss the key challenges in the existing attack models that limit the practicality of the spatial partitioning attacks.

Prior works [37], [35], [18] show that Bitcoin and Ethereum nodes are hosted in high-profile ASes that may have strong relationships with other ASes on the Internet (also called AS dependency [11]). A strong dependency makes spatial partitioning attacks prohibitively costly and, therefore, impractical. To launch a practical attack, the adversary needs to know: (1) the overlay topology of the cryptocurrency network (*i.e.,* logical connections among nodes), (2) the physical topology of ASes (*i.e.,* the routing paths), and (3) on-path adversarial ASes with a weak dependency on other on-path high profile ASes. We note that this approach cannot be applied to either Bitcoin or Ethereum, since both networks shield their overlay topologies [31], [16]. Therefore, in launching an attack that disrupts connections among the cryptocurrency nodes, an adversary is expected to approximate the overlay topology, map it onto the ASes to learn the physical topology, enumerate the probable routing paths, and uncover the routing path anomalies. A shielded overlay topology in Bitcoin and Ethereum compounds this effort, thereby preventing the adversary from launching practical partitioning attacks.

Unique to this work, we show that the adversary can launch a practical attack if it can identify a third cryptocurrency that exhibits network interdependence with both Bitcoin and Ethereum while publicly disclosing the overlay topology. Towards that end, we identified Ripple as a suitable candidate with a notable network size and nodes hosted alongside Bitcoin and Ethereum nodes in the same set of ASes. Moreover, Ripple nodes publicly disclose their Peer-to-Peer (P2P) connections, which form the overlay topology. The overlay topology can reduce the adversarial effort since the adversary can conveniently map the overlay topology to the physical topology of ASes, and launch attacks upon observing the routing path anomalies.[3] As such, given the network interdependence among the three cryptocurrencies, if ASes with Ripple nodes are hijacked, the effect cascades across Bitcoin and Ethereum nodes resulting in practical attacks that affect all three cryptocurrencies simultaneously.

Our contributions extend beyond uncovering novel and practical attack strategies. Realizing the risk of practical partitioning attacks and their impact on notable cryptocurrencies, we develop and evaluate robust countermeasures to counter practical partitioning attacks. Our proposed countermeasures reduce the risk of attacks, notwithstanding the increasing centralization of cryptocurrency nodes and network interdependence.

Contributions and Roadmap. In summary, we make the following key contributions in this work.

1) **Network Distribution Analysis.** We deploy crawlers in Bitcoin, Ethereum, and Ripple networks to analyze their distribution across ASes (§3). Compared to the reports in prior works [37], [35], we observe an increasing centralization of nodes across ASes.
2) **Network Interdependence Measurement.** We measure and characterize the interdependence among the cryptocurrency networks and find a strong similarity in the node hosting pattern across high-profile ASes. We observe that five of the top ten ASes in one cryptocurrency are also among the top ten ASes across the other two cryptocurrencies (§5.1).
3) **Attack Construction.** Based on the node distribution and network interdependence, we propose two types of spatial partitioning attacks, namely the classical attack (§5) and the practical attack (§6). In the classical attack, we consider a myopic adversary (as in prior works [2], [37]) that indiscriminately targets the high-profile ASes irrespective of the AS dependency. In the practical attack, we assume a sophisticated adversary that launches an attack if it has a low dependency on other ASes. We find that an adversary can isolate ≈9.4K and ≈4.3K nodes from all cryptocurrency networks in the classical and practical attacks, respectively.
4) **Attack Simulation and Countermeasures** We simulate the spatial partitioning attacks in a controlled setup to demonstrate their impact on cryptocurrency systems. Accordingly, we propose attack countermeasures and thoroughly investigate their efficacy in real-world deployments (§7).

The rest of this paper includes background and related work in §2, and concluding remarks in §8.

2. Background and Related Work

In this section, we provide a brief background on the spatial partitioning attacks and the prior works that explored partitioning attacks on cryptocurrency networks.

[3] Note that the decrease in the attack effort is due to Ripple's publicly available overlay topology. In Bitcoin and Ethereum, the adversary is expected to estimate the possible overlay topologies. Although this approach is possible, it can be costly and infeasible.

Internet traffic is managed by the Internet Service Providers (ISPs) that own networks of routers called Autonomous Systems (ASes) [25]. ASes own sets of IP prefixes that are used to route traffic between end hosts within or outside an AS. Traffic forwarding rules between ASes are implemented through the Border Gateway Protocol (BGP), and the routing path between ASes is called the AS path. The BGP protocol follows a weak trust model where prefix announcements are not validated by ASes [2]. This weakness creates an opportunity for adversaries to hijack the BGP prefixes of a target AS.

In 2016, Apostolaki *et al.* [2] highlighted the biased distribution of Bitcoin nodes across ASes, with 50 ASes hosting ≈50% of the Bitcoin nodes. They showed that an adversary can exploit the biased distribution of Bitcoin nodes across a few ASes, and the weak trust model of the BGP protocol, to partition the Bitcoin network by hijacking a few BGP prefixes. In 2018, Saad *et al.* [37] reported an increasing centralization of Bitcoin nodes across ASes, highlighting the growing risk of partitioning attacks.

The biased distribution of nodes has also been observed in the Ethereum network [34], [18], and the partitioning effects are found to be similar to the Bitcoin network (*i.e.,* blockchain forks). To counter the spatial partitioning attacks, Apostolaki *et al.* [1] proposed a network called SABER that routes the cryptocurrency traffic through secure and scalable relay networks across the Internet.

In 2015, Heilman *et al.* [27] showed that an adversary could partition Bitcoin nodes by occupying their incoming and outgoing connection slots. In 2020, Tran *et al.* [39] presented a stealthier version of the attack proposed in [27], whereby an adversarial AS floods the IP tables of a victim node to occupy the incoming and outgoing connections of the node. To counter such attacks in the wild, the Bitcoin community adopted *Asmap* [33], in which a node establishes each outgoing connection to a different AS. In 2021, Fan *et al.* improved upon [39] by presenting the *ConMan* attack that partitions a victim node by occupying its connection slots. In contrast to [39], where the adversary patiently waits for several weeks to succeed, *ConMan* showed that the adversary could partition the victim within a few minutes.

As discussed in §1, despite the risk of partitioning attacks highlighted in the prior works, such attacks have not been observed in the wild due to a shielded overlay topology and strong dependency among on-path ASes. Moreover, prior works on Bitcoin and Ethereum partitioning attacks assumed that the adversary targets each network independently [27], [34], [19]. These works did not consider that Bitcoin and Ethereum nodes could exhibit network interdependence by sharing the same set of ASes. We note that network interdependence amplifies the effect of spatial partitioning attacks, whereby an adversary can target multiple cryptocurrencies simultaneously. Finally, the proposed countermeasures like SABER [1], while effective in preventing attacks, may introduce a notion of network centralization by mandating the deployment of relay networks in designated locations. In this work, we bridge these gaps by presenting an up-to-date distribution of three prominent cryptocurrency networks across ASes while acknowledging network interdependence. We also uncover novel attack vectors which enable practical attacks in the wild and propose effective countermeasures that reduce the risk of partitioning attacks.

3. Data Collection and Methodology

In this section, we present our data collection and experiment methodology. For data collection, we used publicly available repositories, including Bitnodes, Ethernodes, and XRP Ledger [8], [10], [12] that provide data from Bitcoin, Ethereum, and Ripple networks, respectively. We deployed crawlers for over one month to collect the IP addresses of nodes from each repository. Our crawlers obtained two snapshots of each network every day. Through the Ripple peer crawling method, we also obtained the Ripple overlay topology [40].[4]

IP to AS Mapping. After collecting the IP addresses of the cryptocurrency nodes, we used the *RouteViews* dataset [13] to perform IP to AS mapping. The *RouteViews* dataset is available on CAIDA data server [5], aggregating BGP prefix announcements by ASes and the Autonomous System Numbers. Through IP to AS mapping, we obtained the ASes and the corresponding prefixes that host Bitcoin, Ethereum, and Ripple nodes. Furthermore, by grouping ASes, we measured (1) the distribution of cryptocurrency nodes across them, and (2) the network interdependence, characterized by the ASes shared by the three cryptocurrency node types.

Routing Path Extraction. We applied the IP to AS mapping on the Ripple overlay topology to identify connections among ASes that host Ripple nodes. We then used CAIDA's *BGPStream* tool [4] to obtain the routing paths taken by the messages exchanged between Ripple nodes. *BGPStream* is an open-source software stack that provides historical BGP announcements and lists on-path ASes. By examining the BGP announcement at the time of data collection, we identified the on-path ASes that can intercept the communication between Ripple nodes.

AS Dependency. Practical partitioning attacks require an adversary to identify on-path ASes with weak relationships with high-profile ASes hosting cryptocurrency nodes. The relationships between ASes are based on their capability of forwarding data to other ASes on the Internet [17]. Such relationships create a dependency among ASes, and in the practical attack, we assume that the adversarial AS has a low dependency on ASes that host the cryptocurrency nodes. To measure the dependency of on-path ASes, we used the AS hegemony metric recently introduced by the Internet Health Report (IHR) [11], [21] (details provided in §6). IHR scales the AS hegemony between 0–1, with 1 indicating a high dependency and 0 indicating a low dependency.

[4] Our analysis is based on the *reachable* nodes since they form the backbone of a cryptocurrency network. We do not cover the *unreachable* nodes behind NAT since they are inevitably partitioned when the *reachable* nodes are isolated. We analyze nodes with a public IP address among the *reachable* nodes.

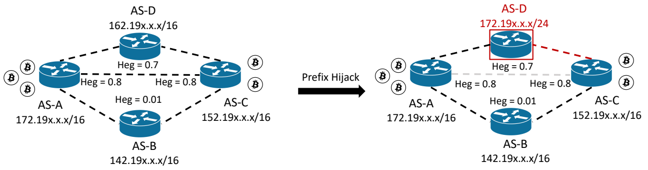


Figure 1. An illustration of the classical partitioning attack. The AS dependency is shown as the global hegemony score (0–1). Despite a high AS dependency, AS-D launches an attack on AS-A which may eventually affect its relationships with other ASes.

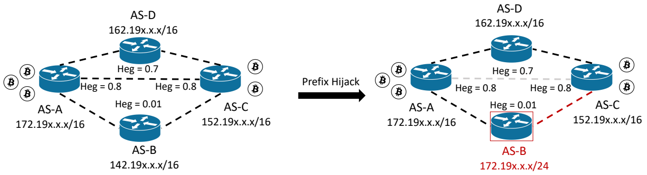


Figure 2. Illustration of practical partitioning attack. The AS dependency is shown as the global hegemony score (0–1). AS-B announces more specific prefixes of AS-A to hijack its traffic. AS-B launches the attack due to a low AS dependency in the AS path. The relationship between low AS dependency and BGP attacks has been observed in the wild [6].

To summarize our data collection and methodology, we (1) collected data from the three cryptocurrency networks, (2) performed IP to AS mapping to extract BGP prefixes and characterize network centralization and interdependence, (3) mapped the Ripple overlay topology onto the physical network topology to identify the routing paths, and (5) measured the AS dependency using the AS hegemony scores.

4. Threat Model

We now present the threat model for the spatial partitioning attack. Taking into account various network entities and adversary types, we present a formal characterization of spatial partitioning attacks.

4.1. Analysis Notations

Network Anatomy. For our analyses, we define A as the set of ASes that host the cryptocurrency nodes, with each $a_i \in \mathsf{A}$ hosting N_i^b, N_i^e, and N_i^r Bitcoin, Ethereum, and Ripple nodes hosted across P_i^b, P_i^e, and P_i^r prefixes, respectively. We also assign R_i^b, R_i^e, and R_i^r ranks to each $a_i \in \mathsf{A}$ based on the percentage of nodes it hosts from each network. [5]

Network Interdependence. To characterize network interdependence, we define $\mathsf{N}_i^c = \mathsf{N}_i^b + \mathsf{N}_i^e + \mathsf{N}_i^r$ as the cumulative number of cryptocurrency nodes hosted by an AS. Based on the N_i^c value, we assign R_i^c rank to an AS, and define P_i^c as the number of prefixes that host all N_i^c nodes.

4.2. Attack Types and Adversaries

We classify the spatial partitioning attacks into two types, namely the *classical attack* and the *practical attack.*

4.2.1. Classical Attack. In the classical attack, we assume a myopic adversary that does not acknowledge the AS dependency and indiscriminately targets the high-profile ASes that host a large number of cryptocurrency nodes. To launch the attack, the adversary selects an AS and identifies BGP prefixes that host the cryptocurrency nodes. The adversary then announces more specific BGP prefixes to hijack the traffic of the victim AS.

Figure 1 provides an illustration of the classical partitioning attack, where two among the four ASes (AS-A and AS-C) are shown to host cryptocurrency nodes. Each AS announces a set of prefixes which helps in determining the AS paths for traffic forwarding. Moreover, ASes are annotated by their respective global hegemony scores reported by the BGP viewpoints [6], and computed using paths to all IP prefixes [11]. In Figure 1, AS-A, AS-C, and AS-D have high hegemony scores, meaning that they are commonly used to reach other hosts on the Internet. In contrast, AS-B has a low hegemony score, which shows a low dependency of other ASes on AS-B to reach the destination IP addresses [6].

Figure 1 also shows a direct link between AS-A and AS-C, indicating that if nodes in AS-A have logical connections with nodes in AS-C, their traffic is directly forwarded by the respective ASes. In the hijack event, AS-D announces more specific prefixes of the victim AS-A. By design, the BGP protocol prefers the shortest and more specific path to the destination IP address. As a result, AS-C forwards the traffic destined for AS-A to AS-D. The change in the traffic flow breaks the logical connections between the cryptocurrency nodes in AS-A and AS-C. Moreover, traffic destined towards AS-A from all other ASes is also forwarded to AS-D, which isolates the cryptocurrency nodes in AS-A.

Note that AS-D launches the attack on AS-A despite a high AS dependency, which is likely to affect their business relationships. We categorize this attack as the classical attack whereby a myopic adversary indiscriminately targets another AS while risking the AS relationships.

4.2.2. Practical Attack. In the practical attack, we assume an adversary that acknowledges the AS relationships and does not indiscriminately target the high profile ASes.[6]. The adversary obtains the routing paths taken by the cryptocurrency traffic and calculates the global hegemony scores of the on-path ASes. If the adversary finds itself on the routing path and has a low global hegemony score, it launches the attack on the ASes that host the cryptocurrency nodes. [7]

Figure 2 provides an illustration of a practical partitioning attack where all the initial conditions are similar to the ones shown previously in Figure 1. In this case, however,

[5] Superscripts denote the first letter of the cryptocurrency name. For instance, in N_i^b, N is the number of nodes and b is for the Bitcoin network.

[6] High profile ASes (*i.e.,* AS7018) typically own thousands of IP prefixes and route a high volume of upstream or downstream traffic for their neighboring ASes.

[7] If the victim node IP address already exists in the /24 prefix, ISPs may filter the adversary's longer prefix announcement [2]. In such a case, the on-path adversary can announce a /24 prefix and a shorter path to the victim AS to launch the partitioning attack and hijack the AS traffic.

TABLE 1. Top ten ASes that host Bitcoin, Ethereum, and Ripple nodes, ranked based on the total number of nodes hosted from the three networks (N_i^c). For each AS, we also show the total percentage of nodes with respect to each cryptocurrency. Note that the number of nodes is an average over all data samples while the number of prefixes shows all unique prefixes from the AS collected from all samples. AS-7018 hosts a large number of Bitcoin and Ethereum nodes, but no Ripple node. Therefore, while its global ranking (R_i^c) is 10, the corresponding R_i^r, N_i^r, and P_i^r values are marked as NA.

ASN	R_i^b	R_i^e	R_i^r	R_i^c	N_i^b	N_i^e	N_i^r	N_i^c	P_i^b	P_i^e	P_i^r	P_i^c
AS-16509	2	2	1	1	442 (6.8%)	1189 (19.1%)	252 (33.6%)	1883	131	120	83	150
AS-14618	10	1	3	2	83 (1.3%)	1289 (20.7%)	65 (8.7%)	1437	36	50	26	51
AS-24940	1	3	2	3	898 (13.8%)	347 (5.6%)	160 (21.4%)	1405	35	32	29	35
AS-16276	4	5	4	4	347 (5.3%)	142 (2.3%)	35 (4.8%)	524	60	49	22	67
AS-14061	3	20	6	5	412 (6.3%)	35 (0.6%)	12 (1.7%)	459	87	44	14	87
AS-7922	5	4	18	6	228 (3.5%)	211 (3.4%)	3 (0.5%)	442	68	67	8	85
AS-396982	7	6	5	7	120 (1.8%)	136 (2.2%)	35 (4.7%)	291	56	56	31	76
AS-701	8	8	89	8	115 (1.8%)	103 (1.7%)	1 (0.1%)	218	119	127	2	172
AS-51167	6	12	25	9	152 (2.3%)	58 (0.9%)	2 (0.4%)	212	49	57	6	74
AS-7018	9	10	NA	10	108 (1.7%)	74 (1.2%)	NA	183	98	89	NA	148

AS-B announces more specific prefixes of AS-A, which results in traffic destined for AS-A being routed to AS-B. Unlike AS-D in the classical attack, AS-B has a low AS dependency, indicated by a low global hegemony score. Note that on a given routing path, the presence of an AS with a low global hegemony score between two or more ASes with high global hegemony scores is considered an anomaly [6]. AS-B exploits this anomaly to target AS-A feasibly. Our practical attack construction is based on such routing path anomalies, whereby the adversary with a low hegemony score launches the attack if it finds itself on a path between ASes with high hegemony scores. For more details on the correlation between AS hegemony and BGP hijacks, we refer to [6].

4.2.3. Adversary Types.

In both attack types, the adversary isolates the cryptocurrency nodes from the rest of the network. The adversary can target a specific cryptocurrency or all three cryptocurrency networks. To model the network-specific choices, we specify four types of adversaries in the classical attack and the practical attack.

Bitcoin Adversary. For the Bitcoin network, we define an adversary, $\mathcal{A}_b$, that only targets ASes that host Bitcoin nodes. To do so, $\mathcal{A}_b$ ranks all ASes based on the number of Bitcoin nodes they host (R_i^b), and selects T_N number of target nodes to isolate. Without the overlay topology knowledge, $\mathcal{A}_b$ can only launch the classical attack.

Ethereum Adversary. For the Ethereum network, we define an adversary, $\mathcal{A}_e$, that only targets ASes that host Ethereum nodes. Similar to $\mathcal{A}_b$, $\mathcal{A}_e$ ranks all ASes based on the number of Ethereum nodes they host, and selects T_N number of nodes to isolate by hijacking their prefixes. Without the overlay topology knowledge, $\mathcal{A}_e$ can only launch the classical partitioning attack.

Ripple Adversary. For the Ripple network, we define an adversary, $\mathcal{A}_r$, capable of launching both the classical attack and the practical attack. For the classical attack, $\mathcal{A}_r$ ranks the high-profile ASes and hijacks their prefixes to isolate T_N number of nodes. For the practical attack, $\mathcal{A}_r$ maps the Ripple overlay topology onto the physical network topology to discover the routing paths. If $\mathcal{A}_r$ is on the routing path with a significantly low hegemony score compared to the other on-path ASes, $\mathcal{A}_r$ announces BGP prefixes of the target ASes to hijack their traffic.

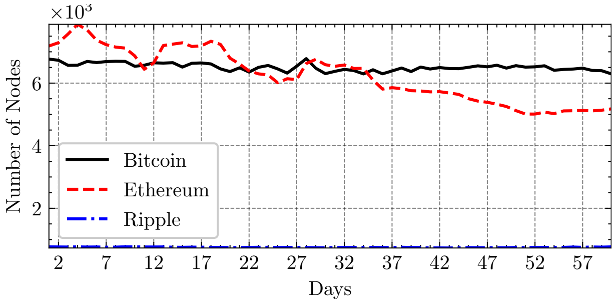


Figure 3. Number of IP addresses collected from the three networks during the measurement duration. On average, Bitcoin, Ethereum, and Ripple had 6,510, 6,233, and 751 nodes with public IP addresses.

Global Adversary. Finally, we define a global adversary, $\mathcal{A}_g$, that exploits the network interdependence to target all three networks simultaneously. For the classical attack, $\mathcal{A}_g$ ranks all ASes based on the cumulative number of nodes they host from all three networks. $\mathcal{A}_g$ then isolates T_N nodes by announcing more specific BGP prefixes of their ASes. For the practical attack, $\mathcal{A}_g$ simply follows the attack procedure of $\mathcal{A}_r$ and observes if it is on the routing path between ASes that host a large number of Bitcoin, Ethereum, and Ripple nodes. Unlike $\mathcal{A}_r$, which targets ASes based on their ranking in the Ripple network, $\mathcal{A}_g$ ranks ASes based on the total number of nodes from all cryptocurrencies and targets the vulnerable ASes based on the AS dependency.

4.2.4. Attack Cost.

We determine the attack cost by calculating the number of prefixes required to isolate T_N number of nodes. The adversary can identify the top-ranked ASes and sort their prefixes based on the number of hosted nodes. Subsequently, the adversary can target each prefix by announcing a more specific prefix and isolating nodes in each announcement. For instance, assume P^b is a set of prefixes, where each P_i^b hosts N_i^b Bitcoin nodes. Moreover, each $P_i^b \in P^b$ is sorted in descending order based on N_i^b. If $\mathcal{A}_b$ aims to isolate a total of T_N nodes, where $T_N = \sum_{\forall i} N_i^b$, the total attack cost $\mathcal{C}_b$ becomes $\mathcal{C}_b = \sum_{\forall i} P_i^b$.[8]

[8] Equations can be extended for cost computation of $\mathcal{A}_e$, $\mathcal{A}_r$, and $\mathcal{A}_g$.

5. Classical Partitioning Attack

In this section, we analyze the classical partitioning attack. We first present the preliminary results, followed by a standalone analysis of each cryptocurrency network and the joint analysis based on network interdependence.

5.1. Preliminary Results

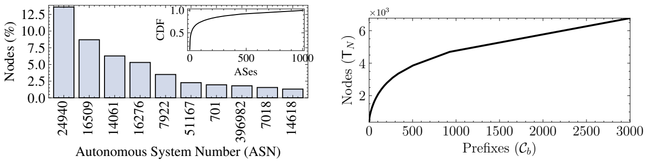
In Figure 3, we report the number of nodes observed in each network between November 2022 and December 2022.[9] On average, each day, we found ≈6,510, 6,233, and 751 nodes in Bitcoin, Ethereum, and Ripple, respectively. In Table 1, we elaborate on the characteristics of each network by reporting their distribution in the top ten ASes, highlighting (1) AS ranking (R_i^b, R_i^e, R_i^r, and R_i^c), (2) the number of nodes hosted by an AS (N_i^b, N_i^e, N_i^r, and N_i^c), and (3) the prefix-wise node distribution in an AS (P_i^b, P_i^e, P_i^e, and P_i^c). The N_i^b, N_i^e, N_i^r, and N_i^c values reported in Table 1 are averages of all data samples obtained during the measurement duration. In contrast, the P_i^b, P_i^e, P_i^e, and P_i^c values are all the distinct prefixes owned by the ASes collected over the entire measurement duration. Therefore, in some cases (*i.e.,* Ripple nodes in AS-701), the number of nodes might appear to be less than the number of prefixes. That is because the AS hosted the nodes only a few times and across different unique prefixes. As a result, the average node count over the measurement duration appeared to be smaller than the unique prefix count. Nevertheless, across all samples, we observed a generally consistent node distribution across ASes, indicating that an attack model formulated on one sample can be generalized across the dataset.

A few notable observations in Table 1 include: (1) a strong network interdependence, since five out of the top ten ASes in one cryptocurrency network are also among the top ten ASes across all three networks, (2) variations in AS rankings despite interdependence (*i.e.,* for AS-14618, $R_i^b = 10$, $R_i^e = 1$, $R_i^r = 3$, and $R_i^c = 2$), and (3) variations in the prefix-wise distribution of nodes across ASes (*i.e.,* for AS-24940, $P_i^b = 34$, and for AS-16509, $P_i^e = 119$). From these observations, we concluded that there is a general similarity in the hosting pattern of cryptocurrency nodes across ASes. Moreover, the attack cost can vary across ASes due to the varying prefix-wise node distribution.

5.2. The Bitcoin Network

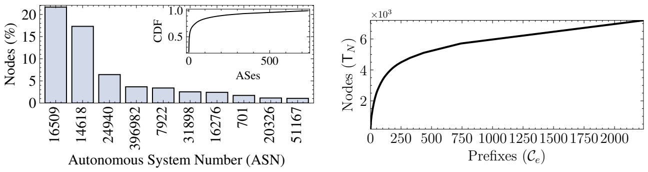
In this section, we analyze the classical partitioning attack on the Bitcoin network. Given that a BGP hijack duration is usually between a few minutes to one day [41], [36], we, therefore, use one snapshot from our dataset for an accurate evaluation of all attacks presented below.

The Bitcoin network analysis revealed that all nodes were hosted in 996 unique ASes, with only 14 ASes hosting 50% of the nodes. In Figure 4(a), we show the distribution of the Bitcoin nodes across the top ten ASes. In terms of the prefix-wise distribution, all Bitcoin nodes were hosted across 3,003 unique prefixes, with 50% of the nodes hosted in 347 prefixes. Compared to the study conducted in 2017 [2], our analysis shows an increasing centralization of Bitcoin nodes.



(a) Nodes Distribution (b) Classical Attack

Figure 4. 4(a) shows the distribution of Bitcoin nodes across ASes. The outer plot shows the percentage of nodes hosted by the top ten ASes while the inner plot shows the CDF of the Bitcoin nodes across all ASes. The Bitcoin nodes were hosted across 996 unique ASes with 14 ASes hosting 50% of the nodes. 4(b) plots the sorted prefix-wise node distribution which is exploited by $\mathcal{A}_b$ in the classical attack. The number of prefixes on the x-axis can also be considered as the attack cost $\mathcal{C}_b$.



(a) Nodes Distribution (b) Classical Attack

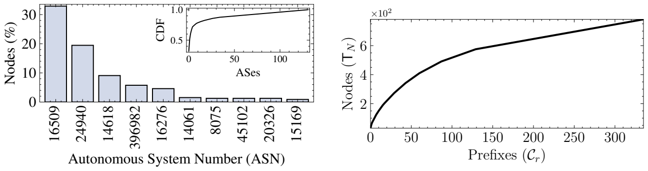
Figure 5. 5(a) shows the distribution of Ethereum nodes. Ethereum nodes are more centralized than Bitcoin, with five ASes hosting 50% of the nodes. 4(b) plots the prefix-wise distribution of nodes which is exploited by $\mathcal{A}_e$ in the classical attack.

To launch a classical partitioning attack on the Bitcoin network, $\mathcal{A}_b$ sorts all prefixes based on their node distribution and iteratively announces more specific prefixes. As a result, $\mathcal{A}_b$ hijacks the set of prefixes from each AS that hosts the Bitcoin nodes. The prefix-wise sorting reduces the attack cost since $\mathcal{A}_b$ can isolate the targeted number of nodes while announcing the minimum number of prefixes. For instance, if $\mathcal{A}_b$ aims to isolate 1,000 Bitcoin nodes in Table 1, it can select various prefix combinations where the total sum of hosted nodes is 1,000. However, if those nodes are hosted in a wide range of prefixes, the attack cost will increase accordingly. Alternatively, $\mathcal{A}_b$ can sort prefixes based on the maximum number of hosted nodes and target them to reduce the attack cost. In Figure 4(b), we show how $\mathcal{A}_b$ can exploit the prefix-wise node distribution to launch the classical partitioning attack. Figures 4(a) and 4(b) demonstrate that the biased distribution of nodes makes them highly vulnerable to the classical partitioning attack.

5.3. The Ethereum Network

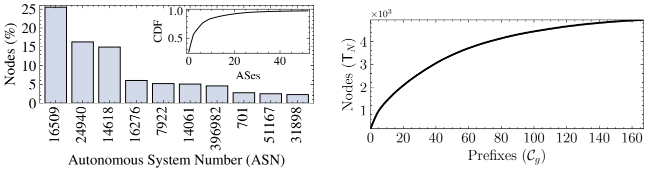
In the Ethereum network, we observed that all nodes were hosted across 734 unique ASes, with only five ASes hosting 50% of the nodes. Compared to the Bitcoin network, we found that Ethereum nodes were more centralized across ASes with AS-16509, AS-24940, and AS-14618 jointly hosting more than 40% of the nodes. A high centralization of nodes across a few ASes indicates that Ethereum is more vulnerable to spatial partitioning attacks than Bitcoin. Figure 5(a) shows the distribution of Ethereum nodes across the top ten ASes. In terms of the prefix-wise distribution,

[9] Our extended paper in [38] provides a longer timeline for the node distribution shown in Table 1 and Figure 3.



(a) Nodes Distribution (b) Classical Attack

Figure 6. 6(a) shows the distribution of the Ripple nodes across ASes. All Ripple nodes were hosted across 131 unique ASes with only two ASes hosting more than 50% of the nodes. 6(b) plots the prefix-wise node distribution, which is exploited by $\mathcal{A}_r$ in the classical attack.



(a) Nodes Distribution (b) Classical Attack

Figure 7. 7(a) shows the distribution of the nodes that jointly host Bitcoin, Ethereum, and Ripple nodes. All nodes are hosted across 53 ASes with AS-16509 hosting more than 20% of the nodes. 7(b) plots the prefix-wise node distribution which is exploited by $\mathcal{A}_g$ in the classical attack.

we found that Ethereum nodes were hosted in 2,238 unique IP prefixes, with 122 prefixes hosting 50% of the nodes.

In Figure 5(b), we plot the sorted prefix-wise node distribution, which can be exploited by $\mathcal{A}_e$ to launch the classical attack on the Ethereum network. Figure 5(b) confirms the centralization of nodes across prefixes, primarily caused by AS-16509, AS-24940, and AS-14618. An interesting observation in our analysis was the prefix-wise distribution of the Bitcoin and Ethereum nodes in AS-16509. As shown in Table 1, on average, AS-16509 hosts 442 Bitcoin nodes across 131 prefixes and 1,189 Ethereum nodes across 120 prefixes. Although the Ethereum nodes outnumber the Bitcoin nodes, their centralized distribution across prefixes makes them more vulnerable to the classical attack.

5.4. The Ripple Network

Consistent with our prior observations in Bitcoin and Ethereum, we continued to observe a biased distribution of the Ripple nodes across ASes. We found that all Ripple nodes were hosted across 131 unique ASes, where only two ASes hosted 50% of the nodes. In particular, AS-16509 hosted ≈33% of all Ripple nodes, which is the largest percentage of nodes hosted by any AS among all three networks. In Figure 6(a), we show the distribution of Ripple nodes across the top ten ASes. In terms of the prefix-wise distribution, we found that the Ripple nodes were hosted across 336 unique prefixes, with only 57 prefixes hosting 50% of the nodes.

Figure 6(b) presents the prefix-wise node distribution which can be exploited by $\mathcal{A}_r$ to launch the classical partitioning attack. We observed that Ripple is the most vulnerable cryptocurrency network to classical partitioning attacks due to two main factors. First, $\mathcal{A}_r$ can target fewer nodes in Ripple, since the network size is considerably smaller than both Bitcoin and Ethereum. Second, in addition to the smaller network size, Ripple nodes are also centralized across a few ASes and their prefixes. Both factors allow $\mathcal{A}_r$ to hijack fewer prefixes in order to isolate all Ripple nodes.

5.5. Joint Network Analysis

For the joint network analysis, we selected all ASes that hosted at least one node from each cryptocurrency and analyzed the distribution of the nodes across those ASes. Our results revealed that 9,427 Bitcoin, Ethereum, and Ripple nodes were hosted by 53 ASes, with only three ASes hosting ≈50% of the nodes. In terms of the prefix-wise distribution, we found that all 9,427 nodes were hosted in 1,224 unique prefixes, and $\mathcal{A}_g$ can isolate 50% of the nodes by hijacking only 80 prefixes.

In Figure Figure 7, we plot the prefix-wise distribution of nodes that share the same prefixes across all three cryptocurrencies. Consistent with the prior observations, we found a biased distribution of nodes across ASes which can be exploited by $\mathcal{A}_g$ to launch the classical partitioning attack. Moreover, due to the strong network interdependence (9,427 nodes sharing the same ASes) the classical attack can simultaneously impact all three cryptocurrency networks.

5.6. Key Takeaways

From the classical partitioning attack analysis, we made the following key conclusions. (1) All three cryptocurrency networks are clustered across a few ASes, making them highly vulnerable to the spatial partitioning attack. (2) The centralization of cryptocurrency nodes has increased over the last five years [2]. (3) There is a strong network interdependence among all three networks, which demonstrates that targeting any high-profile AS affects all three networks simultaneously. (4) The prefix-wise node distribution can be exploited to isolate the maximum number of targeted nodes while minimizing the number of prefix announcements.

6. Practical Partitioning Attacks

As discussed in §1, Bitcoin's vulnerability to the classical partitioning attack is well-known and documented. However, despite the persistent threat and lucrative outcomes for an adversary, classical attacks are not frequently observed in the wild. A probable reason for the attacks' rarity is the shielded nature of the Bitcoin overlay topology, which prevents an on-path adversary from identifying potential victims to target. However, if an AS hosts a significant number of cryptocurrency nodes, its neighboring ASes invariably learn that, as they intercept the cryptocurrency traffic, and can possibly hijack it. Since neighboring ASes may have a dependency on each other (*i.e.,* routing upstream or downstream traffic), it prevents them from launching such an attack without affecting the AS relationships [21].

On the other hand, if the adversary learns the overlay topology of a cryptocurrency network and finds itself on the AS path in the physical topology with a low dependency

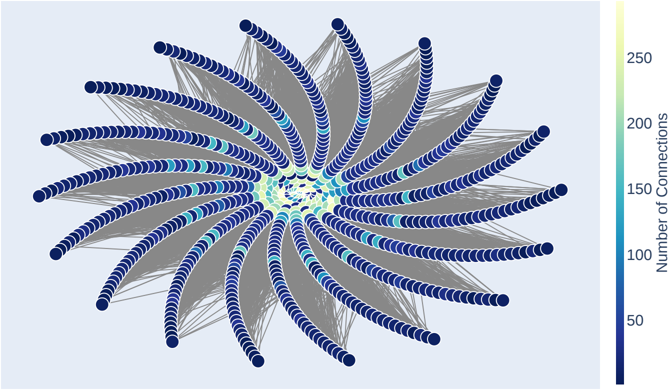


Figure 8. Ripple overlay topology consisting of 781 nodes and 18,335 edges. The graph key color codes the number of connections per node.

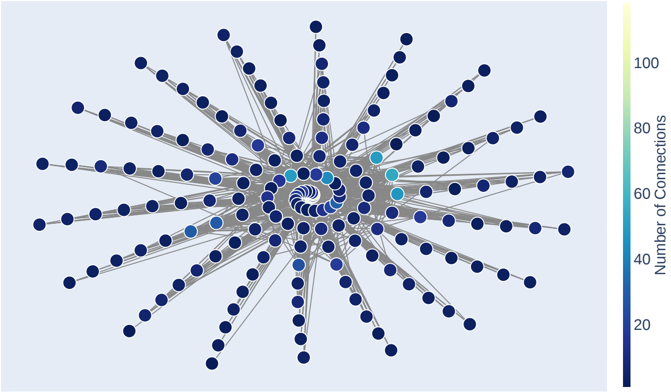


Figure 9. Physical topology of ASes that host Ripple nodes. The graph consists of 175 vertices and 771 edges.

Algorithm 1: Physical Topology Construction

```
Input: Overlay topology $G = (V, E)$
Initialize: Physical topology $G^{'} = (V^{'}, E^{'})$
foreach $\mathrm{ip}_i \in V$ do
    send $\mathrm{ip}_i$ to RouteViews dataset
    receive $\mathrm{AS}_i$ for $\mathrm{ip}_i$
    if $\mathrm{AS}_i \notin V^{'}$ then
        $V^{'} \leftarrow \mathrm{AS}_i$
foreach $(\mathrm{ip}_i, \mathrm{ip}_j) \in E$ do
    send $(\mathrm{ip}_i, \mathrm{ip}_j)$ to RouteViews dataset
    receive $(\mathrm{AS}_i, \mathrm{AS}_j)$ for $(\mathrm{ip}_i, \mathrm{ip}_j)$
    if $(\mathrm{AS}_i, \mathrm{AS}_j) \notin E^{'}$ then
        $E^{'} \leftarrow (\mathrm{AS}_i, \mathrm{AS}_j)$
return $G^{'} = (V^{'}, E^{'})$
```

on high-profile ASes, the adversary can afford to target the high-profile ASes. Therefore, the ability to launch an attack due to a low dependency leads to practical attacks that are also observed in the wild [6]. The state-of-the-art partitioning attack models have not concretely evaluated the possibility of practical attacks on Bitcoin and Ethereum due to their shielded overlay topologies. As discussed in §1, in a shielded overlay topology, the adversary estimates the possible connections between nodes, followed by the probable routing paths for the traffic exchanged between those connections. Since the possible number of overlay topologies can be significant, this process increases the effort of launching a successful attack. In contrast, if the adversary identifies a third cryptocurrency whose nodes (1) share the same set of ASes with Bitcoin and Ethereum nodes (network interdependence), and (2) publicly disclose their overlay topology, the adversary can effectively reduce the effort. With the overlay topology knowledge, the adversary can easily identify the number of AS paths between the source node and the destination node. If the adversary is on one of those paths with a low dependency on victim ASes, the adversary can launch a practical attack.

In order to formulate the practical partitioning attack, we explored the cryptocurrency ecosystem to find a suitable candidate that shares ASes with both Bitcoin and Ethereum while disclosing its overlay topology. We found Ripple as the most suitable candidate since it satisfied the requirements for the practical attack. Additionally, Ripple can also be lucrative for an adversary due to a market capitalization of over $9 billion. At the time of writing this paper, Ripple ranked seventh among the top cryptocurrencies in terms of market capitalization [9]. Therefore, by launching a practical attack, the adversary can target three of the top seven valuable cryptocurrencies.

Attack Construction. For the practical attack analysis, we mapped the Ripple overlay topology onto the physical topology to identify links between ASes. We then used the *BGPStream* API [4] to extract the routing paths, followed by AS dependency measurements using the “Internet Health Report” (IHR) API [11]. We studied irregularities in the routing paths by discovering on-path ASes that can intercept cryptocurrency traffic while enjoying a low dependency on ASes that hosted Ripple nodes. Finally, we developed case studies to show how practical attacks can be launched by exploiting the observed routing path irregularities.

It is plausible to assume that high-profile ASes take strong security measures for their routing paths, and that the practical attack adversary may only find irregularities in the routing paths of ASes that do not host a significant number of cryptocurrency nodes. Therefore, the *efficacy* of the practical attacks relies on discovering irregularities in the routing paths of high-profile ASes that host a significant number of cryptocurrency nodes. Our analysis in the subsequent sections reveals that practical partitioning attacks can be launched on high-profile ASes that host a large number of cryptocurrency nodes.

6.1. Ripple Network Topology

The first step towards constructing the practical attacks is to map the Ripple overlay topology onto the physical topology of ASes, which we show in the following.

Overlay Topology. We specify $G = (V, E)$ as the overlay topology formed by the Ripple nodes and $G^{'} = (V^{'}, E^{'})$ as the physical topology formed by ASes that host those nodes. Figure 8 presents the overlay topology consisting of 781 nodes and 18,335 edges. We found the average node degree to be 46.95, with a maximum value of 293.

Physical Topology. From the overlay topology, we then constructed the physical network topology to identify the

Algorithm 2: AS Path Collection

```
Input: ASes in $V'$
Initialize: ASPaths = [], counter = 0
while counter $\leq$ 50,000 do
    foreach Path from RIB do
        if set(Path) $\cap V' \geq 2$ then
            ASPaths $\leftarrow$ list(Path)
            counter += 1
return ASPaths
```

logical connections among ASes that host Ripple nodes. For the physical topology construction, we took all vertices V in the overlay topology G and performed IP to AS mapping using the *RouteViews* dataset [13]. The AS values returned by the *RouteViews* dataset were inserted in V'. Similarly, for all logical connections between IP address pairs in E, we obtained the corresponding logical connections between ASes using the *RouteViews* dataset. The logical connections between ASes were inserted in E'. In summary, using the *RouteViews* dataset, we mapped all IP addresses in G to their corresponding ASes in the *RouteViews* dataset to obtain G'. Algorithm 1 presents the formal procedure of obtaining the physical topology from the overlay topology.

Due to the biased distribution of nodes across ASes (Table 1), the number of vertices in the physical topology was less than the overlay topology ($|V'| << |V|$). In Figure 9, we plot the physical topology obtained from IP to AS mapping (Algorithm 1). In the physical topology, we observed 175 vertices and 771 edges. AS-16509 had the highest node degree of 118.

Routing Paths. After obtaining the logical connections between ASes that host Ripple nodes, we used the *BGPStream* API [4] to identify the on-path ASes that may intercept the communication among the Ripple nodes. For that purpose, we used the Routing Information Base (RIB) dataset in *BGPStream* API to collect up to 50K AS paths, where at least two ASes in V' were present on the path. As a result, we collected a total of 50K possible paths between any two ASes in V'. For measurement accuracy, we ensured that the RIB dataset date matched the date of Ripple network snapshot in our dataset. Due to time constraints, we restricted our analysis to 50K AS paths, as it provided sufficient data for subsequent analysis of routing path anomalies. In Algorithm 2, we provide the procedure for obtaining AS paths from the physical topology.

6.2. Measuring AS Dependency

Recent works on network security [6], [20] show that ASes (including Tier-1 ASes) can be hijacked if another on-path AS has a weak dependency on them. For the purpose of measuring the AS dependency, we used the AS hegemony score, which is a recent metric that determines the significance of an on-path AS based on its location in the path. The AS hegemony concept was introduced by the Internet Health Report (IHR), which also provides an API to calculate the hegemony scores of ASes [11]. The hegemony scores are scaled between 0–1, with 0 indicating a low dependency, and 1 indicating a high dependency.

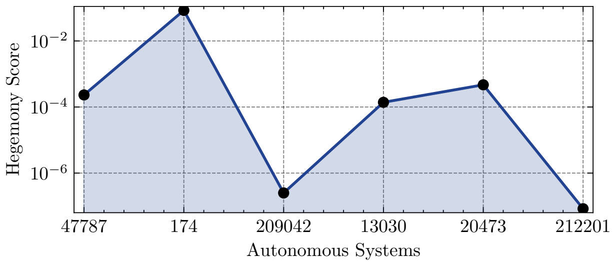


Figure 10. Hegemony scores of ASes on a routing path. The *valley* in the path shows that AS-209042, with a low global hegemony score, is located between two transit ASes (AS-174 and AS-13030) with high global hegemony scores. A *valley* in the AS path is considered an anomaly, and such anomalies have been linked to BGP hijacks [6].

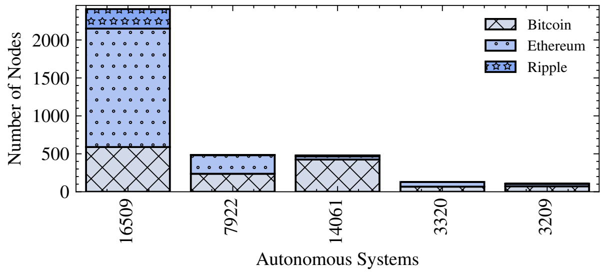


Figure 11. Top five ASes vulnerable to the practical attack, along with the nodes they host from each cryptocurrency network. Among all ASes, three high-profile ASes are AS-16509, AS-7922, and AS-14061.

In [6], Cho *et al.* applied the concept of AS hegemony score and discovered a correlation between BGP attacks and the hegemony score of on-path ASes. Their analysis showed that in an ideal configuration, the global AS hegemony score values must have only one local maximum in the middle of the path, indicating that transit ASes are located in the middle. However, if there is a local minimum between two local maxima (also called a *valley* in [6]), the path has an anomaly, and the AS corresponding to the local minimum can potentially hijack the prefixes of other on-path ASes.

To elaborate on the concept of AS hegemony and the practical attacks, consider the routing path shown in Figure 10. The path consists of six ASes (AS-47787—AS-174—AS-209042—AS-13030—AS-20473—AS-212201), obtained from *BGPStream* along with their global hegemony scores obtained from IHR [11]. The plot presents a local minimum between two local maxima, showing that an ASes with a low global hegemony score is present in the middle of two ASes with high global hegemony scores. Cho *et al.* [6] classified such *valleys* as anomalies in the routing paths and also linked those anomalies to the BGP attacks in the wild.

Following the approach in [6], we used the routing paths obtained in Algorithm 2 and measured the dependency of on-path ASes on the ASes that host cryptocurrency nodes. We marked vulnerable paths upon observing (1) a local minimum between two local maxima, (2) a change of more than 90% in the hegemony score, and (3) no cryptocurrency node being hosted in the AS causing the local minimum.

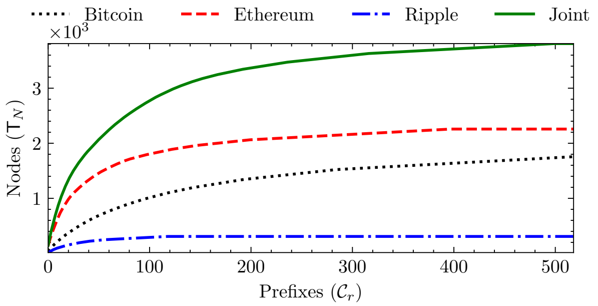


Figure 12. Sorted prefix-wise distribution of the cryptocurrency networks, which can be exploited to launch the practical attack. Adversaries can isolate 1,756, 2,258, and 308 Bitcoin, Ethereum, and Ripple nodes by hijacking up to 519, 395, and 116 prefixes, respectively. The joint network analysis of all three cryptocurrencies is shown in the green line. For ease of understanding, we extend the plot line on the x-axis after the total number of nodes in each network is isolated.

The last condition alleviates the attack's impact on the adversary. If the adversary is hosting cryptocurrency nodes, the practical attack will inevitably isolate them from the rest of the network. Since the adversary may want to avoid being a victim of its own attack, we assume that the adversary does not host any cryptocurrency node. In summary, the practical attack adversary is an on-path AS that does not host a cryptocurrency node and enjoys a low dependency on the victim ASes that host cryptocurrency nodes.

Our results revealed that a practical attack can be launched on 64 ASes that host cryptocurrency nodes from all three networks. Among the vulnerable ASes, we found high-profile ASes including AS-16509, AS-7922, and AS-14061. In Figure 11, we plot the top five ASes vulnerable to the practical attack. Figure 11 also shows the number of Bitcoin, Ethereum, and Ripple nodes hosted by those ASes at the time of conducting the experiment. We found that a total of 4,322 cryptocurrency nodes could be isolated through the practical attack, including 1,756, 2,258, and 308 Bitcoin, Ethereum, and Ripple nodes, respectively.

Taking into account the number of nodes that can be isolated, it is important to understand the key differences between classical and practical attacks. In the classical attack, the adversary can target all ASes and isolate their cryptocurrency nodes. However, as discussed in §4, the classical attack adversary may face the attack consequences due to its dependency on the victim ASes. The risk of launching an attack may outweigh the benefit. In contrast, the practical attack adversary only targets the ASes on which it has a low dependency. As a result, the number of potential victims is a subset of ASes among all ASes, and the number of vulnerable nodes is less than the total number of cryptocurrency nodes. From an adversarial standpoint, if the vulnerable ASes are high-profile ASes hosting a large number of nodes, the attack can be widespread. Our results demonstrate that three out of the top ten ASes in Table 1 are vulnerable to practical attack, thereby creating favorable outcomes for the adversary. In the following section, we provide more details regarding the practical attacks on the cryptocurrency networks.

6.3. The Bitcoin Network

For the Bitcoin network analysis, we assume that $\mathcal{A}_b$ follows the attack procedure from the previous section and identifies all vulnerable ASes. Among them, $\mathcal{A}_b$ selects the ASes that host at least one Bitcoin node. $\mathcal{A}_b$ then announces more specific prefixes than the victim AS to hijack its traffic and isolate the Bitcoin nodes. A low dependency on the victim AS allows $\mathcal{A}_b$ to feasibly target them.

Among the 64 vulnerable ASes identified in our experiment, we found 48 ASes that hosted at least one Bitcoin node and a total of 1,756 nodes. Among those ASes, AS-16509 hosted more than 30% of those nodes. In terms of the prefix-wise distribution, all 1,756 nodes were hosted across 519 unique prefixes, with 76 prefixes hosting 50% of the nodes. In Figure 12, we plot the sorted prefix-wise distribution of the Bitcoin nodes across prefixes which can be exploited by $\mathcal{A}_b$ to launch the practical attack.

6.4. The Ethereum Network

For the Ethereum network analysis, we assume that $\mathcal{A}_e$ identifies all vulnerable ASes in the practical attack and selects ASes that host at least one Ethereum node. In our experiments, we found 42 ASes that hosted at least one Ethereum node and a total of 2,258 Ethereum nodes. Among those ASes, AS-16509 hosted more than 60% of those nodes. In terms of prefix-wise distribution, all 2,258 nodes were hosted across 395 unique prefixes, with 27 prefixes hosting 50% of the nodes.

In Figure 12, we provide the sorted prefix-wise distribution of the Ethereum nodes, which can be exploited by $\mathcal{A}_e$ to launch a practical attack. Figure 12 also shows that the Bitcoin nodes are more distributed across prefixes than the Ethereum nodes. Therefore, for the same number of nodes to be isolated, the attack cost in Bitcoin is higher than in Ethereum. The key factor for the cost difference is the prefix-wise distribution of the Bitcoin and Ethereum nodes in AS-16509. As shown in Table 1, AS-16509 hosts more Ethereum nodes than Bitcoin nodes. Moreover, the prefix-wise node distribution of Ethereum nodes is more centralized in AS-16509 than the Bitcoin nodes.

6.5. The Ripple Network

Among the 64 vulnerable ASes identified in §6.2, we found 20 ASes that hosted at least one Ripple node. Combined, all 20 ASes hosted 308 Ripple nodes, with AS-16509 hosting more than 80% of those nodes. In terms of the prefix-wise distribution, all 308 nodes were hosted across 116 prefixes, with 20 prefixes hosting 50% of the nodes.

In Figure 12, we provide the sorted prefix-wise distribution of the Ripple nodes which can be exploited by $\mathcal{A}_r$ to launch the practical attack. Note that 308 nodes make ≈38% of the Ripple network. Therefore, by hijacking only 116 prefixes, $\mathcal{A}_r$ can partition ≈38% of the Ripple network. The scale of vulnerability shows that Ripple is the most vulnerable of all cryptocurrencies to practical attacks.

6.6. Joint Network Analysis

After the standalone analysis of each network, we conducted a joint analysis based on network interdependence. We found 9 ASes that hosted at least one node from each cryptocurrency network and a total of 3,810 nodes (1,486 Bitcoin, 2,022 Ethereum, and 302 Ripple nodes). AS-16509 hosted more than 60% of those nodes. All nodes were hosted across 497 unique prefixes, with 42 prefixes hosting 50% of the nodes. In Figure 12, we plot the prefix-wise distribution for the joint network analysis.

In our experiments, we also observed that nodes from different cryptocurrencies also share the same set of AS prefixes. More precisely, among the total 497 unique prefixes, 79 prefixes hosted nodes from all three networks. Prefix sharing by nodes from different cryptocurrency networks allows $\mathcal{A}_g$ to isolate the targeted number of nodes by announcing fewer prefixes.

6.7. BGP Attacks in the Wild

Our practical attack analysis exposed anomalies in the routing paths of high-profile ASes. It is therefore pertinent to study if adversaries are already exploiting the routing path anomalies to target them. Empirical evidence for BGP attacks in the wild is necessary for the following two reasons. First, as discussed in §1, it is commonly argued that AS dependency provides inherent protection to high-profile ASes against BGP hijacks. After observing anomalies in AS dependency (§6.2), an empirical evaluation can support our evaluation methodology. Second, it is also considered that RPKI adoption by ASes protects them from BGP hijacks. Recently, notable ASes that host cryptocurrency nodes including AS-701, AS-7018, AS-16509, and AS-7922 have adopted RPKI. As such, if ASes hosting cryptocurrency nodes are targeted despite strong security measures, it creates a need for an additional set of countermeasures against practical partitioning attacks.

For the real-world attack analysis, we collected a dataset of BGP outages and hijacks from April 07, 2022, to November 01, 2022. The dataset was obtained from Cisco's BGP monitors that record anomalous BGP activities [3]. From Cisco's BGP monitors, we collected 8,639 events, including 4,171 outages and 4,468 hijacks. Note that an outage could be due to misconfigurations, and it does not involve any prefix announcement. In a hijack, however, another AS announces the prefixes of a target AS to re-route the traffic. In our analysis, we only focused on the hijacks and found 246 hijack events that targeted 106 ASes hosting cryptocurrency nodes. Moreover, 17 hijack events targeted seven out of the top ten ASes that hosted cryptocurrency nodes. The targeted ASes included AS-701, AS-16509, AS-14061, AS7922, and AS-16276. Despite RPKI adoption, AS-701 and AS-16509 were targeted 5 and 3 times, respectively. Our analysis confirms that routing path anomalies can be exploited in the real world to target the high-profile ASes.

6.8. Key Takeaways

The practical attack analysis significantly advanced our knowledge and understanding of the risks poised by spatial partitioning attacks on cryptocurrency networks. From our analysis in §6, we made the following key conclusions. (1) Routing paths exposed by the Ripple topology reveal anomalies in the physical network that can be exploited to launch practical attacks against ASes that host nodes from the three cryptocurrency networks. (2) There exist anomalies in the routing paths of high-profile ASes that host a large number of cryptocurrency nodes. (3) With the growth of existing cryptocurrencies and the deployment of new cryptocurrencies, routing path anomalies and network interdependence must be carefully considered as an important security risk. (4) Our real-world analysis of BGP hijacks confirms that high-profile ASes can be targeted despite the adoption of new security policies. [10]

7. Attack Implications and Countermeasures

In this section, we evaluate the spatial partitioning attack implications using discrete-event simulations, followed by attack countermeasures. As an example, we use the Bitcoin network for our analysis and evaluation.

Bitcoin nodes establish outgoing connections to other Bitcoin nodes and receive incoming connections from them. The logical connections between nodes form an overlay topology, allowing nodes to exchange blocks and maintain a consistent blockchain ledger as long as a cycle exists in the overlay. Bitcoin nodes establish ten outgoing connections, and if the connection count drops, they attempt new connections to IP addresses stored in their IP tables [27]. When a spatial partitioning attack is launched, nodes in the victim AS lose their outgoing connections to the nodes outside their AS. Eventually, they may complete their outgoing connection slots by connecting to the nodes within their AS. As a result, all newly mined blocks by nodes in the victim AS are only shared with other nodes in the same AS.

If an adversary hijacks ten ASes and the partitioned nodes start mining blocks, the blockchain is forked into ten branches, with each branch being extended in the hijacked AS. When the ASes recover from the hijack, nodes can start establishing connections outside their ASes to share newly mined blocks, including forked branches. After receiving the forked branches, nodes apply the longest chain rule to resolve forks [22]. The longest chain rule is a fundamental principle applied in Bitcoin to resolve forks by adopting a chain that represents the maximum effort in the form of proof-of-work. Therefore, if a branch in one AS is longer than the branches mined in the other nine ASes, the longest branch will be adopted by all nodes and the shorter branches will be discarded. In Figure 13, we illustrate blockchain

[10] Our attack evaluations are based on a single-day snapshot of each network. Since cryptocurrency networks are permissionless, their node distribution may change over time which might show varying results depending on the data collection and analysis timeline. Nevertheless, the key aspects of node centralization, *network interdependence*, and routing path anomalies are generalizable and likely to be found across all observations.

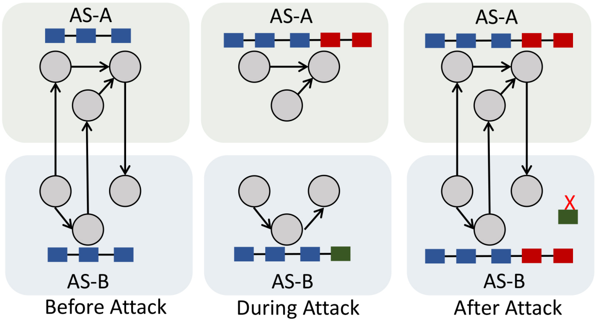


Figure 13. An illustration of blockchain fork due to spatial partitioning attack. Nodes in AS-A and AS-B are connected to each other and maintain a common ledger. During the attack, nodes in each AS are isolated from the rest of the network. Blockchain is forked in both ASes as nodes mine new blocks. Eventually, the fork resolves after the attack, and connections are re-established. Since nodes in AS-A have a longer chain, all nodes adopt the longer chain while discarding the block mined in AS-B.

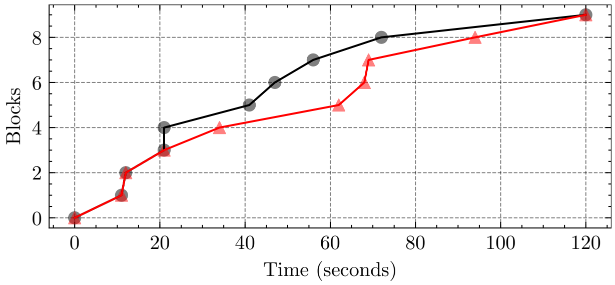


Figure 14. Simulation results for the spatial partitioning attack. The x-axis shows the time from the first mined block, and the y-axis shows the number of blocks mined. The attack was executed at the third block, after which the blockchain forked. The forked branches are shown in black and red colors. As one branch became longer than the other branch, we removed the partitioning after ten blocks to resolve the fork.

forks and fork resolution during and after the spatial partitioning attack. Note that forks allow the adversary to split the mining power among nodes, thereby allowing the adversary to increase the transaction confirmation delay, invalidate legitimate transactions, or double-spend transactions [27], [37], [14], [24].

7.1. Attack Simulation

In this section, we present experiments that demonstrate how spatial partitioning attacks lead to forks. In following the ethical standards, we did not conduct our experiment on the real-world Bitcoin network and instead created a small-scale simulation for an attack demonstration.

Our simulation environment consisted of 16 nodes running as processes in a virtual machine. Each process was assigned a port number through which it communicated messages. We closely modeled the Bitcoin mining protocol by allowing nodes to compute random hashes that matched the predefined target requirement. The matching hash value is considered a valid proof-of-work solution [22], [26]. When a node found a valid solution, it relayed the solution to other nodes in the network. After receiving the solution, all nodes included it in their local blockchain.

Algorithm 3: Practical Attacks Countermeasures

```
Input: Bitcoin Node ₿ running RPC
Initialize: Hegemony List $\mathsf{H}_L$
₿ executes getpeerinfo and receives a set of P peers.
foreach $p_i \in \mathsf{P}$ do
    ₿ executes traceroute command and obtains a routing path
      consisting of ASes A
    foreach $a_i \in \mathsf{A}$ do
        ₿ calculates AS hegemony score $h_i$using IHR dataset, and
          appends $h_i$ to $\mathsf{H}_L$
    foreach $h_i$ and $h_k \in \mathsf{H}_L$ do
        ₿ finds $h_j$ between $h_i$ and $h_k$ where $h_j < h_i$ and
          $h_j < h_k$ // Traverse $\mathsf{H}_L$ and find smaller
          values between two larger values
        ₿ calculates $\Delta_1 = h_i - h_j$ and $\Delta_2 = h_k - h_j$.
        if $\left(\frac{\Delta_1}{h_j} + \frac{\Delta_2}{h_k}\right)/2 \geq 0.9$ then
            ₿ removes connection with $p_i$
            ₿ automatically attempts a new outbound connection to
              complete outgoing slots
        else
            ₿ maintains connection with $p_i$
```

At the start of the simulation, we allowed all nodes to mine blocks, relay those blocks to the other nodes, and maintain a consistent blockchain ledger. After three blocks were mined, we launched the attack and partitioned the nodes into two groups, each consisting of eight nodes. Our attack procedure was similar to the illustration shown in Figure 13, where nodes were divided into two groups in AS-A and AS-B, respectively. As a result of the partitioning, the blockchain forked into two branches in both groups. We continued the attack until one branch became longer than the other branch and mined ten blocks. We then removed the partitioning to allow communication between the two groups. When the longer branch was released, the fork was resolved and the longer branch was adopted by all nodes.

We plot our simulation results in Figure 14. The x-axis shows the time from the first mined block, and the y-axis shows the number of blocks mined by the nodes. For the first three blocks, all nodes had a consistent blockchain ledger with no forks. After the third block, a partitioning was created between the two groups, which resulted in a fork. We observed that one of the branches extended faster than the other branch due to the proof-of-work randomness. When the branch mined a total of ten blocks, we removed the partitioning to resolve the fork. After the fork was resolved, the longest branch was adopted by all nodes in the network.

7.2. Attack Countermeasures

Our analysis so far shows that the spatial partitioning attacks pose a major threat to interdependent cryptocurrency networks. Among the classical and practical attacks described in this paper, the practical attacks are of significance, since they expose routing path anomalies in high-profile ASes, which cannot be ignored despite existing countermeasures in place. In this section, we propose new countermeasures to the spatial partitioning attacks and evaluate them in the Bitcoin network.

In essence, the practical attack can be countered if ASes avoid insecure paths for traffic routing, or the cryptocurrency users migrate their nodes to other ASes in order to minimize network interdependence. However, both of these approaches are infeasible in practice due to the following reasons. First, we do not know why vulnerable ASes are following the current routing policies, despite being vulnerable to attacks. Therefore, we cannot expect them to change those policies simply to provide extra protection to cryptocurrency nodes. Second, due to the underlying incentive for node hosting, it is difficult to expect users to migrate their nodes to other (and possibly more expensive) cloud operators. Finally, based on our analysis in §6.7, relying simply on ASes to upgrade their security standards using RPKI may be insufficient to prevent partitioning attacks.

Acknowledging these challenges, we developed application layer defenses to protect users from practical attacks without requiring ASes to change their routing policies. Notice that in the practical attack, the adversary maps the overlay topology onto the physical topology, and only launches an attack if it exists on an insecure routing path with a low AS dependency characterized by the global hegemony score. As such, if the overlay topology is structured in such a way that the number of insecure paths is reduced, we can limit the attack options available to the adversary.

For instance, the vulnerable path shown in Figure 10 (AS-47787—AS-174—AS-209042—AS-13030—AS-20473—AS-212201) is revealed due to a connection between two Ripple nodes. The adversary exploits this knowledge because it can map the overlay topology onto the physical topology and compute the AS dependency. To construct our countermeasures, we suggest that such computations can also be performed by the cryptocurrency nodes. For instance, when two Bitcoin nodes establish a connection, they can launch a *traceroute* to discover all the on-path ASes that intercept the traffic and also compute their AS dependency by using IHR's API or the corresponding data dumps [11]. If they discover a routing path anomaly, they can disconnect and establish a new connection with other nodes while repeating the process. Eventually, they can complete their outbound connection slots by connecting to only those nodes that exist on secure routing paths with no anomalies. As a result of these overlay topology changes, users can reduce the number of paths that can be exploited by the adversary to launch the attack. More importantly, if the adversary exists on the insecure path, the proposed technique will prevent the adversary from learning that it intercepts traffic between cryptocurrency nodes on that path. This approach can enhance the security of cryptocurrency networks by limiting the knowledge required for launching the practical attack.

In Algorithm 3, we provide the countermeasure methodology which allows a node to determine the security of the routing path with each of its connections. The node first determines the AS path using *traceroute*, followed by the hegemony score calculation. If the change in the hegemony score is greater than 90%, the connection is removed, and a new connection is established. As a result, all connections on insecure routing paths are replaced by connections on relatively secure routing paths. Note that the proposed technique does not require ASes to change their routing policies or users to migrate their nodes to other cloud operators.

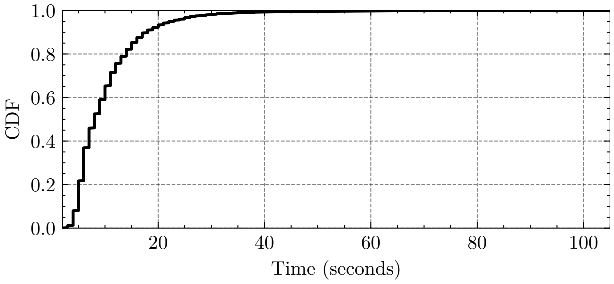


Figure 15. CDF of the computation time taken by our node to calculate the routing path security for 5,070 connections. Our node evaluated ≈92% of the paths in less than 20 seconds.

Experiment Evaluation. We evaluated the effectiveness of our proposed countermeasures by deploying Algorithm 3 on a Bitcoin node and analyzing its performance in the Bitcoin network. To test our methodology at a large scale, we made some modifications at our node. We changed the Bitcoin source code to enable more than 10 outgoing connections. We then collected 5,488 IP addresses of Bitcoin nodes from Bitnodes [8], and connected to them using our node. We then applied procedures outlined in Algorithm 3 to examine the security of AS paths between our node and each of its connections. For benchmark evaluation, we recorded the time it took for our node to determine the security of a routing path. For instance, if a connection was established at time t_1 and the decision to disconnect (or stay connected) was made at time t_2, then $t_2 - t_1$ is the time taken by the node to determine the routing path security.

Among the 5,488 IP addresses obtained from Bitnodes, we successfully connected with 5,070 nodes and evaluated the routing path security. We found 105 connected nodes on potentially insecure routing paths and initiated a disconnect request using the *disconnectnode* RPC API [7]. For all 5,070 connections, our node took an average of 10.19 seconds to compute the routing path security, with a standard deviation of 7.2 seconds. Figure 15 plots the CDF of the computation time taken by our node to calculate the routing path security of 5,070 connections. We found that our node evaluated ≈92% of the paths in less than 20 seconds.[11] Our countermeasures evaluation confirms that routing path anomalies exist in real-world cryptocurrency networks. Moreover, our proposed approach can be deployed by nodes to efficiently determine the routing path security and disconnect from nodes on potentially insecure routing paths.

8. Conclusion

In this paper, we comprehensively analyze the spatial partitioning attacks on three popular cryptocurrency networks; Bitcoin, Ethereum, and Ripple. We uncover an increasingly biased distribution of cryptocurrency nodes across ASes, which puts them at a high risk of BGP attacks. We also show that cryptocurrency networks exhibit a strong

[11] The routing path evaluation time may vary depending upon the location of connected nodes and their network bandwidth.

network interdependence by sharing the same ASes, which amplifies the effect of the spatial partitioning attacks.

An essential contribution of this work is the practical attack discovery made by mapping the overlay topology to the physical cryptocurrency network's topology and identifying insecure routing paths. We also discover irregularities in the routing paths of high-profile ASes, which can be exploited to paralyze all three cryptocurrency networks. Acknowledging this threat, we develop countermeasures for practical attacks that cryptocurrency users can effectively deploy.

References

[1] M. Apostolaki, G. Marti, J. Müller, and L. Vanbever, "SABRE: protecting bitcoin against routing attacks," in *Annual Network and Distributed System Security Symposium, NDSS, San Diego, California, USA*, Feb 2019. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/sabre-protecting-bitcoin-against-routing-attacks/

[2] 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.

[3] BGPStream, "Cisco crosswork cloud." [Online]. Available: https://crosswork.cisco.com/

[4] CAIDA, "Bgpstream," https://bgpstream.caida.org/, 2021, (Accessed on 03/26/2021).

[5] ——, "Caida prefix-to-autonomous system (as) mappings," https://publicdata.caida.org/datasets/routing/, 2022, (Accessed on 07/29/2022).

[6] S. Cho, R. Fontugne, K. Cho, A. Dainotti, and P. Gill, "BGP hijacking classification," in *Network Traffic Measurement and Analysis Conference*, 2019, pp. 25–32. [Online]. Available: https://doi.org/10.23919/TMA.2019.8784511

[7] B. Community, "Disconnectnode." [Online]. Available: https://developer.bitcoin.org/reference/rpc/disconnectnode.html

[8] ——, "Bitnodes: Discovering all reachable nodes in bitcoin," 2021. [Online]. Available: https://bitnodes.earn.com/

[9] C. Community, "Cryptocurrency market capitalization," 2022. [Online]. Available: https://coinmarketcap.com/

[10] E. Community, "Ethernodes: Ethereum mainnet statistics," https://www.ethernodes.org/nodes, 2020, (Accessed on 02/23/2021).

[11] I. Community, "Internet health report," https://ihr.iijlab.net/ihr/en-us/documentation/AS_dependency, 2020, (Accessed on 03/23/2021).

[12] R. Community, "Ripple peer crawler," https://xrpl.org/peer-crawler.html, 2021, (Accessed on 03/23/2021).

[13] ——, "Routeviews – university of oregon route views project," http://www.routeviews.org/routeviews/, 2020, (Accessed on 02/23/2021).

[14] D. Dasgupta, J. M. Shrein, and K. D. Gupta, "A survey of blockchain from security perspective," *J. Bank. Financial Technol.*, vol. 3, no. 1, pp. 1–17, 2019. [Online]. Available: https://doi.org/10.1007/s42786-018-00002-6

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

[16] S. Delgado-Segura, S. Bakshi, C. Pérez-Solà, J. Litton, A. Pachulski, A. Miller, and B. Bhattacharjee, "Txprobe: Discovering bitcoin's network topology using orphan transactions," *CoRR*, vol. abs/1812.00942, 2018. [Online]. Available: http://arxiv.org/abs/1812.00942

[17] P. K. Dey, M. A. Canbaz, M. Yuksel, and M. H. Gunes, "On correlating ISP topologies to their businesses," in *International Conference on Communications, ICC Kansas City, MO, USA*. IEEE, 2018, pp. 1–7. [Online]. Available: https://doi.org/10.1109/ICC.2018.8422620

[18] P. Ekparinya, V. Gramoli, and G. Jourjon, "Impact of man-in-the-middle attacks on ethereum," in *IEEE Symposium on Reliable Distributed Systems, SRDS, Salvador, Brazil*. IEEE Computer Society, Oct 2018, pp. 11–20. [Online]. Available: https://doi.org/10.1109/SRDS.2018.00012

[19] W. Fan, S. Chang, X. Zhou, and S. Xu, "Conman: A connection manipulation-based attack against bitcoin networking," in *Conference on Communications and Network Security*. IEEE, 2021, pp. 101–109. [Online]. Available: https://doi.org/10.1109/CNS53000.2021.9705018

[20] R. Fontugne, A. Shah, and E. Aben, "AS hegemony: A robust metric for AS centrality," in *Special Interest Group on Data Communication*, 2017, pp. 48–50. [Online]. Available: https://doi.org/10.1145/3123878.3131982

[21] ——, "The (thin) bridges of AS connectivity: Measuring dependency using AS hegemony," in *Passive and Active Measurement*, 2018, pp. 216–227. [Online]. Available: https://doi.org/10.1007/978-3-319-76481-8_16

[22] J. A. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol with chains of variable difficulty," in *International Cryptology Conference on Advances in Cryptology , Santa Barbara, CA, USA*, Aug 2017, pp. 291–323. [Online]. Available: https://doi.org/10.1007/978-3-319-63688-7_10

[23] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, and E. G. Sirer, "Decentralization in bitcoin and ethereum networks," in *International Conference on Financial Cryptography and Data Security, Nieuwpoort, Curaçao*, ser. Lecture Notes in Computer Science, vol. 10957. Springer, Feb 2018, pp. 439–457. [Online]. Available: https://doi.org/10.1007/978-3-662-58387-6_24

[24] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Conference on Computer and Communications Security*, 2015, pp. 692–705. [Online]. Available: https://doi.org/10.1145/2810103.2813655

[25] E. Gregori, A. Improta, L. Lenzini, L. Rossi, and L. Sani, "BGP and inter-as economic relationships," in *International Conference on Networking, Valencia, Spain, pages = 54–67, year = May 2011, url = https://doi.org/10.1007/978-3-642-20798-3_5,*.

[26] C. Grunspan and R. Pérez-Marco, "The mathematics of bitcoin," *CoRR*, vol. abs/2003.00001, 2020. [Online]. Available: https://arxiv.org/abs/2003.00001

[27] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA*, August 2015, pp. 129–144. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman

[28] Z. Huang, C. Wang, A. Nayak, and I. Stojmenovic, "Small cluster in cyber physical systems: Network topology, interdependence and cascading failures," *IEEE Trans. Parallel Distributed Syst.*, vol. 26, no. 8, pp. 2340–2351, 2015. [Online]. Available: https://doi.org/10.1109/TPDS.2014.2342740

[29] S. K. Kim, Z. Ma, S. Murali, J. Mason, A. Miller, and M. Bailey, "Measuring ethereum network peers," in *Internet Measurement Conference*. ACM, 2018, pp. 91–104. [Online]. Available: https://dl.acm.org/citation.cfm?id=3278542

[30] Y. Kwon, D. Kim, Y. Son, E. Y. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (FAW) attacks on bitcoin," in *SIGSAC Conference on Computer and Communications Security, CCS, Dallas, TX, USA*, Oct 2017, pp. 195–209. [Online]. Available: https://doi.org/10.1145/3133956.3134019

[31] K. Li, Y. Tang, J. Chen, Y. Wang, and X. Liu, "Toposhot: uncovering ethereum's network topology leveraging replacement transactions," in *Internet Measurement Conference*, D. Levin, A. Mislove, J. Amann, and M. Luckie, Eds. ACM, 2021, pp. 302–319. [Online]. Available: https://doi.org/10.1145/3487552.3487814

[32] Z. Li, J. Hou, H. Wang, C. Wang, C. Kang, and P. Fu, "Ethereum behavior analysis with netflow data," in *Asia-Pacific Network Operations and Management Symposium, APNOMS, Matsue, Japan*. IEEE, Sept 2019, pp. 1–6. [Online]. Available: https://doi.org/10.23919/APNOMS.2019.8893121

[33] B. Magazine, "Bitcoin and erebus attacks," https://bitcoinmagazine.com/technical/video-erebus-attacks-and-how-to-stop-them-with-asmap, 2020, (Accessed on 04/04/2021).

[34] Y. Marcus, E. Heilman, and S. Goldberg, "Low-resource eclipse attacks on ethereum's peer-to-peer network," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 236, 2018. [Online]. Available: http://eprint.iacr.org/2018/236

[35] S. B. Mariem, P. Casas, M. Romiti, B. Donnet, R. Stütz, and B. Haslhofer, "All that glitters is not bitcoin - unveiling the centralized nature of the BTC (IP) network," in *Network Operations and Management Symposium, Budapest, Hungary*. IEEE, 2020, pp. 1–9. [Online]. Available: https://doi.org/10.1109/NOMS47738.2020.9110354

[36] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Pisa, Italy*, L. Rizzo, T. E. Anderson, and N. McKeown, Eds. ACM, 2006, pp. 291–302. [Online]. Available: https://doi.org/10.1145/1159913.1159947

[37] M. Saad, V. Cook, L. Nguyen, M. T. Thai, and A. Mohaisen, "Partitioning attacks on bitcoin: Colliding space, time, and logic," in *IEEE International Conference on Distributed Computing Systems ICDCS, Dellas, Texas, US*, July 2019.

[38] M. Saad and D. Mohaisen, "Extended paper version," https://github.com/blkchnresearch/extended_paper, 2022, (Accessed on 12/10/2022).

[39] M. Tran, I. Choi, G. J. Moon, A. V. Vu, and M. S. Kang, "A Stealthier Partitioning Attack against Bitcoin Peer-to-Peer Network," in *Proceedings of IEEE Symposium on Security and Privacy (IEEE S&P)*, 2020.

[40] V. Tumas, S. Rivera, D. Magoni, and R. State, "Topology analysis of the XRP network," *CoRR*, vol. abs/2205.00869, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2205.00869

[41] P. Vervier, O. Thonnard, and M. Dacier, "Mind your blocks: On the stealthiness of malicious BGP hijacks," in *Annual Network and Distributed System Security Symposium, San Diego, California, USA*. The Internet Society, 2015. [Online]. Available: https://www.ndss-symposium.org/ndss2015/mind-your-blocks-stealthiness-malicious-bgp-hijacks