Resilience of Blockchain Overlay Networks

Aristodemos Paphitis[1], Nicolas Kourtellis[2], and Michael Sirivianos[1]

[1] Cyprus University of Technology, Limassol, Cyprus
am.paphitis@edu.cut.ac.cy, michael.sirivianos@cut.ac.cy
[2] Telefonica Research, Barcelona, Spain
nicolas.kourtellis@telefonica.com

Abstract. Blockchain (BC) systems are highly distributed peer-to-peer networks that offer an alternative to centralized services and promise robustness to coordinated attacks. However, the resilience and overall security of a BC system rests heavily on the structural properties of its underlying peer-to-peer overlay. Despite their success, critical design aspects, connectivity properties, and interdependencies of BC overlay networks are still poorly understood. In this work, our aim was to fill this gap by analyzing the topological resilience of seven distinct BC networks. In particular, we probed and crawled these BC networks for 28 days. We constructed, at frequent intervals, connectivity graphs for each BC network consisting of all potential connections between peers. We analyze the structural graph properties of these networks and their topological resilience. We show that by targeting fewer than 10 highly connected peers, major BCs such as Bitcoin can be partitioned into disconnected components. Finally, we uncover a hidden overlap between different BC networks, where certain peers participate in more than one BC network. Our findings have serious implications for the robustness of the overall ecosystem of the BC network.

Keywords: Blockchain · P2P Networks · Robustness.

1 Introduction

The success of Bitcoin has resulted in the emergence of numerous blockchains and cryptocurrencies, with more than 20,000 cryptocurrencies in existence as of 2023. The distinctive features of blockchain technology have enhanced its visibility and are expected to disrupt various sectors that traditionally rely on trusted centralized third parties. Due to their ability to decentralize trust and improve asset management [13], numerous blockchain solutions have been proposed for a wide range of use cases, including healthcare, advertising, insurance, copyright protection, energy, cybersecurity, and government [6,62,15,14].

Blockchains (BC) rely on decentralized peer-to-peer (P2P) networks for their operation. Peers need to constantly maintain a local copy of all transactions and blocks, so the availability of the P2P overlay is essential for blockchain-data propagation. Generally, the security and resilience of networks depend on the structure of the underlying topology. Despite the significant amount of research

on BC systems, the design and connectivity properties, as well as the interdependencies of BC networks, are not fully understood.

To develop secure and robust blockchain-based tools and infrastructure, it is crucial to examine the underlying P2P network of blockchains to identify potential limitations and vulnerabilities. Despite the security provided by proof-of-work consensus, attacks on the P2P network could weaken consensus in specific parts of the BC network. By analyzing and understanding the resilience of these networks, we can mitigate damage from both natural failures and targeted attacks.

Blockchains are already being used to process large amounts of money; considering their potential application in other aspects of everyday life, they become an attractive target for ill-intentioned attacks by malicious actors. Attackers can exploit network vulnerabilities to carry out various attacks on BC consensus and fairness [32]. Therefore, it is important to investigate whether small-scale attacks against a few nodes could provide attackers with a significant advantage.

Despite the rich literature on network resilience [43,2,7,47,2], the research community has not yet investigated the robustness properties of blockchain networks. In this paper, our aim is to fill this gap by providing a first look at the resilience of seven distinct blockchain overlays. In particular, we are interested in the partition tolerance of these networks. We present and discuss the results of our analysis based on the connectivity graphs that we have collected. Our analysis focused on several key aspects of blockchain overlays, including their resilience against random failures and targeted attacks, their spatial centralization within Autonomous Systems, and their interdependencies. We first present the results of our analysis on the partition tolerance of blockchain overlays against random failures and targeted attacks, examining how these types of disruption can affect the stability and reliability of the network. Next, we delve into the issue of spatial centralization in Autonomous Systems and its impact on network resilience, exploring the concentration of nodes within the same AS and its impact on network stability. Finally, we discuss their interdependencies, examining the interconnections among blockchain overlays through common peers and links.

2 Background and Related Work

Although the Bitcoin and Ethereum overlay networks have been thoroughly studied, their resilience against attacks has not been adequately assessed. We believe that this omission in the literature is mainly due to a lack of accurate knowledge of the underlying topology.

2.1 Selected overlay networks

In this section, we provide background information on the blockchain networks under study. Seven networks were chosen, all of which are consistently included in the top 50 cryptocurrencies by market capitalization, according to [16] for the past few years. We list them alphabetically: Bitcoin, BitcoinCash, Dash,

Dogecoin, Ethereum, Litecoin, and ZCash. With the exception of Ethereum, the aforementioned BCs are descendants of Bitcoin using very similar overlay implementations and node discovery protocols.

Bitcoin Overlay Network In the Bitcoin overlay network, nodes communicate through non-TLS TCP connections to form an unstructured P2P network. Bitcoin's security heavily depends on the global consistent state of the BC, which relies on its Proof-of-Work based consensus protocol. The communication protocol is briefly documented in [23], but there is no formal specification. To understand its subtleties, we looked into previous studies [9,48,36] and Bitcoin's official source code [22] (reference client). When a node joins the network for the first time, it queries a set of DNS seeds that are hardcoded in the reference client. The response to this lookup query includes one or more IP addresses of full nodes that can accept new incoming connections. Once connected to the network, a node receives unsolicited `addr` messages from its connected peers that contain IP addresses of other peers in the network. In addition, the client can send to peers `getaddr` messages to gather additional peers. The reply to a `getaddr` message may contain up to 1000 peer addresses. All known addresses are maintained in an in-memory data structure managed by the address manager(`ADDRMAN`), and are periodically dumped to disk, in the `peers.dat` file. This allows the client to connect directly to those peers on subsequent starts without having to use DNS seeds. When node A initiates a connection with peer B, it is considered an *outbound* connection for A and an *inbound* for B. The default Bitcoin parameters dictate 8 outbound connections and up to 117 inbound.

Ethereum Overlay network Ethereum's network communication comprises three distinct protocols, described in Ethereum's official documentation [27]. Node discovery in Ethereum is based on the Kademlia routing algorithm, a distributed hash table (DHT) [45]. In Ethereum, each peer has a unique 512-bit node ID. A bitwise XOR is used to compute the distance between two Node IDs. Nodes maintain 256 buckets, each containing a number of entries. Each node assigns known peers to a bucket, according to the XOR distance from itself. To find peers, a new node first adds a hard-coded set of bootstrap node IDs to its routing table. Then sends to these bootstrapping nodes a `FIND_NODE` message that specifies a random target node ID. Each peer responds with a list of 16 nodes from its own routing table that are closest to the requested target. Subsequently, the node tries to establish a number of connections (typically between 25 and 50) with other peers in the network and performs the node discovery procedure continuously.

2.2 Related Work

Arguably, the aspects of the network layer of blockchain systems have received much less attention than security and consensus [32]. Dotan *et al.* [26] recognize that blockchain overlay networks have different requirements than traditional

communication networks and observe that their fundamental design aspects are not well understood. Their work identifies differences and commonalities between blockchains and traditional networks and highlights open research challenges in network design for distributed decentralized systems.

Network measurements by Decker and Wattenhofer [19] have revealed that propagation delay is a critical parameter positively correlated with the appearance of blockchain forks. However, more recent studies indicate that Bitcoin's network infrastructure shows signs of improvement [28].

Gencer *et al.* where the first to point out that major cryptocurrencies face centralization issues [33]. A large fraction of reachable nodes are located in a handful of Autonomous Systems (AS). This opens the door for adversaries to launch network attacks at the Internet level by hijacking the BGP protocol [5]. Such attacks can isolate a large group of nodes from the rest of the network and introduce delays in message propagation. In fact, such attacks are becoming more sophisticated and are not easy to detect [55]. Additionally, less than five mining pools control the majority of hashing power. Furthermore, by combining knowledge of network topology and message distribution, researchers were able to identify highly influential nodes that have an advantage in block production and dissemination, strengthening centralization indications [8,34,25,29].

A large percentage of nodes that participate in the Bitcoin network are unreachable, making it difficult to accurately analyze their behavior and characteristics. However, previous research has shown that these unreachable nodes still play a significant role in the network. Wang and Pustogarov [58] found that a significant number of unreachable nodes propagate a large number of transactions and initiate a small number of connections to the reachable part of the network. The number of unreachable nodes is estimated to be between 10 and 100 times the number of reachable nodes [35]. These nodes have been found to have less secure wallets and initiate fewer connections to the reachable part of the network than the default bitcoin client. In general, understanding the behavior and characteristics of unreachable nodes in the Bitcoin network is important to improve our understanding of the network as a whole.

Numerous attack vectors, or methods that can compromise blockchain systems, have been proposed and analyzed in the literature [38,60,5,44,55,54]. Review articles have analyzed these attack vectors, highlighting how network attacks can be related to other types of attack and how the state of the network can facilitate the success of an attack. Such reviews [32] provide important information on the various ways networks can be targeted and the factors that can increase the probability of a successful attack. Understanding these attack vectors and their relationships to network conditions is crucial for developing effective defenses and countermeasures.

Accurate inference of the topology, or the arrangement and interconnection of nodes, in peer-to-peer (P2P) network overlays is a challenging problem that has yet to be fully solved. Although some research has successfully developed methodologies to accurately uncover the topology of Bitcoin and Ethereum networks [46,49,20,36,42], these approaches are often no longer applicable due to

changes in the protocol or the official Bitcoin client [21,51,59]. Additionally, some of these methods [20,42] have an prohibitive cost to execute due to transaction fees imposed by the network. Furthermore, very few of these works present network metrics, which could provide insights into the characteristics and properties of the network. Recently, the study by Paphitis *et al.* [52] has shed more light on the structural properties of blockchain overlay networks. Their findings suggest that major blockchains exhibit dissimilar structural characteristics and show signs of vulnerability to malicious attacks due to the presence of highly central nodes. In this work, we are specifically investigating the topological robustness of such networks and their tolerance against partitioning due to random failures and targeted attacks. Moreover, we investigate whether their spatial centralization in various Autonomous Systems, and their hidden interdependencies, could further facilitate such attacks.

3 Methodology

In order to study the resilience of blockchain P2P overlays, information on the structure of the networks is needed. This section explains our main idea, which bypasses the need for an accurate topology mapping of the network. We prove that this idea is well founded and we proceed to describe the methods we used to collect and validate data.

Topology inference in blockchain overlays is a challenging problem that has not yet been solved. Our approach is to solve a simpler problem while still being able to measure the structural robustness of these networks. Instead of trying to accurately capture existing connections between online nodes, which is almost impossible due to the design of blockchain networks, we focus on collecting all ***possible*** connections that may exist over a period of time. A connection between two nodes is considered possible if one node includes the other in its list of known addresses. Using this strategy, we trade accuracy for completeness and are able to synthesize connectivity graphs that include the vast majority of potential links between nodes. This method also captures actual connections, that is, all active links between nodes. Our main aim is to identify structural deficiencies in the overlays, and we believe that if the synthesized graph of all possible connections can be partitioned, then the actual realized topology of the overlay is likely to be partitionable as well. In our data collection, we do not differentiate between mining nodes or full nodes. We view all nodes as important contributors to the health of the system and as vital in the dissemination of transactions and blocks. If most of these nodes were partitioned, the blocks would not propagate in the P2P network, thus preventing network synchronization.

The goal of our data collection process is to capture the contents of `peer.dat` of every reachable peer in the network. This consists of the peer's *view of the network*, which contains all available peers to which it can connect. This is easily achieved by repeatedly asking peers for addresses they know of. To discover the nodes (peers) of the overlay networks, we modified the crawler maintained by the popular site *bitnodes.io* to meet our needs [10,61]. We added features that enable: a) crawling multiple chains using distinct processes; b) storing the mapping of

each node to its known-peers; c) and synchronizing the processes to dump the collected data for each blockchain at the same timestamp. Implementing an Ethereum crawler is substantially different since it uses a different protocol. The Ethereum crawler was built on the open source Trinity client [56] and all blockchain-related processing was disabled. We only implemented those parts of the protocols necessary to instantiate connections to Ethereum peers and participate in the discovery process.

3.1 Validation

A simple proof that the actual connection graph is not likely to be resilient when the synthesized one already is not, is provided here to further support our argument. As already described in the previous paragraphs, a synthesized graph G consists of all possible connections that could exist in the network. In this case, the actual graph R, which contains only the real links (active links between nodes), would be a spanning subgraph of G. A spanning subgraph is a subgraph that contains all the vertices (nodes) of the original graph but not all the edges (links). Our proposition is trivially proved considering Lemma 1 by Harary [37] which states the following: if R is a spanning subgraph of G, then the connectivity of R cannot be greater than the connectivity of G: $k(R) \leq k(G)$. That is, if G is disconnected i.e., $k(G) = 0$, then R is also disconnected. Thus, if the measured graph of possible connections can be partitioned by removing some nodes, then the actual graph will be partitioned as well.

Validation against controlled monitor To assess the viability of our goal, we set up an unmodified Bitcoin monitoring node using the official implementation [17]. We allowed the monitoring node to perform its initial bootstrap of the blockchain for one week. Subsequently, every ten minutes we retrieve the following information from the monitor: a) all inbound and outbound connections, b) a snapshot of the `peers.dat` file, and c) the `addr` reply to a `getaddr` probing message. We observe that by issuing enough `getaddr` messages, we are able to reconstruct the `peers.dat` file almost to its entirety.

During our validation period, the monitoring node created a total of 12,241 connections with other peers, 466 were outbound and 11,775 inbound. We observed 994 unique IP addresses, 368 corresponding to outbound connections, and 634 to inbound connections. Four of these addresses were in both sets. The crawler did not capture 444 of the 944 connected IP addresses. Looking at this weakness, we found that the missing IP addresses were not included in the `peers.dat` file. As expected, these were inbound connections from unreachable peers on the network. Further inspection revealed that most of these peers created short-lived connections that were dropped after the initial handshake. Only 30 of these peers (6% of inbound) created long-lasting connections of more than 40 minutes (a similar duration was used in [20]). Interestingly, the client version strings of these 30 nodes indicate that they were either network monitoring nodes (like `bitnodes.io`), experimental wallets, or client applications under development. We also observed a few client strings that have appeared in the

past and were identified as non-contributing nodes [31] by the community. If we exclude these non-contributing peers, the total number of unique IP addresses that the monitoring node connected to is 570 and our crawler missed 10 of them. The ten missing nodes correspond to a percentage of 1.75%. Furthermore, we analyzed the messages sent from these missing nodes to the monitoring node and noticed that all these nodes were far behind on their blockchain. Their most recent block was several days behind the latest block observed by the monitor.

Validation against against external data sources To further validate the coverage of our crawler against external data sources, we compared our results with the *DSN Bitcoin Monitoring* infrastructure in `https://www.dsn.kastel.kit.edu/bitcoin`, originally presented in [49]. Since the IP addresses of [49] are anonymized, we compare the number of reachable nodes we capture with the number of nodes scanned by the DSN Monitor. Counting only the reachable peers, we found that our crawler was able to retrieve a few hundred more nodes on a daily basis. Similarly, we compare the node counts with the historical data collected by a Bitcoin core developer [39] and the `bitnodes.io` crawler with similar results. We also note that although our data set is not very recent, comparing the number of peers collected to recent captures of the DSN Bitcoin Monitoring, we see that the size of the network has not changed significantly.

The previous paragraphs indicate that our method is adequate to create a network snapshot, capable of capturing all active connections that exist in the network, along with any potential connections that could be realized among the participating peers.

3.2 Datasets & Experiments

Using the methodology mentioned above, we crawled the selected BC networks from the datacenter of a European University. The monitoring server has an 8-core/3.2GHz CPU, 64GB RAM, and 2.1TB of HDD storage. The crawling operations were carried out for a period of about one month (26/06-22/07/2020). Previous work [18,19,41,46] used a similar duration for their analyzes. At the end of the crawling period, we had collected 335 network snapshots for each BC network; 2345 graphs in total. The collected data set is anonymously available for review at [4]. Our ethical considerations are outlined in App. A.

We denote by C the set of the 7 BC networks crawled. At the end of every two hours period, we have seven different `edge` sets, one per BC $c \in C$. At the end of each day, all edge sets belonging to the same network are merged into a 24-hour set. All sets are annotated with the date t of their crawl. Each set of edges corresponds to a graph, denoted S_c^t, representing a snapshot of the blockchain network c, on date t.

The following analysis uses the established procedure for the exploration of the resilience of a network [1,40]. The procedure starts by ranking the nodes by a network metric and subsequently removing the element in the network with the highest ranking. At each removal, the network is analyzed to calculate its remain-

ing size and the number of connected components. The most common node-level network metrics used are node degree and betweenness centrality [2,43].

4 Results

This section presents and discusses the results of the analysis performed on the synthesized graphs that were collected, focusing on several key aspects of BC overlays. The first aspect is the structural robustness of BC overlays to random failures and targeted attacks. The study examines the impact of these types of disruptions on the stability and reliability of the network. The next aspect is the issue of spatial centralization in Autonomous Systems and how it affects the network's resilience. This exploration includes an examination of the concentration of nodes within the same AS and how it impacts the network's stability. The study also investigates the interdependencies between BC networks, analyzing how these networks connected to other networks through peers and links.

4.1 Network Resilience to Attacks

This section answers the following question: To what extent are blockchain overlays prone to random failures and targeted attacks? We start this investigation by first describing the attack model. Then, we define three strategies that an attacker could employ to partition a BC network and evaluate the efficacy of each strategy. The practicality of the attack is beyond the scope of this work.

Attack Model. An adversary may have various incentives to attack a blockchain system. In this work, we specifically study attacks on the underlying topology of BC networks with the goal of impairing the main functions of the network. Specifically, we define the following two goals of an attacker:

1. Network partitioning: to force the overlay into two or more network partitions. A network partition is the decomposition of a network into independent subnets, so that no information flow between the partitions is possible due to the absence of links between nodes.
2. Disturb the information propagation mechanisms by introducing intolerable delays. Such delays can typically increase the time to reach consensus among all participants and create a split in the application layer of a BC system. In fact, propagation delays are known to be a key contributor to BC forks [19].

Such attacks would impair the main functions of a BC network, potentially causing a decrease in users' trust in the system. Attackers with external incentives would be highly motivated to carry out such attacks. To measure the effectiveness of each goal, we use the following three metrics: a) the size of the largest weakly connected component, b) the number of connected components, and c) the network diameter. To this end, we consider the following attack strategies:

1. Targeted attacks on unique nodes, based on a selected network metric. We test out-degree, betweenness centrality, and page-rank.

2. Random attacks using random node removal emulate failures that can occur in the network in a random fashion and are used as a baseline.
3. Attack minimum-cut edges, in order to partition the network by removing edges that are positioned in key places in the graph.

Targeted Node Attacks The removal of a node simultaneously cuts all its adjacent links, therefore, it is more efficient for an attacker compared to the removal of targeted links. We focus on how to remove nodes in the most efficient way to minimize the number of node removals necessary to cause a disruption. A node can be removed from the network by various means, including DoS attacks. We follow a static procedure in the sense that each node is given a static priority of removal, based on a chosen metric. For example, when using the out-degree metric, the higher the degree, the greater the importance of the node to be attacked. After removing a node, the priorities are not recalculated. We remove only **reachable** nodes from the network one by one, following the given priority. After each removal, we calculate the size of the largest connected component and the approximate diameter of the resulting graph. We report the effectiveness of the three node ranking metrics (betweenness centrality, out-degree, and page-rank), and compare with the baseline random removal strategy. We performed the procedure described on all 24-hour snapshots per BC. Due to the high number of graphs collected, we stopped execution after removing 12% of nodes per snapshot.

As can be seen in Figure 1, in Bitcoin and Bitcoin Cash, the betweenness and out-degree strategies have roughly the same shape. The size of the largest connected component decreases significantly after the removal of only a few nodes. Further removal of nodes gradually shrinks the size to a threshold where the connected component abruptly falls to 1% of its initial size. This occurs after the removal of 6% and 4% of the nodes, respectively. Similar behavior has also been found on the Internet [43]. This finding may not seem very worrisome, since the reported percentages correspond to a few thousand nodes. However, closer inspection (shown in Figure 3) of these two networks indicates that removal of the first five nodes reduces the size of the largest connected component by 60%, which is rather alarming. Unlike the size of the largest component, the diameter of the network starts to increase earlier in this process. This is more pronounced in Bitcoin Cash.

In Ethereum, the out-degree strategy is more potent. Unlike Bitcoin and Bitcoin Cash, the size of the largest component does not drop initially. After removing 2% of the nodes, the size gradually drops to a threshold, close to 5%, where its size abruptly drops to 1%. The diameter of the network starts to increase early, but not as quickly as in Bitcoin Cash.

When targeting high-betweenness nodes in Zcash, the largest component initially falls abruptly. Similarly to Bitcoin, the first removal of nodes reduces the largest component by 40%. When 4% of the nodes are removed, the largest component drops to 50% of its initial size and then shrinks almost linearly.

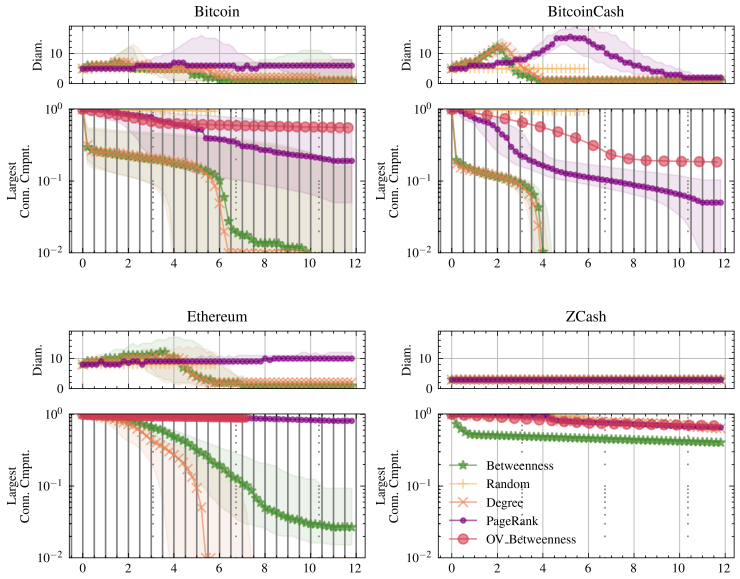

Fig. 1: Evolution of the approximate diameter (upper part) and size of largest weakly connected component (lower part) when the network in under targeted attack. The X-axis reports percentage of nodes removed. The lines correspond to the median value across all snapshots. The shaded area indicates values between 1st-3rd quartile. Orange x: Out-degree of unique nodes; Beige +: Random unique nodes; Green *: Betweenness of unique nodes; Red o: Betweenness of overlapping nodes;

Targeting high out-degree nodes is less damaging in Zcash. More than 5% of the nodes must be removed to observe a 20% reduction in the largest component.

The number of connected components for Bitcoin, BitcoinCash, and Ethereum during the same experiment is plotted in Figure 2. We cannot observe a notable rise in the number of components until the networks are significantly diminished.

Dash, Dogecoin, and Litecoin seem equally resilient to random and targeted attacks (plots omitted due to space limitations). The size of their largest component decreases linearly with the number of nodes removed, and their diameter is not significantly affected.

4.2 Attack minimum-cut edges

Targeting minimum cut edges does not have a significant effect in the networks' state and requires the removal (or disruption) of a considerable number of network links. To compute the minimum edge cuts, we used the algebraic connectivity of the derived graphs. The algebraic connectivity of a graph is defined as the second smallest eigenvalue of its Laplacian matrix L, $\lambda_2(L)$, and is a lower bound

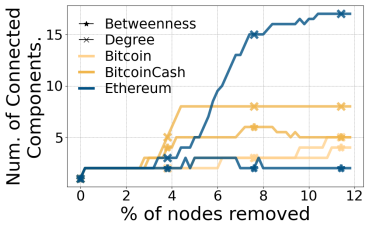

Fig. 2: Evolution of number of connected components during the same experiment as with Fig. 1

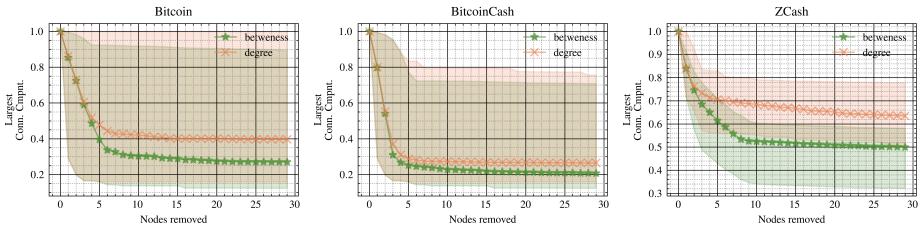

Fig. 3: Evolution of the largest weakly connected component when the network is under targeted attack. The difference with Fig. 1 is that this plot X-axis reports number of nodes removed. Orange x: Out-degree of unique nodes; Green *: Betweenness of unique nodes;

on node/edge connectivity [30]. Since calculating the algebraic connectivity of a graph is computationally very expensive (i.e., more than 3 compute hours per snapshot), we analyzed one snapshot per network. Using the computed eigenvector, we count how many edges are required to be removed to split the network in two parts, and compute their sizes and ratio of the two subnets (cut-ratio, computed as largest subnet over the total). The results are presented in Table 2. Most cuts are highly unbalanced. Bitcoin Cash has an almost perfect cut, although a large fraction of edges have to be removed (6.5% of edges or 10k edges). Bitcoin and Zcash are somewhat affected, by removing less than 0.5% of their network edges. Overall, targeting minimum cut edges does not have a significant effect on the networks' state and would require the removal (or disruption) of a considerable number of edges connecting nodes.

4.3 Spatial centralization of blockchain nodes

As already pointed out by previous works [5,54], BGP routing attacks can be mounted against Bitcoin by taking advantage of the fact that a high percentage of Bitcoin nodes reside in only a small number of Autonomous Systems (AS). We also verify this node centralization by mapping the collected IP address to ASes using the `https://ip-api.com` API. Furthermore, we were able to identify a single AS that hosts 20% of highly connected Bitcoin nodes in all timestamps,

Table 1: Resilience of graphs to targeted node attacks. We report the number and percentage of nodes that, when removed, reduce the largest component to 0.5 and 0.01 of its initial size, respectively.

Network	Bitcoin	Bitcoin Cash	Dash	Dogecoin	Ethereum	Litecoin	Zcash
# of Nodes (50% reduction)	10	10	-	-	300	-	6
% of Nodes (99% reduction)	6.5%	4%	>12%	>12%	5.5%	>12%	>12%

Table 2: Resilience of synthesized graphs in edge and node removal when attacking minimum-cut edges.

	Bitcoin	Bitcoin Cash	Dash	Dogecoin	Ethereum	Litecoin	Zcash
Edges Removed	5545 (0.1%)	10603 (6.5%)	1451 (0.02%)	581 (0.44%)	2220 (2.71%)	544 (0.08%)	363 (0.33%)
Network Split	9964/ 43949	11367/ 11895	46/ 8556	11/ 1069	436/ 15345	37/ 6576	258/ 1231
Cut Ratio	0.815	0.511	0.995	0.990	0.972	0.994	0.827

making it a strong candidate for such attacks. In more detail, we identify the 100 highest connected nodes in each snapshot. We then look at the distribution of these nodes in ASes. Our results are summarized below.

1. 20% of the top Bitcoin nodes are located in a single AS.
2. A single AS hosts a significant number of highly connected nodes in all BCs (see Sec. 4.4).
3. Ethereum's top clients are spread over more than 550 ASes and have the most wide distribution. Bitcoin nodes are spread in 200 ASes, BitcoinCash, Dash, Dogecoin in 160 and Zcash and Litecoin in 65.

To measure the effect of targeted attacks against Autonomous Systems, we performed the following test. For each snapshot, we identified the top 10 ASes with the highest geometric mean of out-degree. Then we simulated the effect of an attack against these ASes by removing all collocated nodes. The results are plotted in Figure 4. The blue dots correspond to the relative size of the largest connected component, on the left y-axis (median values across snapshots). The shaded area indicates values between the 1st and 3rd quartile. The yellow bars indicate the percentage of nodes removed (averaged) and correspond to the scale on the right y-axis.

Notably, these plots reveal the high centralization of BC nodes in the same Autonomous Systems, an observation made by previous works as well. Interestingly, all networks are sensitive to such attacks, mainly due to the centralization

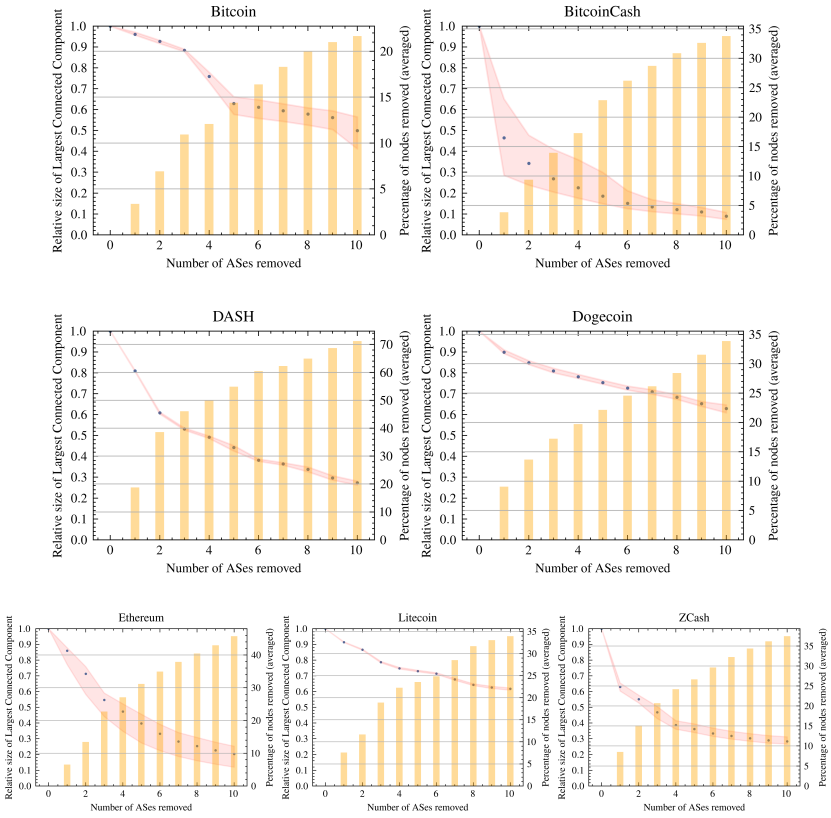

Fig. 4: Targeting selected ASed. X-axis reports number of ASes removed. The Y-axis on the left reports the size of the Largest Connected Component (blue dots). Right Y-axis reports the (average) percentage of nodes removed (yellow bars).

of nodes. This is true for DASH, Dogecoin, and Litecoin, where a single AS hosts 20% of each network's nodes. On the contrary, Bitcoin is less affected by this strategy (compared to attacking individual nodes), indicating that high-degree nodes are scattered in different ASes. Note that results may differ using a different selection strategy.

4.4 Dependency in Blockchain Overlays

In this section, we address the following questions: Are there network entities (peers, links) that participate in more than one BC network, concurrently? How do these common entities affect the resilience of overlay networks?

Chatzigiannis *et al.* [12] showed that miners can distribute their computational power over multiple pools and PoW cryptocurrencies to reduce risk and

Table 3: Edge and Node overlaps (aggregated). ON: number of networks where a unique entity (node or edge) was found to be overlapping, regardless of time

	ON=2	ON=3	ON=4	$ON >= 5$
Nodes	34814	3909	1489	779
Edges	143577	11034	1958	222

increase profits. Despite [12], there are no other indications that peers in BC systems participate in more than one cryptocurrencies at the same time. It would not come as a surprise to find that end users are present in multiple networks, however, this has not been observed or reported for participating peers so far.

We define as *overlapping nodes* those nodes that participate in more than one network at the same timestamp. The intuition of our analysis is as follows. In each snapshot, we compare the set of *overlapping* nodes with all other nodes, in order to draw insights on *overlapping* nodes' properties. Before describing the details of our study, we outline our mathematical notation to help explain our analysis. As mentioned above (cf. Section 3.2), C is the set of BCs. The notation S_c^t represents a snapshot of a blockchain network c, at t timestamp.

We define the set **S** as our collected data set, which consists of all snapshots S_c^t. We denote as S^t the subset of **S** that contains all networks at timestamp t. Subsequently, for each snapshot $S_c^t \in S^t$ we define two groups, G_c^t and $G_c^{'t}$, such that $G_c^t = S_c^t - G_c^{'t}$. The first set, G_c^t, is constructed so that $\forall$ nodes $n \in G_c^t, n \notin S_{C \setminus c}^t$. That is, the set G_c^t contains the nodes that participate only in blockchain c at timestamp t. On the contrary, the set $G_c^{'t}$ contains the *overlapping nodes*; those that participate in blockchain c **and at least another blockchain** $c^{'} \in C \setminus c$, at the same timestamp t.

A first approach to finding network overlaps is to look at our aggregated data set, **S**, and count how many nodes and edges (i.e., pairs of endpoints), appear in more than one network, regardless of time. Table 3 shows the summary of these results. Evidently, there exists a significant number of network entities (both nodes and edges) that reside in more than one BC network.

A second step is to investigate whether overlapping entities occur frequently or sporadically over time. For this, we count all overlapping peers in each S_c^t. In Figure 5 (left), for each BC network c, we plot the ratio of $|G_c^{'t}|$ over $|S_c^t|$, i.e., the number of overlapping peers in snapshot c over the total number of nodes in the snapshot. Our observations show that in all networks, there is a consistently high percentage of nodes that overlap and belong to more than one BC network. Based on this and previous results, we can conclude that there is significant overlap between BC overlays and that this overlap occurs consistently over time.

Attacking Overlapping Network Entities To examine how overlapping nodes could impact the resilience of blockchain overlays to targeted attacks, we repeat the test of the previous Section (4.1) with a small variation. From

each set $S^t_{c\in C}$, we remove all G^t_c sets. This new set, $S'^t_{c\in C}$, contains all nodes that participate in more than one network at the same timestamp. We then sort the unique elements of $S'^t_{c\in C}$ in descending order based on their maximum normalized betweenness centrality. Since a node can participate in more than one network, we sort the nodes based on the highest value they have across all networks at time t. We use the Min-Max method to normalize the betweenness centrality values for each snapshot. After sorting the nodes, we proceed to remove them from each snapshot S^t_c at the same time. The nodes are removed in the same order from all snapshots.

The results of targeting overlapping nodes first are plotted in Fig. 1 with red circles. The plot reports the average change in the largest connected component over all snapshots S^t_c. Clearly, this strategy is less effective compared to the strategies used earlier, which target the top central nodes within a specific network. However, it provides the benefit of attacking multiple networks simultaneously. An interesting finding is that Litecoin is more susceptible to this kind of attack compared to attacks focused on single BC node metrics (not shown in figure). This is partly explained by the fact that Litecoin has one of the highest percentages of overlapping nodes (see Fig. 5).

Closer inspection of the data at hand shows that an attacker is able to shrink the largest connected component of Bitcoin Cash, Bitcoin, Zcash, and Litecoin networks by 70%, 40%, 25% and 20% respectively. This demonstrates that by targeting overlapping nodes, a powerful adversary can still mount a successful partitioning attack in *4 different networks at the same time*.

Another effect of overlapping nodes is shown in the strategy described next. Similarly to the selection performed in Section 4.3, we calculated the geometric mean of the out-degree of all networks, for each AS, across all timestamps. That is, for each Autonomous System we took into consideration all nodes from all chains and calculated the geometric mean of their out-degree. We then remove each AS, simulating an attack against the AS, and calculate the effect on each network. Removal of an AS simultaneously removes all nodes (from all networks) that reside on that particular AS. The results of this selection strategy are plotted in Figure 5. The significance of overlapping nodes is profound. A disruption in just 6 ASes could have considerable effects in five networks at the same time. In fact, ASes do not need to be broken down; as demonstrated by Apostolaki*et al.* [5] they could be manipulated by false BGP routing advertisements (BGP hijacks). Notably, a different selection strategy would produce different results.

5 Discussion

Our results suggest that BC overlay networks are robust against random failures but weak against targeted attacks, a known characteristic of scale-free networks [2]. This further suggests that BC overlay networks are not random, contrary to their intended design [19]. These results are in line with those obtained by Miller *et al.* [46] and Delgado *et al.* [20]. Our analysis supports the findings of Paphitis *et al.* [52], which suggest that larger BC networks are more susceptible

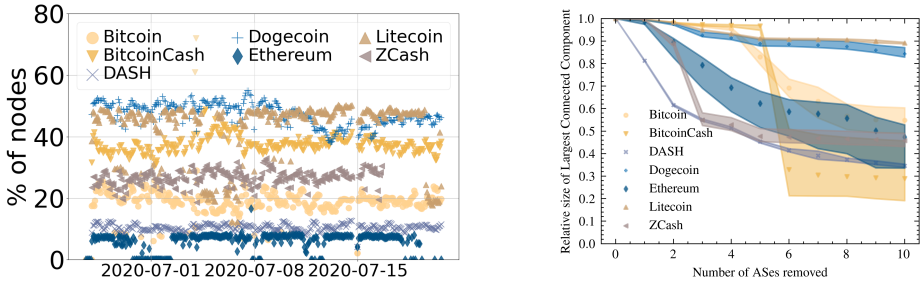

Fig. 5: *Left*: Percentage of nodes that were found in more than one BC network at the same timestamp. X-axis indicates the timestamp. *Right*: Size of the largest connected component of all networks when selected ASes are attacked.

to targeted attacks as a result of the presence of highly connected and centrally positioned peers.

Implications of partitioning the connectivity graph Peer-to-peer networks are known for their dynamic nature, allowing them to adapt to changing conditions. However, our research reveals that even this inherent dynamicity is insufficient in countering targeted attacks. The connectivity graphs we construct serve as representations of all potential connections that could exist in the actual network. Each edge in the connectivity graph signifies that two nodes are aware of each other's presence and have the ability to establish a connection.

Conversely, the absence of an edge in the connectivity graph indicates that two nodes are unaware of each other's existence and are highly unlikely to establish a connection. Partitioning of the connectivity graph has significant implications. Nodes within a specific partition not only become disconnected from other network partitions but also lack the knowledge required to establish links with nodes in different partitions. In essence, the nodes are confined to their own partition and remain unaware that a portion of the network has become disconnected.

Limitations Measurement errors in network analysis are not infrequent [50] and our approach is not an exception. In fact, the proposed method introduces a number of false edges in the graph. Second, it is possible, albeit rare, that certain edges may be overlooked (see Section 3.1). To understand how much the calculated network properties are affected by these errors, we looked into related studies that investigate the effect of measurement errors in network data. In [52], we provide an in-depth analysis of these limitations and examine the impact of false-positive edges. Our findings suggest that the observed connectivity graphs demonstrate greater resilience compared to the actual connections in the real network.

Wang *et al.* [57] studied the effect of measurement errors on node-level network measures and found that networks are relatively robust to false positive edges. Booker [11] measures the effects of measurement errors on the attack vulnerability of networks. Similarly to [57], Booker also finds that false positive edges have the least impact on the effectiveness of random and targeted attacks. From the same work, it is also evident that an error rate of 5% in missing links is acceptable, when analyzing the impact of different targeting strategies on the network structure. We believe that the error rate observed in our study is small enough to allow us to draw meaningful conclusions.

We readily admit that it is possible to miss connections from unreachable peers towards reachable peers. This resilience assessment relies on the assumption that these links constitute a small minority of all possible links. Our validation results in Section 3.1 support this assumption. This assumption is also supported by [35], which estimates an average degree of 9.8 for unreachable peers on the Bitcoin network. Our measurements estimate an average degree of 37 for unreachable peers. Furthermore, in [58] Wang and Pustogarov estimate that unreachable peers establish only 3.5 connections to the network, on average. Interestingly, they also find that such unreachable peers are not merely disposable nodes of the network. Instead, they are involved in the propagation of 43% of Bitcoin transactions. Our resilience study demonstrates that attacking a handful of key peers can disconnect a large number of unreachable peers and thus can severely affect message propagation in the network.

Moreover, transient disruptions of the network would increase the likelihood of forks and could facilitate attacks against consensus. DDoS attacks or BGP hijacks against a carefully selected AS could partition 10% to 50% nodes from a network, while a disruption in a handful of ASes has the potential to remove almost half of all BC nodes in most systems simultaneously.

6 Summary

Our results raise alarm about the resilience of the studied blockchains against partitioning and message propagation delay attacks. We demonstrate that by using our methodology, a deliberate and methodical attacker can uncover a small set of entities central to the topology and target them to substantially suppress message propagation in more than one BC network simultaneously. Importantly, all networks seem vulnerable to at least one type of attack strategy. This highlights the need to employ measures to enhance network robustness or employ open topology protocols, rather than relying on topology hiding techniques to secure the overlay network.

7 Acknowledgements

This project has received funding from the European Union's Horizon 2020 Research and Innovation program under the Marie Skłodowska-Curie INCOGNITO

project (Grant Agreement No. 824015), CONCORDIA project (Grant Agreement No. 830927), SPATIAL project (Grant Agreement No. 101021808) and the Cyprus's Research and Innovation Foundation (Grant Agreement: COMPLEMENTARY/0916/0031). The authors bear the sole responsibility for the content presented in this paper, and any interpretations or conclusions drawn from it do not reflect the official position of the European Union nor the Research Innovation Foundation.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**, 47–97 (Jan 2002). https://doi.org/10.1103/RevModPhys.74.47, https://link.aps.org/doi/10.1103/RevModPhys.74.47
2. Albert, R., Jeong, H., Barabási, A.: Error and attack tolerance of complex networks. Nature **406**(6794), 378–382 (Jul 2000). https://doi.org/10.1038/35019019
3. Allman, M., Paxson, V.: Issues and etiquette concerning use of shared measurement data. In: IMC. ACM (2007)
4. Anonymous: A first look into blockchain overlays - dataset. Online (2021), https://drive.google.com/drive/folders/1115O8SY8U9NLZARzhcO1Q-8Vzdn3WaSy
5. Apostolaki, M., Zohar, A., Vanbever, L.: Hijacking bitcoin: Routing attacks on cryptocurrencies. In: 2017 IEEE Symposium on Security and Privacy, S&P 2017. IEEE Computer Society (2017). https://doi.org/10.1109/SP.2017.29, https://doi.org/10.1109/SP.2017.29
6. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: Medrec: Using blockchain for medical data access and permission management. In: 2016 2nd International Conference on Open and Big Data (OBD) (2016). https://doi.org/10.1109/OBD.2016.11
7. Baumann, A., Fabian, B.: How robust is the internet? - insights from graph analysis. In: Crisis (2014)
8. Ben Mariem, S., Casas, P., Donnet, B.: Vivisecting blockchain p2p networks: Unveiling the bitcoin ip network. In: ACM CoNEXT Student Workshop (2018)
9. Biryukov, A., Tikhomirov, S.: Deanonymization and linkability of cryptocurrency transactions based on network analysis. In: IEEE European Symposium on Security and Privacy (EuroS&P) (2019). https://doi.org/10.1109/EuroSP.2019.00022
10. bitnodes.io: Global bitcoin nodes distribution. Online (September 2020), https://bitnodes.io
11. Booker, L.B.: The effects of observation errors on the attack vulnerability of complex networks (2012)
12. Chatzigiannis, P., Baldimtsi, F., Griva, I., Li, J.: Diversification across mining pools: optimal mining strategies under pow. J. Cybersecur. **8**(1) (2022)
13. Chen, W., Xu, Z., Shi, S., Zhao, Y., Zhao, J.: A survey of blockchain applications in different domains. In: ICBTA. pp. 17–21. ACM (2018)
14. Chen, W., Xu, Z., Shi, S., Zhao, Y., Zhao, J.: A survey of blockchain applications in different domains. In: Proceedings of the 2018 International Conference on Blockchain Technology and Application. ICBTA 2018, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3301403.3301407, https://doi.org/10.1145/3301403.3301407
15. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. IEEE Access **4** (2016). https://doi.org/10.1109/ACCESS.2016.2566339
16. CoinMarketCap: Coinmarketcap. Online (2021), https://coinmarketcap.com

17. Core, B.: 0.20.1 release notes. Online (2021)
18. Daniel, E., Rohrer, E., Tschorsch, F.: Map-z: Exposing the zcash network in times of transition. In: LCN. IEEE (2019)
19. Decker, C., Wattenhofer, R.: Information propagation in the bitcoin network. In: 13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013. IEEE (2013). https://doi.org/10.1109/P2P.2013.6688704
20. Delgado-Segura, S., Bakshi, S., Pérez-Solà, C., Litton, J., Pachulski, A., Miller, A., Bhattacharjee, B.: Txprobe: Discovering bitcoin's network topology using orphan transactions. In: Financial Cryptography. Lecture Notes in Computer Science, vol. 11598. Springer (2019)
21. Developers, B.C.: Bitcoin core 0.10.1 release notes. Online (April 2015), `https://github.com/bitcoin/bitcoin/blob/v0.10.1/doc/release-notes.md`, `https://github.com/bitcoin/bitcoin/blob/v0.10.1/doc/release-notes.md`
22. Developers, B.C.: Bitcoin core integration/staging tree. Online (2021), `https://github.com/bitcoin/bitcoin`
23. Developers, B.C.: Bitcoin p2p network. Online (2021), `https://developer.bitcoin.org/devguide/p2p_network.html`
24. Dittrich, D., Kenneally, E., et al.: The menlo report: Ethical principles guiding information and communication technology research. Tech. rep., US Department of Homeland Security (2012)
25. Donet Donet, J.A., Pérez-Solà, C., Herrera-Joancomartí, J.: The bitcoin p2p network. In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) Financial Cryptography and Data Security. pp. 87–102. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
26. Dotan, M., Pignolet, Y.A., Schmid, S., Tochner, S., Zohar, A.: Sok: Cryptocurrency networking context, state-of-the-art, challenges. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. ARES '20, ACM (2020). https://doi.org/10.1145/3407023.3407043
27. Ethereum: Ethereum peer-to-peer networking specifications. Online (2014), `https://github.com/ethereum/devp2p`
28. Fechner, J., Chandrasekaran, B., Makkes, M.X.: Calibrating the performance and security of blockchains via information propagation delays: revisiting an old approach with a new perspective. Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (2022)
29. Feld, S., Schönfeld, M., Werner, M.: Analyzing the deployment of bitcoin's p2p network under an as-level perspective. Procedia Computer Science **32**, 1121–1126 (2014). https://doi.org/https://doi.org/10.1016/j.procs.2014.05.542, `https://www.sciencedirect.com/science/article/pii/S187705091400742X`, the 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014)
30. Fiedler, M.: Algebraic connectivity of graphs. Czechoslovak mathematical journal **23**(2) (1973)
31. Forum, B.: Uasf nodes wrongly reporting ip (2017), `https://bitcointalk.org/index.php?topic=1954151.0`
32. Franzoni, F., Daza, V.: Sok: Network-level attacks on the bitcoin p2p network. IEEE Access **10**, 94924–94962 (2022). https://doi.org/10.1109/ACCESS.2022.3204387
33. Gencer, A.E., Basu, S.S., Eyal, I., van Renesse, R., Sirer, E.G.: Decentralization in bitcoin and ethereum networks. In: Financial Cryptography (2018)

34. Gochhayat, S.P., Shetty, S.S., Mukkamala, R., Foytik, P.B., Kamhoua, G.A., Njilla, L.L.: Measuring decentrality in blockchain based systems. IEEE Access **8**, 178372–178390 (2020)
35. Grundmann, M., Amberg, H., Baumstark, M., Hartenstein, H.: Short paper: What peer announcements tell us about the size of the bitcoin P2P network. In: Financial Cryptography. Lecture Notes in Computer Science, vol. 13411, pp. 694–704. Springer (2022)
36. Grundmann, M., Neudecker, T., Hartenstein, H.: Exploiting transaction accumulation and double spends for topology inference in bitcoin. In: Financial Cryptography Workshops. Lecture Notes in Computer Science, vol. 10958. Springer (2018)
37. Harary, F.: The maximum connectivity of a graph. Proceedings of the National Academy of Sciences of the United States of America **48**(7) (1962)
38. Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on bitcoin's peer-to-peer network. In: 24th USENIX Security Symposium (USENIX Security 15). USENIX Association (Aug 2015)
39. Jr, L.D.: Bitcoin historical node count (2022), `https://luke.dashjr.org/programs/bitcoin/files/charts/historical.html`
40. Kim, H., Anderson, R.J.: An experimental evaluation of robustness of networks. IEEE Systems Journal **7**, 179–188 (2013)
41. Kim, S.K., Ma, Z., Murali, S., Mason, J., Miller, A., Bailey, M.: Measuring ethereum network peers. In: IMC. ACM (2018)
42. Li, K., Tang, Y., Chen, J., Wang, Y., Liu, X.: Toposhot: uncovering ethereum's network topology leveraging replacement transactions. In: Internet Measurement Conference. pp. 302–319. ACM (2021)
43. Magoni, D.: Tearing down the internet. IEEE Journal on Selected Areas in Communications **21**(6) (2003)
44. Marcus, Y., Heilman, E., Goldberg, S.: Low-resource eclipse attacks on ethereum's peer-to-peer network. IACR Cryptol. ePrint Arch. **2018** (2018), `http://eprint.iacr.org/2018/236`
45. Maymounkov, P., Mazières, D.: Kademlia: A peer-to-peer information system based on the XOR metric. In: IPTPS. Lecture Notes in Computer Science, vol. 2429. Springer (2002)
46. Miller, A., Litton, J., Pachulski, A., Gupta, N., Levin, D., Spring, N., Bhattacharjee, B.: Discovering bitcoin's network topology and influential nodes. University of Maryland, Tech. Rep (2015)
47. Muro, M.A.D., Valdez, L.D., Rêgo, H.H.A., Buldyrev, S.V., Stanley, H.E., Braunstein, L.A.: Cascading failures in interdependent networks with multiple supply-demand links and functionality thresholds. Scientific Reports **7** (2017)
48. Neudecker, T.: Characterization of the bitcoin peer-to-peer network (2015-2018). Tech. Rep. 1, Karlsruher Institut für Technologie (KIT) (2019). https://doi.org/10.5445/IR/1000091933
49. Neudecker, T., Andelfinger, P., Hartenstein, H.: Timing analysis for inferring the topology of the bitcoin peer-to-peer network. In: UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld. IEEE Computer Society (2016)
50. Newman, M.E.J.: Measurement errors in network data. ArXiv **abs/1703.07376** (2017)
51. Nick, J.: Guessing bitcoin's p2p connections. Online (2015), `https://jonasnick.github.io/blog/2015/03/06/guessing-bitcoins-p2p-connections/`

52. Paphitis, A., Kourtellis, N., Sirivianos, M.: Graph Analysis of Blockchain P2P Overlays and their Security Implications. In: LNCS. Lecture Notes in Computer Science, vol. 9th International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec 2023). Springer (2023)
53. Rivers, C., Lewis, B.: Ethical research standards in a world of big data. F1000Research **3**(38) (2014). https://doi.org/10.12688/f1000research.3-38.v2
54. Saad, M., Cook, V., Nguyen, L., Thai, M.T., Mohaisen, A.: Partitioning attacks on bitcoin: Colliding space, time, and logic. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS) (2019). https://doi.org/10.1109/ICDCS.2019.00119
55. Tran, M., Choi, I., Moon, G.J., Vu, A.V., Kang, M.S.: A stealthier partitioning attack against bitcoin peer-to-peer network. In: 2020 IEEE Symposium on Security and Privacy (SP) (2020). https://doi.org/10.1109/SP40000.2020.00027
56. trinity.ethereum.org: The trinity ethereum client. Online (2021), `https://trinity.ethereum.org`
57. Wang, D.J., Shi, X., McFarland, D.A., Leskovec, J.: Measurement error in network data: A re-classification. Soc. Networks **34**, 396–409 (2012)
58. Wang, L., Pustogarov, I.: Towards better understanding of bitcoin unreachable peers. CoRR **abs/1709.06837** (2017)
59. Wuille, P.: Replace global trickle node with random delays. Online (2022), `https://github.com/bitcoin/bitcoin/pull/7125`
60. Yang, J., Sun, G., Xiao, R., He, H.: Detectable, traceable, and manageable blockchain technologies bhe: An attack scheme against bitcoin p2p network. Wireless Communications and Mobile Computing (2022)
61. Yeow, A.: Bitnodes network crawler (2021), `https://github.com/ayeowch/bitnodes`
62. Zyskind, G., Nathan, O., Pentland, A.S.: Decentralizing privacy: Using blockchain to protect personal data. In: 2015 IEEE Security and Privacy Workshops (2015). https://doi.org/10.1109/SPW.2015.27

A Ethics

In this work we followed standard ethical guidelines [24,53,3] for the collection and sharing of measurement data. We only collect and process publicly available data, make no attempt to deanonymize users or link people and/or organizations to their IP address. No personally identifiable information was collected.

While crawling the networks we only take part in the peer discovery mechanism of each network and gather IP addresses known to each node. Those addresses were only used to synthesize connectivity graphs on which our research was based. We did not try to identify any user by her IP address and no information was redistributed. In fact, our crawler created short lived connections to any discovered peer in the network and did not respond to any other requests except the expected initial handshake. We do not respond to any other messages or requests. In addition, we employed low bandwidth utilization to avoid resource exhaustion. Our measurements did not cause any disruption or exposure of the BC networks under study.

Our results unveil particular nodes whose targeting has the potential to disrupt the overlay's operation. To prevent misuse of this portion of the results, we

do not publish the IP address of any node in our dataset. We instead replace the IP address with a persistent random identifier and we privately maintain a private map of IPs to random identifiers for verification and reproducibility purposes.