You Shall Not Join: A Measurement Study of Cryptocurrency Peer-to-Peer Bootstrapping Techniques

Angelique Faye Loe
Royal Holloway, University of London
Information Security Group
angelique.loe.2016@live.rhul.ac.uk

Elizabeth Anne Quaglia
Royal Holloway, University of London
Information Security Group
elizabeth.quaglia@rhul.ac.uk

ABSTRACT

Cryptocurrencies are digital assets which depend upon the use of distributed peer-to-peer networks. The method a new peer uses to initially join a peer-to-peer network is known as bootstrapping. The ability to bootstrap without the use of a centralized resource is an unresolved challenge. In this paper we survey the bootstrapping techniques used by 74 cryptocurrencies and find that censorship-prone methods such as DNS seeding and IP hard-coding are the most prevalent. In response to this finding, we test two other bootstrapping techniques less susceptible to censorship, Tor and ZMap, to determine if they are operationally feasible alternatives more resilient to censorship. We perform a global measurement study of DNS query responses for each of the 92 DNS seeds discovered across 42 countries using the distributed RIPE Atlas network. This provides details of each cryptocurrency's peer-to-peer network topology and also highlights instances of DNS outages and query manipulation impacting the bootstrapping process. Our study also reveals that the source code of the cryptocurrencies researched comes from only five main repositories; hence accounting for the inheritance of legacy bootstrapping methods. Finally, we discuss the implications of our findings and provide recommendations to mitigate the risks exposed.

CCS CONCEPTS

• **Networks** → **Peer-to-peer protocols**; • **Computer systems organization** → **Peer-to-peer architectures**.

KEYWORDS

cryptocurrency; peer-to-peer; bootstrapping; censorship-resistance

ACM Reference Format:
Angelique Faye Loe and Elizabeth Anne Quaglia. 2019. You Shall Not Join: A Measurement Study of Cryptocurrency Peer-to-Peer Bootstrapping Techniques. In *2019 ACM SIGSAC Conference on Computer and Communications Security (CCS'19), November 11–15, 2019, London, United Kingdom.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3319535.3345649

CCS '19, November 11–15, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6747-9/19/11…$15.00
https://doi.org/10.1145/3319535.3345649

1 INTRODUCTION

Cryptocurrencies are decentralized digital assets that do not rely on trusted third parties for the execution of transactions between parties. Decentralization is achieved through the use of distributed ledger technology to provide an append-only chronicle of all transactions between parties sending and receiving funds. The immutability of transactions is achieved through a fiscally incentivized consensus protocol publicly executed by peers participating in a peer-to-peer network [51]. The decentralized nature of cryptocurrencies provides a promise of technological amelioration to traditional payment systems [23, 27, 51, 59]. Due to the unprecedented pace of mainstream adoption and investment into these digital assets, they have attracted considerable scrutiny and research [19, 24, 26, 33, 34, 40, 47–50].

We present a measurement study of the techniques used to initially connect 74 different cryptocurrency peers to their relevant distributed peer-to-peer networks. When a new peer first joins a peer-to-peer network the action describing this initial connection is known as *bootstrapping*. By surveying the bootstrapping techniques of these cryptocurrencies we are able to uncover the following elements about their operation: the inability to resist known censorship techniques to prevent peer connectivity, the feasibility of employing censorship-resistant connection techniques, details of their peer-to-peer traffic and network infrastructures, evidence of connection manipulation and outages impacting peer connectivity, and insight into the inheritance of their source code. Based on our findings we are able to present a number of security and social implications of cryptocurrency bootstrapping censorship not known from previous studies [15, 58].

We first review the bootstrapping techniques known from related studies into distributed system connectivity. We classify the techniques into three categories based on their ability to withstand censorship and record the advantages and disadvantages of each method. The categories identified are censorship-prone methods, such as Domain Name System (DNS) seeding and IP hard-coding, censorship-mitigated methods, such as via Internet Relay Chat (IRC) or Tor; and finally, censorship-resistant techniques such as IPv4 scans using ZMap [14, 16, 29]. Due to the widespread use of cryptocurrency exchanges, we also identify how these trusted third parties fit into the peer-to-peer ecosystem in relation to bootstrapping functionality.

Next, we present the results of our survey to determine the bootstrapping techniques used by the cryptocurrencies researched. The survey highlights that 95% of cryptocurrencies use censorship-prone techniques of bootstrapping based on either DNS seeding or IP hard-coding. Furthermore, 32% of cryptocurrencies use a single DNS provider for their DNS seeds, exposing a single point of service

failure. Also, 88% of the cryptocurrencies use distinctive destination ports, making traffic identification trivial. Based on these findings we employ known censorship techniques to ascertain if any fallback techniques of bootstrapping exist on the cryptocurrencies tested. Unfortunately, in all but one case our findings highlight that fallback techniques were equally prone to basic censorship. This led us to test the feasibility of bootstrapping via a censorship-mitigated technique using Tor, which presented various results for connectivity success. Finally, we test the ability to bootstrap via a censorship-resistant technique using a peer initiated ZMap IPv4 scan. Unfortunately, this technique is limited by large latency overheads and, most importantly, not a single cryptocurrency could connect using this method.

In the second part of our measurement study, as a consequence of highlighting the prevalence of DNS seeding as a bootstrapping method, we exploit this side-channel to measure details about the cryptocurrencies peer-to-peer networks over a number of months. Using the globally distributed network of RIPE Atlas probes, our research consists of measuring the details of the DNS query responses to the 92 DNS seeds uncovered during our bootstrapping survey. This study is done across 46 locations in 42 different countries with varying legal standpoints towards cryptocurrencies. Our measurements uncover the topology of each cryptocurrency's peer-to-peer network and highlight that some cryptocurrencies use centralized rendezvous servers as part of their connectivity strategy whilst others unintentionally expose the IPs of their peers through the harvesting of DNS query responses. Through this measurement study we also determine a range of statistics regarding the number of peer IPs seen on a per cryptocurrency and per country basis, as well as highlighting the number of IPs active across multiple cryptocurrencies. Our data also reveals how DNS seeding outages and query response manipulations impact bootstrapping operations in 60% of the countries we investigate.

Furthermore, our research exposes that the root cause of the prevalence of censorship-prone bootstrapping techniques is due to the widespread practice of copying source code. Our findings highlight that all of the cryptocurrencies researched derive from only five parent source code repositories. Additionally, we present a number of security implications and social implications related to cryptocurrency censorship based on the discoveries in our study. We conclude by providing a catalogue of tactical and strategic recommendations to mitigate the shortcomings to cryptocurrency bootstrapping identified by our research.

2 BACKGROUND AND RELATED WORK

In this section we introduce the concept of peer-to-peer bootstrapping. We then perform an analysis of related work outlining various peer-to-peer bootstrapping techniques and classify them based on their ability to withstand censorship. Finally, we discuss how cryptocurrency exchanges fit into the process of peer-to-peer bootstrapping to facilitate the trade and acquisition of cryptocurrencies.

2.1 Peer-to-Peer Bootstrapping

The method in which new peers initially join a peer-to-peer network is commonly referred to as bootstrapping. Bootstrapping can be succinctly described as '*the process that a new peer who intends to join a peer-to-peer network uses to discover contact information for another peer in the existing network*' [29]. The ability to bootstrap onto a peer-to-peer overlay network without the use of centralized resources remains an outstanding challenge [42]. Any centralized elements on a peer-to-peer network mark a point for regulation and censorship and are ideally avoided.

However, a challenge arises when a new peer with no knowledge of other peers wishes to join the network. They must first be able to determine if they are the first peer on the network and, if not, they must be able to find and connect to at least one other peer in order to join the existing peer-to-peer network [37, 38]. We will now review the methods of peer-to-peer bootstrapping.

2.2 Censorship-Prone Bootstrapping

In this section we review two legacy methods of peer-to-peer bootstrapping which are heavily prone to censorship. For context, we also provide two brief examples of peer-to-peer networks impacted by bootstrapping censorship.

The first method to bootstrap peers onto a peer-to-peer network is to ship the software with a preconfigured list of peer IP addresses, known as hard-coding. Hard-coding is not ideal because the list of peers can quickly become obsolete and it gives an adversary a method to learn the details about the peer-to-peer network [29].

The second censorship-prone method to bootstrap onto a peer-to-peer network is the use of DNS seeds. DNS is a hierarchical client-server protocol which maps domain names to IP addresses [41]. For example, a new peer querying a DNS server for the DNS seed node-london.cryptonex.org for the Cryptonex cryptocurrency will have an A record return a single IP or multiple IPs. The peer will then try to connect to the IPs when bootstrapping onto the network. DNS seeding is not ideal because DNS is a client-server based Internet protocol which is easily censored due to its centralized nature and mainly cleartext communication mechanisms.

Both methods have the advantage of being easy to configure and implement. The developer need only hard-code the relevant IPs or DNS seeds into the source code. DNS seeding also requires the additional step of configuring the DNS zone and records. We will also see in Section 5.2 that DNS seeding unintentionally leaks information about peer IP addresses.

The ability to censor peer-to-peer networks based on these bootstrapping methods can be recalled from peer-to-peer file-sharing technologies used to exchange digitally encoded music and videos in the late 1990s and early 2000s [30]. The two most prominent file sharing services, Napster and The Pirate Bay (TPB), were shutdown or censored after they were found legally accountable for copyright infringement. Napster was shutdown in 2001 after losing a legal battle with the Recording Industry Association of America. The service was easily shutdown due to the use of a centralized bootstrapping method dependent on front-end servers which allowed peers to learn about the content hosted on other peers on the network [52]. Censorship mechanisms currently used to limit access to TPB are IP black-lists and DNS level blocking, both of which would be highly effective at censoring IP hard-coding and DNS seed bootstrapping methods [43].

Table 1: Comparison of Peer-to-Peer Bootstrapping Methods. Noted columns: vulnerability to censorship, ease of configuration, if hard-coding is shifted to another method, such as IRC or Tor, if the other shifted method is also vulnerable to censorship, centralized dependencies, obfuscating peer-to-peer traffic, high latency and high bandwidth requirements. ● = yes, ○ = no, × = N/A.

Method	*Censorship*	*ease of conf*	*shift h/c*	*method cens.*	*central dep.*	*hides p2p*	*high latency*	*high b.w.*
Hard-Coding	prone	●	×	×	○	○	○	○
DNS Seeding	prone	●	×	×	●	○	○	○
via IRC	mitigated	○	●	●	●	○	○	○
via Tor	mitigated	○	●	●	○	●	●	○
ZMap Scan	resilient	○	○	○	○	○	○	●

2.3 Censorship-Mitigated Bootstrapping

In this section we explore the related work covering two bootstrapping methods which address the disadvantages of IP hard-coding and DNS seeding.

The first method relies on the use of IRC servers [37]. IRC is a client-server Internet protocol which allows clients to send chat messages to other clients via distributed chat servers. This method exploits the distributed network of IRC servers as rendezvous points for bootstrapping. By using these servers, centralized points of failure for initial peer connection are removed. The issue with this method is the underlying requirement to find an IRC server to connect to. Unfortunately, finding IRC servers also depends on using a hard-coded list of IPs or networks, or via DNS seeds [12]. Due to this deficiency, IRC bootstrapping is not widely deployed.

The second censorship-mitigated bootstrapping technique is dependent on the use of Tor hidden services. This method channels peer-to-peer traffic via the Tor network. Tor is a volunteer-based overlay network enabling its users to browse the Internet anonymously [14]. To obtain anonymity, traffic is first encrypted and then passed through a series of at least three hops, namely an entry, middle, and exit relay. An advantage of this method is that peer-to-peer traffic appears to be Tor traffic. Unfortunately, several countries wishing to regulate and censor Internet access attempt to detect and block Tor and other Virtual Private Network (VPN) traffic [39]. Censoring Tor is possible because default configurations of Tor are easily identified based on distinct TCP destination port usage and the characteristics of the TLS handshake between the Tor client and entry relay [2]. Tor can also be censored at IP level because all Tor relays can be identified by publicly available information stored in Tor directory authorities.

Tor addresses these shortcomings by introducing *pluggable transports* and Tor bridges [45]. Pluggable transports utilize the steganographic concept of *security by obscurity* by attempting to make Tor traffic appear as standard TLS traffic. In this way, pluggable transports provide an advantage as peer-to-peer traffic is more difficult to identify.

Tor bridges are entry relays which do not have their IPs publicly available on Tor directory authorities [45]. Unfortunately, default Tor bridges also suffer from the dependence of hard-coding IPs into the Tor browser bundle leaving them vulnerable to censorship. However, to help circumvent censorship private Tor bridges are not hard-coded into the Tor browser bundle. By design, the discovery of private bridges depends on the use of manual side-channel requests, such as email. This discovery overhead makes the use of private bridges for peer-to-peer connectivity challenging.

One of the biggest disadvantages of utilizing Tor for peer-to-peer bootstrapping is the bandwidth limitations and latency. This is due to the overheads of the three-hop encrypted relay design. In the case of bootstrapping peer-to-peer cryptocurrencies with a requirement to download large blockchains, the former shortcomings can pose significant operational issues, as we shall see in Section 4.3. Finally, it is important to note that other research observes that Tor is not a panacea for anonymity, especially when considering cryptocurrencies [24, 32].

2.4 Censorship-Resistant Bootstrapping

We finally discuss related work regarding the most censorship-resistant method of bootstrapping based on IPv4 scanning.

The earliest peer-to-peer bootstrapping scan-based method was based on geographically targeted IP scanning [29]. The idea behind this method is the creation of a profile of the IPs already part of the network. Once the IPs are determined, a targeted scan is formulated based on distribution information learnt from DNS. The advantage of this method is the focus to remove centralized elements of bootstrapping and the need to hard-code IPs. Unfortunately, this method still relies on the centralized DNS protocol and also assumes that the peer-to-peer network is already established so that learning the details of the peer IPs is possible.

However, this method importantly explores a peer-initiated scan of the IPv4 address space in order to learn about other peers. The computational feasibility to scan the entire IPv4 address space is now possible due to advances in the ZMap network scanner [16]. ZMap is able to scan the entire IPv4 address space for as single TCP port in 4.5 minutes given a 10 Gpbs Ethernet connection. The advantages of this scanning method is the removal of any centralized dependency. However, the stated scan time of 4.5 minutes assumes that a high bandwidth connection to the Internet is available. Also, this technique of bootstrapping is very bandwidth intensive. It generates 4.6 Gbps of traffic for each peer wishing to bootstrap [16]. Furthermore, typical broadband speeds are below 10 Gbps. Broadband speeds in Singapore average 60 Mbps, which ranks as the world's fastest, whilst speeds in the United States and Sweden average 26 Mbps and 46 Mbps respectively. On the other end of the scale Venezuela has an average broadband speed of only 1 Mbps and Yemen has the slowest average speed of < 1 Mbps [35].

These bandwidth limitations in particular countries are an important point to consider based on the motivations for certain peers to acquire cryptocurrencies. For example, peers in Venezuela may wish to invest in cryptocurrencies due to current political uncertainty causing hyperinflation to the countries' fiat currency [36]. The security implications of cryptocurrency censorship will be discussed further in Section 7.1.

Table 1 summarizes the different peer-to-peer bootstrapping methods and the advantages and disadvantages of each.

2.5 Cryptocurrency Exchanges

A brief discussion of cryptocurrency exchanges provides context regarding the pervasiveness of centralized infrastructure inherent in cryptocurrency acquisition and frames how peer-to-peer cryptocurrency bootstrapping fits into the process.

Cryptocurrency exchanges are a heavily utilized centralized infrastructure used to acquire and trade cryptocurrencies. This is in direct contradiction with the original peer-to-peer concepts outlined in the original Bitcoin paper [51]. The incongruity between the ethos of these institutions and the initial principles of Bitcoin has been the subject of other research [19, 49, 57].

The centralized nature of exchanges makes them natural targets for theft [57]. There is also strong evidence of cryptocurrency market manipulation based on the use of exchanges [33]. Despite these shortcomings, an estimated 99% of cryptocurrency trading volume is executed through exchanges [31].

Peers can avoid the use of these exchanges by participating directly on the peer-to-peer networks through the use of core reference client software. However, we note that, whilst cryptocurrency exchanges typically execute trades off-chain because of the latency associated with clearing transactions on the blockchain, they must also eventually broadcast their transactions on-chain [28]. In order for on-chain transactions to be successfully appended to the blockchain, exchanges must also bootstrap and join as peers onto the peer-to-peer networks using core reference client software. Cryptocurrency exchanges have thus become a centralized proxy mechanism for peer connectivity.

3 RESEARCH METHODOLOGY OVERVIEW

In this section we present the selection process of the cryptocurrencies that we survey and outline our research steps.

3.1 Selection of Cryptocurrencies

Despite the volatility in the cryptocurrency market caused by new forks of well-known currencies, and new cryptocurrencies being introduced [5, 6], we needed to lock in a dataset for our research. The top 100 cryptocurrencies based on USD market capitalization was selected on February 28, 2018 [9]. These cryptocurrencies can be classified into four categories: mineable coins, non-mineable coins, mineable tokens and non-mineable tokens, noted in Table 5 in Appendix B.

The difference between coins and tokens is that coins have their own native blockchains, and tokens are built using a template on an existing blockchain [60]. For tokens, the core elements of peer-to-peer connectivity, including bootstrapping, depend on the method used by the underlying coin. Therefore, studying the bootstrapping behaviour of tokens is captured by researching the underlying coin.

Mineable and non-mineable cryptocurrencies are differentiated by their methods of acquisition. Non-mineable cryptocurrencies are generally acquired through centralized exchanges. Conversely, prior to the creation of exchanges, mineable cryptocurrencies were acquired either through direct peer-to-peer transfer or through the process of mining. The process of mining is described in the original Bitcoin paper [51].

Figure 1 shows the selected cryptocurrencies classified into four categories, 25 of these are non-mineable coins, 25 are mineable coins, and 50 are tokens. Non-mineable coins were excluded from our study as they depend on centralized exchanges for acquisition. That is, in relation to this study, they do not have a bootstrapping mechanism associated with peer-to-peer networking.

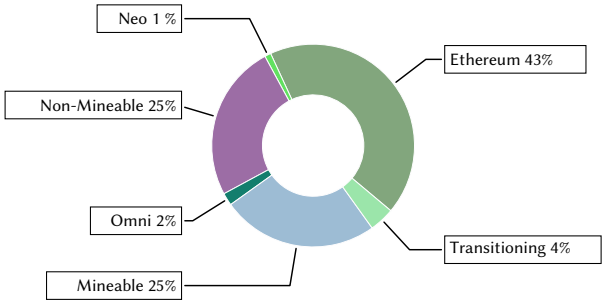

Figure 1: Breakdown of the Top 100 Cryptocurrencies

The 50 tokens are represented by the green hues, namely Omni, Neo, Ethereum and Transitioning tokens. The underlying coin for the Omni tokens is Bitcoin, whilst the underlying coins for the Ethereum and Transitioning tokens is the Ethereum coin. Ethereum tokens are also commonly referred to as ERC20 tokens. The four 'Transitioning' tokens were ERC20 tokens in the midst of migrating to their own native blockchains during our research period. Finally, the Neo token resides on an underlying coin called Gas. Gas is a non-mineable coin; therefore, the Neo token and the Gas coin were both excluded from our research for the reasons noted previously.

In summary, our research consisted of measuring the bootstrapping methods of the 25 mineable coins, thus including the 43 ERC20, four Transitioning and two Omni tokens in our study.

3.2 Summary of Research Steps

A first element of our research is to survey the bootstrapping methods of 25 mineable coins and their 49 underlying tokens, noted in Section 3.1. The survey is completed by installing the core-reference client of each mineable coin onto a virtual machine and recording its connectivity behaviour during the bootstrapping process. We then cross-reference the behaviour against the open source code repositories of each coin. We then iteratively test basic censorship mechanisms, such as IP blacklisting and DNS sinkholing in order to determine fallback methods of bootstrapping.

Upon source code inspection, we uncover that several coins have configurable means to connect via Tor. Therefore, we test and record the efficacy of this bootstrapping technique. Furthermore, we test a censorship-resistant method of bootstrapping by using ZMap to scan for potential peers and record the results.

We then perform a global measurement study over 46 geographically distributed RIPE Atlas probes in 42 countries to query the 92 DNS seeds discovered in our bootstrapping survey. RIPE Atlas is a geographically spread, volunteer-based network of probes used to measure Internet metrics such as network latency, traceroute paths, and DNS, SSL/TLS, HTTP and NTP responses [10].

Finally, as our research dictates the source code survey of numerous cryptocurrencies, we uncover the root cause of inheritance of legacy bootstrapping methods by mapping the software fork

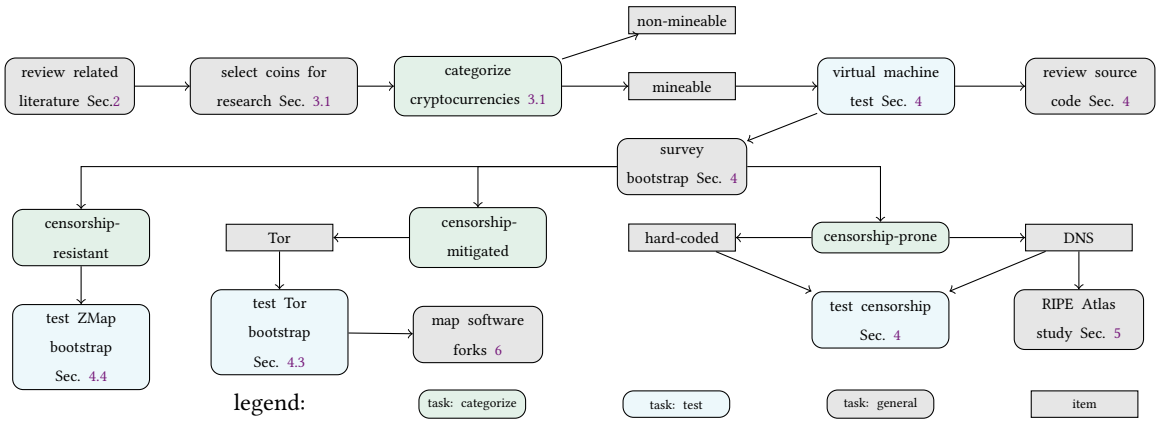

Figure 2: Research Methodology Flow Chart

lineage of the cryptocurrencies studied. In Figure 2 we outline the flow of our research steps and note the corresponding sections.

4 SURVEY OF CRYPTOCURRENCY BOOTSTRAPPING METHODS

This section presents the survey of bootstrapping methods used on the 25 mineable coins and the 49 underlying tokens selected in Section 3.1. The highlight of the survey indicates that 95% of the cryptocurrencies tested use only censorship-prone bootstrapping methods. We also test and record the challenges of channelling peer-to-peer bootstrapping traffic via Tor and test the feasibility of bootstrapping using ZMap IPv4 scanning.

4.1 Determining Bootstrapping Methods

The first task is to install the core-reference client software on 25 virtual machines and determine the bootstrapping behaviour through the use of network analysis tools and application debug logs. Next, we cross-reference the identified behaviour with the open source code of each cryptocurrency. As we expect, in all cases the bootstrapping connection behaviour reflects what is in the source code. We determine that three main options of peer-to-peer bootstrapping are DNS seeding, hard-coding IP seeds, and bootstrapping via Tor.

Table 2 records each coins bootstrapping method. Also, in Appendix B, the tooling used to discover the bootstrapping methods is recorded in Table 4 and the details of the source code for bootstrapping functionality can be found in Table 6.

DNS seeding is the most common bootstrapping method identified; therefore, we also review the DNS provider diversity for the 92 DNS seeds discovered, which are noted in Table 7 in Appendix B. It is recommended that DNS name services are sourced from redundant providers to mitigate denial of service risks [20]. However, nearly one third of the coins, namely Monero, Dogecoin, Electroneum, Zclassic, Syscoin, Cryptonex, Monacoin, and Zcoin use only a single DNS provider.

Our survey allows us to determine the characteristics of cryptocurrency peer-to-peer traffic. In Section 2.3 we noted that countries wishing to regulate and censor Internet access attempt to block Tor traffic. Recall that Tor traffic can be identified based on destination TCP port usage, and IP connectivity to publicly published relays. Unfortunately, the same characteristics impact the ability to regulate and censor peer-to-peer cryptocurrency traffic.

95% of the cryptocurrencies evaluated in this survey use distinct TCP and UDP destination ports, making the traffic easy to identify. In Table 6 in Appendix B we record the destination ports of all the cryptocurrencies surveyed. The ports on which cryptocurrencies operate are easily learned directly from reviewing the open source code or from using standard network packet analysis tools. Methods to mitigate the trivial identification of cryptocurrency peer-to-peer traffic are discussed in Section 8.2.

4.2 Identifying Bootstrapping Resilience Through Basic Censorship Testing

We now proceed to test the resilience of each cryptocurrency's bootstrapping ability by iteratively applying different censorship methods. To emulate DNS censorship, we populate the virtual machines local hosts file with the static mapping of all domain names noted in Table 7 in Appendix B and point them to the loopback IP address. The latter method emulates a technique known as DNS sinkholing [25]. Censorship of hard-coded seeds is achieved by IP blocking using a software firewall.

Our tests reveal that some cryptocurrencies have fallback methods of bootstrapping when the primary method is censored. We identify the fallback bootstrapping methods by first implementing DNS sinkholing or IP blocking and subsequently reviewing the application connection logs, and network packet traffic analyser outputs. Finally, we confirm the fallback bootstrapping method by cross-referencing the relevant source code.

Seven coins are unable to bootstrap based on DNS censorship alone. These coins are Bitcoin Gold, Dogecoin, Zclassic, Syscoin, Cryptonex, Zcoin, and Vertcoin. The first four listed coins are unable to bootstrap because no secondary option of bootstrapping exists–such as hard-coded seeds. For Zclassic, there is a configuration option for a fallback bootstrap mechanism via hard-coded seeds, but no values exist. In the case of Zcoin, and Vertcoin, connection attempts are seen to hard-coded IPs which are verified in

the source code. However, no successful connections are completed after a period of 24 hours.

Five coins are unable to bootstrap based on IP censorship alone. These coins are Ethereum, which also covers 47 underlying tokens, Ethereum Classic, Siacoin, and Bitcore. Therefore, if we consider the underlying tokens, 80% of the cryptocurrencies tested do not have resilient bootstrapping methods configured. Our tests reveal that if DNS censorship is combined with IP censorship, that only Verge is able to bootstrap by default through the use of Tor.

Furthermore, Syscoin is unable to reconnect to the peer-to-peer network when DNS seed censorship is present. DNS censorship should have limited success for peer reconnection because in typical operation previously connected peers should be locally cached [34]. This indicates a persistent dependency on DNS for ongoing connectivity requirements, rather than limiting the dependency on this censorship-prone resource for bootstrapping alone.

Table 2 summarizes the results of this section and the fallback methods of bootstrapping that we discovered.

4.3 Results of Tor Bootstrapping

Reviewing the source code for 25 cryptocurrencies uncovers the option to support connectivity through a proxy server for 20 of the coins. Therefore, we test the ability to bootstrap via Tor. This allows us to explore the rate of success when bootstrapping through a censorship-mitigated method. Verge is not included in testing because it uses Tor by default.

We perform our test by installing the Tor expert bundle or torsocks on the Windows or Ubuntu virtual machines and configure the local SOCKS proxy to channel the cryptocurrency peer-to-peer traffic. We also ensure the local peer caches are cleared to ensure a true bootstrapping experience.

Only 13 coins are able to successfully bootstrap, five are unsuccessful, and two bootstrap but have various issues. The five coins unable to bootstrap are Bitcoin Gold, Zcash, Bytecoin, Dogecoin, and Electroneum. The two coins which bootstrap but have issues are Bitcoin Cash and Monero. For Bitcoin Cash finding other peers to connect to for bootstrapping is successful. However, after 24 hours of operation the estimated time to download the full blockchain through Tor is quoted as one year and 27 weeks, making this method of connectivity infeasible in practice. Monero is able to proxy some of its peer-to-peer traffic via Tor, but the UDP based DNS queries leak outside of the SOCKS proxy and setting the DNS_PUBLIC=tcp option results in several errors with connectivity. The README.md on the 'Using Tor' section of Monero, Electroneum, and Bytecoin explicitly indicate that these coins are not meant to integrate with Tor [4, 7]. Therefore, any issues experienced are within the expectations set by their core developers. We discuss the reasons for this similarity in Section 6.

Although bootstrapping via Tor can mitigate the risk of censorship, it may not provide the levels of anonymity desired [24, 32]. Also, proxied connections must accept latency and performance penalties limiting the ability to download large blockchains in a timely manner. Furthermore, in a highly regulated environment, extra caution is required to avoid the censorship of Tor itself. This includes the use of pluggable transports to obscure the character of Tor traffic and the manual collection of private bridge IP addresses.

Table 2: Cryptocurrency Bootstrap Methods and Connection Results. Tor Option = Tor connectivity configurable, Tor Outcome = connection results of Tor bootstrap, DNS Censor = DNS censorship able to prevent bootstrap, DNS Single = DNS seeds use diverse providers, Config Issues = other issues, ZMap Time = time in mm:ss to discover eight peers, ZMap Bootstrap = bootstrap success, ● = yes, ○ = no, × = N/A, ⊙ = successful bootstrap, ◎ = bootstrap has issues, ⊗ = bootstrap not successful after 24h, ◇ = GUI option for Tor, †= DNS seed configured, yet NXDOMAIN returned, ‡= hard-coding misconfiguration, (#) = number of underlying tokens.

Cryptocurrency	Ticker	Bootstrap	Tor Option	Tor Outcome	DNS Censor	DNS Single	Config Issues	ZMap Time	ZMap Bootstrap
Bitcoin (2)	BTC	DNS Seed Hard-Coded	●	⊙	○	○	○	7:58	○
Ethereum (47)	ETH	Hard-Coded	○	×	×	×	○	×	×
Bitcoin Cash	BCH	DNS Seed Hard-Coded	●	◎	○	○	○	7:58	○
Litecoin	LTC	DNS Seed Hard-Coded	●	⊙	○	○	○	23:43	○
Dash	DASH	DNS Seed Hard-Coded	●	⊙	○	○	○	26:32	○
Monero	XMR	DNS Seed Hard-Coded	●	◎	○	●	†	18:43	○
Eth. Classic	ETC	Hard-Coded	○	×	×	×	○	15:54	○
Bitcoin Gold	BTG	DNS Seed	●	⊗	●	○	○	16:08	○
Zcash	ZEC	DNS Seed	●	⊗	○	○	○	32:15	○
Bytecoin	BCN	Hard-Coded	●	⊗	×	×	○	29:24	○
Verge	XVG	Tor	●	⊙	×	×	○	×	×
Dogecoin	DOGE	DNS Seed	●	⊗	●	●	○	29:29	○
Siacoin	SC	Hard-Coded	○	×	×	×	○	11:42	○
Electroneum	ETN	DNS Seed Hard-Coded	●	⊗	○	●	†	10:59	○
HShare	HSR	DNS Seed Hard-Coded	●	⊙	○	○	†	18:27	○
Zclassic	ZCL	DNS Seed Hard-Coded	●	⊙	●	●	‡	16:41	○
Komodo	KMD	DNS Seed Hard-Coded	●	⊙	○	○	○	6:50	○
Syscoin	SYS	DNS Seed	●	⊙	●	●	○	6:45	○
Digibyte	DGB	DNS Seed Hard-Coded	●	⊙	○	○	○	15:45	○
Cryptonex	CNX	DNS Seed	○	×	●	●	○	7:16	○
Monacoin	MONA	DNS Seed Hard-Coded	●	⊙	○	●	○	14:58	○
Bitcore	BTX	Hard-Coded	●	⊙	×	×	○	11:20	○
Zcoin	XZC	DNS Seed Hard-Coded	●◇	⊙	●	●	○	28:46	○
Vertcoin	VTC	DNS Seed Hard-Coded	●	⊙	●	○	○	7:58	○
SmartCash	SMRT	DNS Seed Hard-Coded	●	⊙	○	○	○	13:32	○

The results in this section are summarized in Table 2. Support cases with the core development teams, and our labs Internet Speed are noted in Appendix A.

4.4 Results of ZMap Bootstrapping

In the previous sections we have surveyed the censorship-prone methods of bootstrapping present in the majority of cryptocurrencies researched, and we have also tested their resilience to basic censorship to determine fallback methods of bootstrapping. We also tested the ability to bootstrap through a censorship-mitigated method via Tor. In this section we test the feasibility of bootstrapping through a censorship-resistant method by using scanning the IPv4 address space using ZMap. The majority of core reference software peers limit their connections to eight other peers. Therefore, on our ZMap scanning parameter we configure the scan to halt as soon as eight peers with the relevant destination TCP ports have been discovered. We first record the amount of time it takes to discover the peers using ZMap, then we test the ability to bootstrap using the discovered peer IPs.

Our results show that the mean time to discover eight peers was 16min 29sec. Surprisingly, none of the cryptocurrencies could bootstrap successfully using this method. As this result was unexpected, we attempted to run a full IPv4 scan on port tcp/8333 (the port used by Bitcoin). The scan took just under four hours to complete and only discovered 241 IPs, none of which allowed Bitcoin to successfully bootstrap. Also, Siacoin has a list of 101 hard-coded seeds configured in the source code, therefore we ran an exhaustive scan to see if any of these seeds were discovered. Regrettably, although 221 IPs were found listening on Siacoins' destination TCP port (with a total scan time exceeding five hours), none of these IPs coincided with the hard-coded seeds. Consequently, bootstrapping was not successful using any of the discovered IPs. Because none of the cryptocurrencies could bootstrap using eight discovered IPs, we also ran a series of exhaustive scans and noted scan times would typically take between four to five hours to complete, making this method infeasible in practise. The scan times would also be impacted by the variance in geographical network latency, noted in Table 3.

We could surmise that one of the challenges impacting this method of bootstrapping was the bandwidth requirements to run a full scan in a timely manner. Furthermore, we noted an issue could arise when destination port numbers were commonly used by other services. For example, Bytecoin uses port tcp/8080, a commonly used port for proxy and web services. Furthermore, Vertcoin, Bitcoin Cash, and Bitcoin all use the same destination port, so finding peers unique to the particular peer-to-peer network would be fraught with issues. Finally, Ethereum uses dynamic ports for peer connectivity, therefore, this method of peer discovery was not compatible with this coin.

Despite the major advantage that this method of bootstrapping requires no centralized resources, our results show that this is not a feasible method of bootstrapping due to the nil success rate, varying discovery times, and the fact that the peer-to-peer traffic is still identifiable by the destination port of its operation. Results for this section can be seen on Table 2.

5 GLOBAL RIPE ATLAS STUDY ON DNS SEEDS

After determining the bootstrapping methods of 25 coins and 49 underlying tokens, we now turn our investigation to the measurements obtained from our global RIPE Atlas DNS seed bootstrapping study. We first provide statistics and insights about the distribution of IP addresses returned from the DNS query responses. Then, we expose several negative results impacting DNS seed bootstrapping, such as outages and query response manipulation.

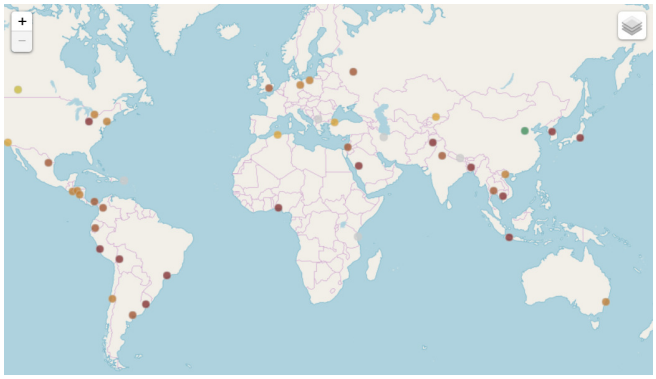

Figure 3: RIPE Atlas Probe Locations. The colours represent the latency associated with each DNS query response. red = high, amber = medium, green = low. The latency times are summarized in Table 3 under the column labelled Latency.

5.1 Detailed Research Methodology of Study

We begin by elaborating on the research steps outlined in Section 3 which are specific to this element of our study.

5.1.1 RIPE Atlas Probe Selection. To begin our RIPE Atlas study, we need to select the RIPE Atlas probes that will be used to perform the DNS queries for our research. Based on a data collection interval of two months, we choose 46 RIPE Atlas probes from 42 countries across Asia Pacific, Europe Middle East Africa, North America, and South America.

In several countries trading and acquiring cryptocurrencies is illegal or is under legal scrutiny [8, 55]. Therefore, where possible, we have selected RIPE Atlas probes from these countries, with the intent of measuring possible DNS response manipulation. A summary of the countries selected, and their legal standpoints towards cryptocurrencies can be seen in Table 3. A longitudinal view of each RIPE Atlas probe in our study can be seen in Figure 3.

5.1.2 DNS Seed Enumeration and Selection. Our next step is to enumerate all the DNS seeds used by the cryptocurrencies. These are noted in the source code and verified on the virtual machines, outlined in Section 4. All 92 DNS seeds can be seen in Table 7 in Appendix B. Only seeds with a 'Yes' or 'Partial' result column R:0 (indicating Return Code 0), of Table 7, are selected for analysis because they are the only responsive seeds.

DNS query responses include Return Codes which indicate the status of the response. The Return Codes seen in the RIPE Atlas study are 0–NOERROR, indicating that the query is successfully completed, 2–SERVFAIL indicating that a server failure has occurred to the DNS query, and 3–NXDOMAIN indicating that the requested domain name does not exist.

Return Code manipulation is a typical DNS censorship technique [18, 54]. For example, if cryptotest1.io was configured with an A

record, the correct query response would include Return Code 0 along with the correct the IP address. However, in a censored environment, the query response may be manipulated to Return Code 3, thus withholding the IP address from the client and ultimately preventing access to the resource. In Section 5.3.1 we outline an example of Return Code manipulation witnessed in our study.

5.1.3 Data Collection Intervals. For the majority of DNS seeds, the RIPE Atlas data collection was conducted between May 6–July 6, 2018. From May 6–June 2, each seed was queried on each probe every 24 hours. We were able to increase the interval to every six hours from June 3–July 6 because we accrued more RIPE Atlas credits to support an increased polling interval for our measurement study.

Due to initial struggles with the verification of the DNS seed bootstrapping process for the coins HShare, Komodo, Zcash and Zclassic, the DNS seeds for these coins were tested between August 12–September 23. The polling interval for these coins was every 6 hours. Instead of excluding these coins from our study, due to issues with source code verification, we opted to perform the same manner of RIPE Atlas data collection on these coins, albeit during a different time period.

5.1.4 Data Processing Steps. The raw data from the DNS queries made from the RIPE Atlas probes is output into a JSON format. We extract the fields relevant to our study using a Python script and output this data into a CSV which is then further processed into an SQL script with the appropriate INSERT statements to input into a MySQL relational database. The data queried from MySQL forms the basis of our measurement study.

5.2 DNS Seed Measurement Results

In this section we present the main results of our global measurement study into DNS seeds used for cryptocurrency bootstrapping. We will capture the statistics surveyed for each cryptocurrency, highlight peers active across multiple coins, and reveal the variance in distinct IPs recorded in countries with differing legal standpoints towards these digital assets.

5.2.1 IP Statistics Per Cryptocurrency. We begin our data analysis by capturing the statistics regarding the number of IP addresses returned for each cryptocurrency. In Figure 4 we record the number of distinct IPs and total number of IPs of the cryptocurrencies that we research. To understand the difference between the set of distinct and total IPs for each cryptocurrency we provide the following example.

Example 5.1. This example uses private IPs, therefore no information about real peers is disclosed.

Assume two DNS queries are made for the DNS seed example-seed.mycryptocurrency.org. The first query response returns the set of IPs $\mathcal{S}_1 = \{10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4\}$, and the second query response returns the set of IPs $\mathcal{S}_2 = \{10.0.0.1, 10.0.0.5\}$. The number of total IPs from the two queries is simply $|\mathcal{S}_1| + |\mathcal{S}_2| = 4 + 2 = 6$. The number of distinct IPs from the two queries is $|\mathcal{S}_1 \cup \mathcal{S}_2| = |\{10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4, 10.0.0.5\}| = 5$.

Our measurement data reveals that the following coins return less than 26 distinct IPs: Monacoin (18), Cryptonex (21), Zcoin (24), and Zclassic (25). Conversely, the remaining coins with the highest number of distinct IP addresses returned are: Bitcoin (44077), Litecoin (7393), Dash (5489), Dogecoin (2812), Bitcoin Cash (2512), Syscoin (1280), Vertcoin (967), Zcash (619), Bitcoin Gold (331), Digibyte (295), Komodo (86), and SmartCash (67).

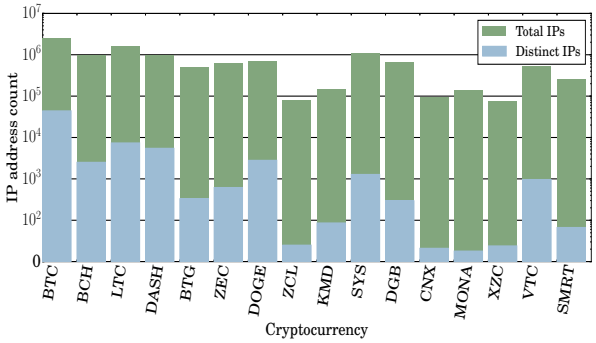

Figure 4: Number of Total and Distinct IPs from RIPE Atlas DNS Seed Query Responses. The y–axis is logarithmic.

For the coins returning less than 26 distinct IPs, the bootstrapping process relies on a static set of servers for connection to the peer-to-peer network. That is, the DNS query responses reveal the IPs of rendezvous servers, rather than the IPs of other peers on the network. Unfortunately, this bootstrapping method reflects the legacy method of how Napster peers connected to the peer-to-peer network, noted in Section 2.2.

The coins returning less than 26 distinct IPs have a disadvantage in terms of being able to withstand basic censorship. However, they inadvertently prevent side-channel information leakage. In contrast, consider coins returning a large set of dynamically changing IPs in the DNS query responses. A clear side-channel is exposed to harvest information about the IPs of peers. This information can be collected without the need to install any core reference client software. That is, DNS query responses reveal the IPs of peers on the network, which is an unintended consequence of DNS seeding.

5.2.2 Peers Active Across Two or More Cryptocurrencies. Our measurement study reveals that 5.2% of the distinct IPs identified are active across two or more cryptocurrencies. We calculate this result as follows: Let C be the set of cryptocurrencies which use DNS seeding as a bootstrap method, where:
$C = \{\text{BTC, BTH, LTC, DASH, XMR, BTG, ZEC, DOGE, ETN, HSR, ZCL, KMD, SYS, DGB, CNX, MONA, XZC, VTC, SMRT}\}$. Let $\mathcal{D}_i$ be the set of distinct IPs returned for each $i \in C$. Our data shows that:

$$\sum_{i \in C} |\mathcal{D}_i| = |\mathcal{D}_{\text{BTC}}| + |\mathcal{D}_{\text{BTH}}| + |\mathcal{D}_{\text{LTC}}| + \ldots + |\mathcal{D}_{\text{SMRT}}| = 66016$$

Also, if we let $\mathcal{I}$ represent the set of distinct IPs in our entire data set, we find that $|\mathcal{I}| = 62732$. This reveals that:

$$\sum_{i \in C} |\mathcal{D}_i| - |\mathcal{I}| = 3284$$

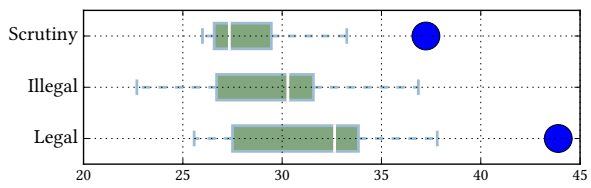

Figure 5: Distinct IP Address Count Statistics of Countries Grouped by Cryptocurrency Legal Status.

That is, 3284 out of the 62732 peers measured are active across two or more cryptocurrencies. IPs active across multiple cryptocurrencies could be individual peers with diversified investments. However, they could be the IPs of cryptocurrency exchanges, which act as a proxy for numerous peers. Identifying and censoring the IPs used by exchanges would have a profound impact to the trade and acquisition of these digital assets for numerous peers.

5.2.3 IP Statistics by Legal Status. The final statistic we capture is the number of distinct IP addresses seen in countries grouped by their legal status towards cryptocurrencies, noted in Table 3. In Figure 5 we see that lower distinct IP counts tend to be in countries where cryptocurrencies are illegal or are under legal scrutiny, and that the higher IP counts belong to countries where cryptocurrency ownership is fully legal.

5.3 DNS Manipulation

In this section we present results regarding evidence of manipulation of the cryptocurrency bootstrapping process based on the reliance on DNS seeding. To maintain privacy, we only disclose the AS number containing the peer IPs identified in our study.

5.3.1 NXDOMAIN Manipulation. In our study we find that three probes inject manipulated results for DNS query responses on Non-Existent Domains. All other probes return NXDOMAIN, but these probes return NOERROR. The ability to manipulate the return code in a DNS query response and present fabricated peer IPs reflects the ability to modify the bootstrapping process. Presenting false peer IPs could isolate the peer from the peer-to-peer network. Worse, manipulated IPs could resolve to a reconnaissance host which records connection attempts to the cryptocurrency peer-to-peer network [44]. This is concerning for peers in countries where cryptocurrencies are illegal, as noted in Table 3.

The behaviour of manipulating DNS query responses in China is well documented [18, 54]. However, we surprisingly record this behaviour in Brazil and in the USA. All the DNS seeds for Monero return NXDOMAIN on all probes, except for China–which is expected. Interestingly, we also see return code manipulation on a probe in Yorkton Heights, USA. The latter probe returns an IP in AS45028 for seeds.moneroseeds.ae.org. The IP is reachable, but it suspiciously does not accept connections to TCP ports 18080 or 28080 on which Monero operates. The China probe returns an IP in AS4837, which is unreachable. It also returns the same IP when other probes report NXDOMAIN for DNS seeds: dnsseed.dashpay.io, node-singapore.cryptonex.org, and seeds.electroneum.com. The Yorkton Heights probe also exhibits the same manipulated NXDOMAIN behaviour on seed1.smartcash.org, seed2.smartcash.org, node-singapore.cryptonex.org, and seeds.electroneum.com. Finally, on the seed singapore.cryptonex.org we see the Brazil host also returning a false NOERROR. Results are summarized in Table 3.

5.3.2 TTL Manipulation. In this section we identify a commonly manipulated parameter in DNS query responses called the Time to Live (TTL) value. We show examples of this manipulation captured in our study and analyse how this manipulation impacts cryptocurrency bootstrapping behaviour.

TTL values tell DNS servers how long to cache query responses in order to reduce network traffic and improve performance. The prevalence of TTL manipulation globally has been noted to occur in 20% of DNS query responses [54]. The main intention behind TTL manipulation is generally motivated by performance considerations.

Example 5.2. This example uses a private IP, ensuring no information about real resources is revealed. Assume a website ttlexample.earth has a single A record referencing IP address 10.0.0.50 with a TTL of 3600s. To reduce DNS query traffic an ISP may manipulate the TTL from 3600s to 86400s. By manipulating the TTL, the IP address for this website is cached for 23 hours longer than the authentic parameter. As a result, DNS query traffic is minimized for this record over a 24-hour period.

For websites returning a static set of IP addresses, TTL manipulation decreases traffic load and increases performance. However, although well intended, TTL manipulation may not be suitable for domains that host dynamic IP content, such as the coins we see in Section 5.2, which return peer IP addresses. Considering the churn rate associated with peer-to-peer networks [42], manipulating TTL values to high values could also lead to the return of stale peer IPs which are no longer active on the network, thereby adversely impacting the bootstrapping process.

During the period of our research, the data reveals that TTL manipulation would have caused peer bootstrapping behaviour changes for: Dogecoin, Bitcoin, Vertcoin, and Bitcoin Gold. For Dogecoin the primary DNS seed seed.multidoge.org returns SERVFAIL thereby returning no peer IP addresses on 93% of our selected probes between June 2–16. Yet on probes in El Salvador, Honduras and Ohio, USA, we still see NOERROR responses between June 2–4 because of overwritten TTL values. Luckily, the Dogecoin secondary seed seed2.multidoge.org remains responsive during this period otherwise bootstrapping would have failed because Dogecoin does not have a diverse bootstrapping mechanism configured.

For Bitcoin, the DNS seed dnsseed.bitcoin.dashjr.org returns a SERVFAIL for all probes during the May 11–June 2, 2018 period. However due to TTL manipulation, the probes in China, El Salvador, Honduras, and Panama continue to return results between May 11–12, and Australia continues to return results between May 11–24. For Bitcoin, outage on this seed dnsseed.bitcoin.dashjr.org is not a major issue impacting bootstrapping behaviour because all other seeds were returning results during this period. Also, Bitcoin provides a large list of hard-coded IP addresses for bootstrapping as a fallback method, as noted on Table 2.

Table 3: Details of RIPE Atlas Probes. In this table we record the country, and region of each probe, the associated legal status towards cryptocurrencies for the country, the number of probes chosen in each country and the latency of the DNS query responses in ms. We also highlight if the probe(s) in that country experienced issues related to common DNS manipulation techniques and other DNS related issues. Abbreviations: NXD = NXDOMAIN manipulation present, TTL = TTL manipulation present, Goog. = Google DNS issues present, ● = yes, ○ = no, † = average, ‡= probe went offline June 28, 2018.

Country	Region	Status	# of probes	Latency < x ms	NXD	TTL	Goog.
Algeria	EMEA	illegal	1	50	○	●	○
Argentina	SAMER	legal	1	30	○	●	○
Australia	APAC	legal	1	20	○	●	○
Bangladesh	APAC	illegal	1	100	○	●	○
Bolivia	SAMER	illegal	1	100	○	○	○
Brazil	SAMER	legal	1	200	●	○	○
Cambodia	APAC	scrutiny	1	800	○	○	●
Canada	NAMER	legal	2	60†	○	○	●
Chile	SAMER	legal	1	40	○	○	○
China	APAC	scrutiny	1	10	●	●	○
Columbia	SAMER	legal	1	200	○	●	○
Ecuador	SAMER	illegal	1	200	○	○	●
El Salvador	SAMER	legal	1	50	○	●	○
Germany	EMEA	legal	1	40	○	○	●
Honduras	SAMER	legal	1	100	○	●	○
India	APAC	scrutiny	1	100	○	○	○
Indonesia	APAC	scrutiny	1	30	○	○	○
Iran	EMEA	illegal	1	100	○	○	○
Israel	EMEA	legal	1	200	○	○	●
Japan	APAC	legal	1	10	○	○	○
Kyrgyzstan	APAC	legal	1	200	○	○	●
Macedonia	EMEA	illegal	1	100	○	●	○
Mexico	NAMER	legal	1	100	○	○	○
Nepal ‡	APAC	illegal	1	100	○	○	○
New Zealand	APAC	legal	1	20	○	○	○
Nicaragua	SAMER	legal	1	100	○	○	●
Nigeria	EMEA	scrutiny	1	100	○	○	○
Pakistan	APAC	scrutiny	1	200	○	○	●
Panama	SAMER	legal	1	200	○	●	●
Peru	SAMER	legal	1	100	○	○	●
Poland	EMEA	legal	1	20	○	○	○
Puerto Rico	SAMER	legal	1	100	○	○	○
Russia	EMEA	scrutiny	1	40	○	○	○
Saudi Arabia	EMEA	legal	1	100	○	○	○
South Korea	APAC	legal	1	200	○	●	○
Tanzania	EMEA	legal	1	10	○	○	●
Thailand	APAC	scrutiny	1	40	○	○	○
Turkey	EMEA	scrutiny	1	200	○	○	●
United Kingdom	EMEA	legal	1	30	○	○	○
Uruguay	SAMER	legal	1	30	○	●	○
USA	NAMER	legal	4	50†	●	●	●
Vietnam	APAC	scrutiny	1	100	○	○	●

We also see the seed dnsseed.pknight.ca for Vertcoin unresponsive between July 4–6. Yet, due to TTL manipulation we continue to see DNS responses returning peer IP addresses in China, El Salvador, Honduras and Panama. At this time Vertcoin was also in a precarious position. It only had one other seed responsive during this period and, as noted in Section 4, the hard-coded IP addresses in Vertcoin's secondary bootstrapping configuration are unreachable. Finally, an outage in Bitcoin Gold DNS seeds, which we cover in more detail in Section 5.4, highlights TTL manipulation in Algeria, Argentina, Bangladesh, China, Columbia, El Salvador, Honduras, Macedonia, South Korea, Uruguay and the USA. In all of these cases, assuming the peer IPs were active, TTL manipulation provided the increased availability of DNS seeds that would have otherwise been unavailable from the DNS servers used by the probes in the countries noted.

However, the ability to manipulate TTL values on cryptocurrency DNS seeds could also provide negative consequences. Manipulating TTL values too high could isolate a bootstrapping peer from the network if all the returned IP addresses belonged to peers no longer active on the network. For example, measurement studies of Bitcoin nodes indicate that the majority of IP addresses are active less than five days [26]. Therefore, for Bitcoin, manipulating a TTL to a value that exceeds five days could return stale peer IPs and interrupt the bootstrapping process. Excessive TTL manipulation would have the most profound impact on the coins in Table 2 which do not have a fallback bootstrapping method. The results of this section are summarized in Table 3.

5.4 Recorded Outages and Issues

In this section we review the DNS seed outages impacting cryptocurrency bootstrapping during our study. We identify an outage impacting Bitcoin Gold and highlight issues specific to Google DNS servers across 14 countries impacting a total of six cryptocurrencies.

5.4.1 Bitcoin Gold Outage. Our measurement study reveals a near miss outage for bootstrapping on Bitcoin Gold between June 22–23, 2018. Bitcoin Gold has two responsive seeds: dnsseed.btcgpu.org and dnsseed.bitcoingold.org. We see that for IPv6 responses, both seeds return SERVFAIL codes. Also for IPv4 responses, the former seed also returns SERVFAIL codes for all probes between 19:00UTC June 22 and 18:00UTC June 23. During this time period a single IPv4 address in AS22612 is returned on the dnsseed.bitcoingold.org seed. Typically, each query response for this seed returns 29 peer IPs. However, during this period, bootstrapping onto the Bitcoin Gold depends on a single IP address. This situation is especially precarious for Bitcoin Gold, as Table 2 indicates DNS seeding is the only bootstrapping option configured for this coin.

5.4.2 Google DNS Issues. On June 30, 2018 we also find that Google DNS servers across 14 countries–noted on Table 3–return no IPv4 addresses across the DNS seeds of six cryptocurrencies: Zcoin, Monacoin, Litecoin, Digibyte, Dash, and Cryptonex. In all cases, a NOERROR response code is provided, but no peer IPs are returned. The issue appears to relate to seeds in the top-level domains .io and .org. Furthermore, for Bitcoin Gold, during the entire data collection interval from May 6–July 6, the exact issue is seen in the same 14 countries using Google DNS. IPv6 records are returned during this

time (with the exception of the outage time noted in Section 5.4). Therefore, Bitcoin Gold peers in these regions using Google DNS with limited IPv6 functionality would struggle to bootstrap under these conditions.

This section has highlighted several negative consequences of using DNS seed bootstrapping for cryptocurrencies. By depending on DNS, we see that bootstrapping is subject to connection manipulation and outages in 60% of the countries that we investigate. The outages recorded relate to operational issues on individual coins and the use of specific DNS providers.

6 UNCOVERING BOOTSTRAPPING INHERITANCE

During our research we concurrently review the source code of 25 cryptocurrency coins and note considerable similarities in the style and syntax of the code. Based on this finding, we complete an analysis of the code repositories to map the software lineage (known as a software forks) of each cryptocurrency researched. This uncovers a chain of inheritance of each source code repository to only five main code bases.

The software forks identify the predominance of Bitcoin source code in over two thirds of the coins. The similarity of bootstrapping methods in the software forks of Bitcoin can be traced to the chainparams.cpp file, seen in Table 6 in Appendix B. The source code reveals that Bitcoin elects to use DNS seeding as the primary bootstrapping method and we see that the core developers of the software forks of Bitcoin merely inherit the legacy bootstrapping mechanisms as an oversight rather than an explicit design decision.

In Figure 6 we summarize our findings of software fork lineage. We reflect that the software forks represent the inheritance of source code which includes suboptimal elements such as censorship-prone bootstrapping. The figure also illustrates any hard forks of the 25 cryptocurrencies that we research, and we refer the interested reader to Appendix C for further details. A total of five main colour hues exist on the diagram representing the original source code parents for the cryptocurrencies: blue–Bitcoin (17), green–Bytecoin (3), orange–Verge (2), brown–Ethereum Classic (2), and yellow–Siacoin (1), where the number in the brackets indicates the number of coins having the same source code lineage. With the exception of Verge, the other four source code parents use legacy bootstrapping methods. Also, the cryptocurrencies Litecoin, Monero, Zcoin, and Zcash have two shades to indicate that they are both software fork children and software fork parents.

To the best of our knowledge, through our measurement study we provide the first insight into the lack of variance in cryptocurrency source code. As well as inheriting bootstrapping methods we note that any other vulnerability disclosure may materially impact coins beyond the individual cryptocurrency being researched.

7 IMPLICATIONS OF FINDINGS

In this section we explore the security and social implications of our findings which are specific to cryptocurrencies and are new with respect to previous measurement and censorship studies [15, 58]. In contrast to the outcomes of censoring file sharing networks, which mainly share digital music and videos, we highlight that the consequences of censoring cryptocurrencies are more profound.

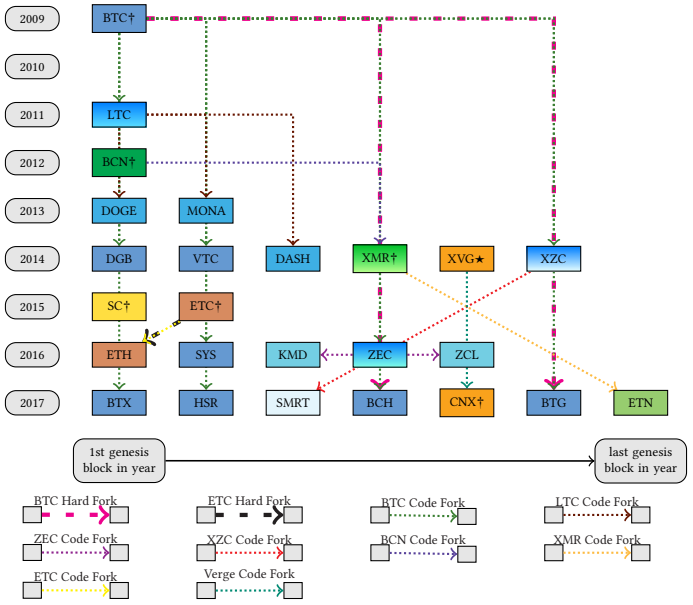

Figure 6: Cryptocurrency Hard Forks and Software Forks. †= configured censorship-prone bootstrapping explicitly, ⋆ = configured censorship-mitigated bootstrapping explicitly, no symbol = inherited bootstrapping method.

7.1 Security Implications of Findings

Based on the results of our measurement study we summarize the security implications discussed throughout the paper. Whilst there already are other Internet measurement and censorship studies, our work is the first to focus on the simultaneous measurement of a large group of cryptocurrencies.

We have highlighted that 95% of cryptocurrencies surveyed use DNS seeding and/or IP hard-coding as methods to bootstrap. Therefore, basic methods of censorship, such as DNS sinkholing or IP blacklisting are effective at preventing peer connectivity to cryptocurrency networks. With regards to DNS seeding, we see that four cryptocurrencies return less than 26 distinct IPs, highlighting the use of rendezvous servers for peer connectivity. With this in mind, the high degree of peer bootstrap centralization leaves these currencies more susceptible to outages and censorship.

Conversely, we see that ten cryptocurrencies return sets of hundreds or even thousands of peer IPs in their DNS query responses, thus revealing an unintended side-channel to harvest information about cryptocurrency peers. The ability to harvest cryptocurrency peer IPs via DNS query responses brings an interesting consideration concerning personal data protection laws, such as the European Union's General Data Protection Regulation (GDPR). GDPR applies to the processing of personal data of individuals within the EU. According to the regulation, if information is provided which enables the identity of an individual behind an IP then it is considered personal data [1]. It is possible to combine the data from time stamped DNS query responses with ISP logs to potentially identify individuals, or groups of individuals using the IP recorded.

Additionally, 32% of cryptocurrencies do not use redundant DNS providers for their DNS seeds, leaving themselves exposed to a single point of DNS service failure. Furthermore, evidence of DNS manipulation is seen due to the lack of DNSSEC on any of the cryptocurrencies surveyed. This could lead to peers in countries where cryptocurrencies are illegal being identified through the use of a reconnaissance host IP injected into the DNS query response. Additionally, due to the churn associated with peer-to-peer networks, TTL manipulation could constrain peer connection due to stale records being cached at ISP DNS servers. Also, 88% of the cryptocurrencies surveyed use distinctive destination ports, leaving them open to basic port-based censorship.

Our study has also revealed that bootstrapping via Tor and ZMap scanning are not always feasible alternatives to 'vanilla' options. This highlights that a defence-in-depth approach to cryptocurrency bootstrapping should be applied. Finally, we reveal the lack of variance in cryptocurrency source code leaves a cascading impact for vulnerability disclosure for coins and tokens that share the same parent source code fork. We discuss both tactical and strategic recommendations to these security implications in Section 8.

7.2 Social Implications of Censorship

In this section we highlight the potentially profound social impacts of cryptocurrency censorship with two examples.

7.2.1 Limitation to Anonymous Funding Streams. Cryptocurrencies are used as anonymous funding streams for various organizations and individuals.

Firstly, we note that Bitcoin donations are a funding stream for the whistleblowing organization WikiLeaks, as well as the investigative reporting platform the Organised Crime and Corruption Reporting Project [11, 13]. Cryptocurrencies are also used to anonymously fund political dissidents such as Alexei Navalny, an open critic of Vladimir Putin. Since 2016, Navalny has received anonymous donations totalling 591 BTC, representing approximately 3 million USD [21]. The Tor project also accepts cryptocurrency donations to support their anonymous browsing software [14].

The motivation to censor cryptocurrencies is also considered to limit the ability to anonymously fund terrorist organizations such as Al-Qassam Brigades and Daesh, and also to disrupt money laundering activities [17, 22, 53, 56][1].

If access to cryptocurrencies was censored, the funding streams to these entities would be disrupted. Hindering funding to investigative journalism, whistleblowing, and anonymity projects like Tor could limit the freedom of the press, thus undermining a key aspect of modern democracy. Furthermore, if political opponents to current governments have their funding streams limited, this undermines the democratic element of fair representation for political beliefs. In contrast, censoring cryptocurrencies may also mitigate terrorist funding sources, and help disrupt money laundering activities. Therefore, the act of censoring these digital assets requires consideration because the consequences of regulation have a more significant impact to democratic ideals than the censorship of file sharing services.

[1] Al-Qassam Brigades is considered a terrorist group by the USA, EU, New Zealand, Australia, and the UK.

7.2.2 Disrupting Fiat Currency Relief. In Venezuela, cryptocurrencies like Bitcoin provide an option for asset 'escape value' where the local currency is under threat from hyperinflation due to civil unrest [46]. Where government banking systems are prone to collapse, cryptocurrencies provide a conduit of asset transfer protecting citizens from eroding currency values. Censoring access to cryptocurrencies would undermine the ability for citizens to escape the impacts of hyperinflation.

8 RECOMMENDATIONS AND CONCLUSIONS

In this section, we provide both tactical and strategic recommendations to improve bootstrapping functionality based on our findings and conclude our paper.

8.1 Tactical Recommendations

Tactical recommendations unfortunately assume the continued use of censorship-prone bootstrapping methods. However, they mitigate obvious design issues until strategic changes can be implemented.

Firstly, the current code bases should be reviewed for any incorrect configurations. We see examples of hard-coded IPs being invalid and unreachable. We also see a lack of diversity in bootstrapping methods. Therefore, the status of hard-coded seeds should be periodically updated, and diverse bootstrapping options should be configured to prevent a single form of censorship limiting access.

The third tactical recommendation we make targets the cryptocurrencies that return a small static set of IP addresses to rendezvous servers rather than returning a large dynamic set of peer IPs. We suggest core developers consider returning peer IPs in DNS query responses rather than the IPs of centralized servers. This recommendation requires consideration as it limits centralization; however it introduces a side-channel to harvest peer IPs.

The fourth recommendation is to ensure that DNS seeding or IP hard-coding is never persistently used post-bootstrapping for peer reconnection. Peer caching should be used to limit the ongoing requirement on DNS for reconnection. Our final recommendation is to ensure that DNS seeds are diversified to at least two DNS providers to mitigate any outages that may occur to a single provider.

8.2 Strategic Recommendations

Strategic recommendations have overheads for implementation, performance, and protocol complexity. However, used in combination, they offer robust mitigations to the unwanted consequences of censorship-prone bootstrapping methods. We suggest these recommendations be included, not as default options, but as easily configurable fallback methods.

Our first strategic recommendation covers the use of DNS seeding as a bootstrapping option. Unfortunately, this recommendation still depends on DNS seeding, but addresses its shortcomings. DNS over TLS (DoT)/DNS over HTTPS (DoH), and DNSSEC provide channel encryption and query response integrity respectively. DoH has the added advantage of mitigating the identification of peer-to-peer bootstrapping traffic, as it runs on TCP port 443. DNSSEC would address the DNS manipulation that we see in Section 5. DoT or DoH would also ensure that the traffic was TCP based, thus

removing the risk of UDP based DNS leakage when using Tor. However, this recommendation poses a large computational overhead to protocol operations due to the requirement to cryptographically sign a large dynamic set of IPs when using DNSSEC.

Our second recommendation would be creating GUI-based options for Tor bootstrapping, similar to what Zcoin offers, and also offer Tor pluggable transport connectivity. This would allow peers the option to have obfuscated access to cryptocurrency peer-to-peer networks. We saw in Section 4.3 that Tor bootstrapping can be a realistic method for connection, but the technical configuration required may make this option prohibitive for several users. As part of this recommendation we also suggest that Tor hidden service .onion addresses be offered for initial connectivity.

In conclusion, we note that in isolation, none of the tactical and strategic recommendations is able to provide a full remedy to the censorship challenges inherent in peer-to-peer bootstrapping and address the social and security implications they cause. However, we know from numerous measurement studies of Internet censorship techniques that the methods employed to block specific categories of traffic are not always effective [15, 58]. Therefore, our measurement study has highlighted that by offering multiple options for bootstrapping, a defence-in-depth approach can be adopted, thus providing resilience for this element of cryptocurrency peer-to-peer functionality. Furthermore, we are optimistic that by displaying the extent of source code inheritance further research will consider this finding when exploring other cryptocurrency vulnerabilities and risk exposure.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous ACM CCS '19 peer reviewers for their helpful insights and recommendations.

REFERENCES

[1] 2016. *GDPR - Personal Data.* Technical Report. European Parliament and Council of the European Union. https://gdpr-info.eu/issues/personal-data.
[2] 2017. Evaluate, possibly revise, and then implement ideas for TLS certificate normalization. https://trac.torproject.org/projects/tor/ticket/7145.
[3] 2018. Bitcoin Developer Documentaton: Technical Glossary - Fork. https://bitcoin.org/en/glossary/fork.
[4] 2018. Electroneum README.md - Using Tor. https://github.com/electroneum/electroneum/blob/master/README.md.
[5] 2018. Hard Fork News. https://cointelegraph.com/tags/hard-fork.
[6] 2018. ICO Launch Calendar. https://coinlauncher.io.
[7] 2018. Monero README.md - Using Tor. https://github.com/monero-project/monero/blob/master/README.md.
[8] 2018. *Regulation of Cryptocurrency Around the World* . Technical Report. Law Library of Congress - Global Legal Research Center. http://www.loc.gov/law/help/cryptocurrency/cryptocurrency-world-survey.pdf.
[9] 2018. Top 100 Cryptocurrencies by Market Capitalization. https://coinmarketcap.com.
[10] 2018. Welcome to RIPE Atlas. https://atlas.ripe.net.
[11] 2019. Donate to WikiLeaks. https://shop.wikileaks.org/donate.
[12] 2019. IRC Networks and Servers. https://www.mirc.co.uk/servers.html.
[13] 2019. OCCRP Donate. https://www.occrp.org/en/donate.
[14] 2019. The Tor Project. https://www.torproject.org.
[15] Giuseppe Aceto and Antonio Pescapé. 2015. Internet Censorship detection: A survey. *Computer Networks* 83 (2015), 381–421.
[16] David Adrian, Zakir Durumeric, Gulshan Singh, and John Halderman. 2014. Zippier ZMap: Internet-wide scanning at 10 Gbps. In *8th USENIX Workshop on Offensive Technologies (WOOT 14).* San Diego, CA, USA.
[17] Ana Alexandre. 2019. Brazilian Police Arrest Suspect for Money Laundering With Bitcoin. https://cointelegraph.com/news/brazilian-police-arrest-suspect-for-money-laundering-with-bitcoin.
[18] Anonymous. 2012. The Collateral Damage of Internet Censorship by DNS Injection. *ACM SIGCOMM Computer Communication Review* 42, 3 (2012), 21–27.
[19] Andreas Antonopoulos. 2017. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies* (2 ed.). O'Reilly Media, Sebastopol, CA, USA.
[20] Samantha Bates, John Bowers, Shane Greenstein, Jordi Weinstock, Jonathan Sittrain, and Yunhan Xu. 2018. Evidence of Decreasing Internet Entropy: The Lack of Redundancy in DNS Resolution by Major Websites and Services. *Working Paper 24317, National Bureau of Economic Research* (2018). https://www.hbs.edu/faculty/Pages/item.aspx?num=53830.
[21] Anna Baydakova. 2019. Russian Opposition Leader Raises Three Million in Bitcoin. https://www.coindesk.com/russian-opposition-leader-raises-3-million-in-bitcoin-donations.
[22] Matthew Beedham. 2018. An American woman has been funding ISIS with Bitcoin. https://thenextweb.com/hardfork/2018/11/27/american-funding-isis-bitcoin.
[23] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized Anonymous Payments from Bitcoin(extended version). http://zerocash-project.org/media/pdf/zerocash-extended-20140518.pdf.
[24] Alex Biryukov and Ivan Pustogarov. 2015. Bitcoin over Tor isn't a Good Idea. In *IEEE Symposium on Security and Privacy.* San Jose, California, 112–134.
[25] Guy Bruneau. 2010. DNS Sinkhole. https://www.sans.org/reading-room/whitepapers/dns/dns-sinkhole-33523.
[26] Joan Antoni Donet Donet, Cristina Pérez-Solà, and Jordi Herrera-Joancomartí. 2014. The Bitcoin P2P Network. In *Bitcoin '14: Proceedings of the 1st Workshopon Bitcoin Research.* Barbados.
[27] Evan Duffield and Daniel Diaz. 2015. Dash: A Payments-Focused Cryptocurrency. https://github.com/dashpay/dash/wiki/Whitepaper.
[28] Jake Frankenfield. 2018. Off-Chain Transactions. https://www.investopedia.com/terms/o/offchain-transactions-cryptocurrency.asp.
[29] Chris GauthierDickey and Christian Grothoff. 2008. Bootstrapping of Peer-to-Peer Networks. In *Proceedings of the International Symposium of Applications and the Internet (SAINT'08).* Turku, Finland, 205–208.
[30] Gearlog. 2010. LimeWire, Napster, The Pirate Bay: A Brief History of File Sharing. https://www.geek.com/gadgets/limewire-napster-the-pirate-bay-a-brief-history-of-file-sharing-1359473.
[31] Tom Goldenberg. 2018. Watch Out Crypto Exchanges, Decentralization Is Coming. https://www.coindesk.com/future-crypto-exchanges-decentralization-coming.
[32] Benjamin Greschbach, Tobias Pulls, Laura M. Roberts, Philipp Winter, and Nick Feamster. 2016. The Effect of DNS on Tor's Anonymity. https://arxiv.org/pdf/1609.08187.pdf.
[33] John M. Griffin and Amin Shams. 2018. Is Bitcoin Really Un-Tethered? https://papers.ssrn.com.
[34] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse Attacks on Bitcoins Peer-to-Peer Network. In *24th USENIX Security Symposium (USENIX Security 15).* Washington, D.C, 129–144.
[35] Dan Howdle and Mark Ashton. 2018. Worldwide broadband speed league 2018. https://www.cable.co.uk/broadband/speed/worldwide-speed-league.
[36] Esther Kim. 2019. Venezuela Bitcoin Trading Record Underscores Fiat Currency Hyperinflation. https://bitcoinist.com/venezuela-bitcoin-trading-record-underscores-fiat-currency-hyperinflation/.
[37] Mirko Knoll, Matthias Helling, Arno Wacker, Sebastian Holzapfel, and Torben Weis. 2009. Bootstrapping Peer-to-Peer Systems using IRC. In *18th IEEE International Workshops Enabling Technologies: Infrastructures for Collaborative Enterprises.* Groningen, Netherlands, 122–127.
[38] Mirko Knoll, Arno Wacker, Gregor Schiele, and Torben Weis. 2008. Bootstrapping in Peer-to-Peer Systems. In *14th IEEE International Conference on Parallel and Distributed Systems.* Melbourne, Australia, 271–278.
[39] Richie Koch. 2018. Here are all the countries where the government is trying to ban VPNs. https://protonvpn.com/blog/are-vpns-illegal.
[40] Matthew Leising. 2017. The Ether Thief. https://www.bloomberg.com/features/2017-the-ether-thief.
[41] Cricket Liu and Paul Albitz. 2006. *DNS and BIND* (5 ed.). O'Reilly Media, Sebastopol, CA, USA.
[42] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. 2005. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Surveys and Tutorials* 7, 2 (2005).
[43] Spandas Lui. 2018. How To Bypass ISP Blocking Of The Pirate Bay And Other Torrent Sites For Free [Updated]. https://www.lifehacker.com.au/2018/06/how-to-bypass-isp-blocking-of-the-pirate-bay-and-other-torrent-sites-for-free.
[44] Abhishek Mairh, Debabrat Barik, Kanchan Verman, and Debasish Jenna. 2011. Honeypot in network security: a survey. In *ACM International Conference on Communication, Computing and Security.* Odisha, India, 600–605.
[45] Srdjan Matic, Carmela Troncoso, and Juan Caballero. 2017. Dissecting Tor Bridges: A SecurityEvaluation of Their Private and PublicInfrastructures. In *Network and Distributed Systems Security Symposium. The Internet Society.* San Diego, CA, USA, 1 – 15.

[46] Sam Meredith. 2019. Bitcoin trading in crisis-stricken Venezuela has just hit an all-time high. https://www.cnbc.com/2019/02/14/venezuela-crisis-bitcoin-trading-volumes-hit-an-all-time-high-.html.
[47] Ali Montag. 2018. Warren Buffett explains one thing people still don't understand about bitcoin. https://www.cnbc.com/2018/05/01/warren-buffett-bitcoin-isnt-an-investment.html.
[48] Ali Montag. 2018. 'Wolf of Wall Street' Jordan Belfort on bitcoin: 'Get out if you don't want to lose all of your money'. https://www.cnbc.com/2018/06/29/wolf-of-wall-street-jordan-belfort-get-out-of-bitcoin.html.
[49] Tyler Moore, Nicolas Christin, and Janos Szurdi. 2017. Revisiting the Risks of Bitcoin Currency Exchange Closure. *ACM Transactions on Internet Technology* 18, 4 (2017).
[50] Peter Nagel. 2018. Psychological Effects during Cryptocurrency Trading. http://www.scriptiesonline.uba.uva.nl/document/659099.
[51] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.
[52] Richard Nieva. 2013. Ashes to ashes, peer to peer: An oral history of Napster. http://fortune.com/2013/09/05/ashes-to-ashes-peer-to-peer-an-oral-history-of-napster.
[53] Helen Partz. 2019. Danish Man Faces Over 4 Years in Prison for Laundering 450K With Bitcoin. https://cointelegraph.com/news/danish-man-faces-over-4-years-in-prison-for-laundering-450k-with-bitcoin.
[54] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. 2017. Global Measurement of DNS Manipulation. In *26th USENIX Security Symposium*. Vancouver, Canada, 427–443.
[55] Allen Scott. 2018. 11 Countries Where Bitcoin Is Still Illegal. https://bitcoinist.com/11-countries-bitcoin-still-illegal.
[56] Brenna Smith. 2019. How To Track Illegal Funding Campaigns Via Cryptocurrency. https://www.bellingcat.com/resources/how-tos/2019/03/26/how-to-track-illegal-funding-campaigns-via-cryptocurrency/.
[57] Andrea Tan and Yuji Nakamura. 2018. Cryptocurrency Markets Are Juicy Targets for Hackers: Timeline. https://www.bloomberg.com/news/articles/2018-06-20/cryptocurrency-markets-are-juicy-targets-for-hackers-timeline.
[58] Michael Carl Tschantz, Sadia Afroz, and Vern Paxon. 2016. SoK: Towards Grounding Censorship Circumvention in Empiricism. In *IEEE Symposium on Security and Privacy*. San Jose, California, 914–933.
[59] Nicolas van Saberhagen. 2013. CryptoNote 2.0. https://github.com/monero-project/research-lab/blob/master/whitepaper/whitepaper.pdf.
[60] Aziz Zainuddin. 2017. Coins, Tokens and Altcoins: What's the Difference? https://masterthecrypto.com/differences-between-cryptocurrency-coins-and-tokens.

A SUPPORT CASES

For Section 4.3 we raise several support cases when we experience connectivity issues when attempting to bootstrap via Tor. Not all support cases have responses, but we are committed to working with support and core development teams regarding our results. The details of the support cases are: Bitcoin Gold has issue 350 opened on GitHub August 29, 2018, Zcash support is emailed on August 23, 2018, Bytecoin support is emailed September 6, 2018, Dogecoin has issue 1147 opened on GitHub August 29, 2018, and Electroneum has case CS-9094 currently in progress which is opened September 6, 2018. The only support case we managed to get a response on was with Electroneum, and the results of our work with their support team are still pending. Also, in Section 4.3 we note that Bitcoin Cash had a quoted blockchain download time of one year and 27 weeks after running the download for 24 hours. To provide context to the quoted download time, we note that the download is performed in our lab which has a tested download speed of 225 Mbps and upload speed of 145 Mbps when not running via a SOCKS proxy and reduces to 5 Mbps download and 3 Mbps upload when running via Tor. Also, our ZMap bootstrapping tests were done in May 2019 from a different lab with 35 Mbps download and 12 Mbps upload speeds. Finally, in Appendix C our research regarding Komodo and its use of the Bitcoin Merkle Root is confirmed on support case 10714 which is raised on their online ticketing system on August 31, 2018.

B TABLES

This section provides four tables that cover the details of our research. In Table 4 we provide information about the tooling used to complete our research. Also, in Table 5 we show the top 100 cryptocurrencies we exported in our selection process. Table 6 provides the SHA256 hash of each cryptocurrency's core reference client executable. Table 6 also provides links to the relevant source code covering the method of bootstrapping for each cryptocurrency and details of the TCP and UDP ports used by the peer-to-peer software. In Table 7 we enumerate all of the DNS seeds determined from our research and provide information about the DNS return codes for each seed.

Table 4: Tooling Used for Research. We note that the version given for RIPE Atlas is based on the authors personal RIPE Atlas probe firmware version. †indicates Ubuntu's Uncomplicated Firewall on version 16.04 of the OS. ‡indicates Windows Defender Firewall on version 10 of the OS. The term 'core ref.' is shorthand for core reference client software.

Tooling	Category	Version	Usage
Baretail	logfile viewer	3.50a	viewing debug / connection logs
Notepad++	text editor	7.5.8	development, log file viewing
MySQL	relational database	5.7.23	processing RIPE Atlas JSON data
Python	programming language	3.5.2	processing RIPE Atlas JSON data
RIPE Atlas	measurement platform	4940	collect DNS response data
TCPView	network activity monitor	3.05	view peer-to-peer connections
Tor	SOCKS proxy server	0.3.3.7	testing Tor bootstrapping
UFW†	software firewall	16.04	test censorship of each core ref.
WDF‡	software firewall	10	test censorship of each core ref.
Wireshark	network traffic debugger	2.6.2	debug network traffic of core ref.

Table 5: Top 100 Cryptocurrencies by Market Capitalization. Symbols: * = tokens transitioning to native 'mainnet' blockchains †= Omni token ‡= NEO token unmarked tokens = ERC20 tokens ⋆ = mineable ERC20 token for illustration.[9].

	non-mineable	mineable
coins	Ripple, NEO, Cardano, Stellar, IOTA, NEM, Qtum, Lisk, Nano, Steem, Stratis, Waves, BitShares, Decred, Ardor, Ark, PIVX, Factom, Byteball Bytes, ReddCoin, GXShares, Neblio, Nxt, Particl, Blocknet	Bitcoin, Ethereum, Bitcoin Cash, Litecoin, Dash, Monero, Ethereum Classic, Bitcoin Gold, Zcash, Bytecoin, Verge, Dogecoin, Siacoin, Electroneum, HShare, Zclassic, Komodo, Syscoin, Digibyte, Cryptonex, Monacoin, Bitcore, Zcoin, Vertcoin, SmartCash
tokens	EOS*, TRON*, VeChain*, Tether†, OmiseGO, ICON*, DigixDAO, Binance Coin, Populous, RChain, Maker, Status, Aeternity, Waltonchain, Augur, 0x, Veritaseum, Revain, Gas‡, KuCoin Shares, Basic Attention Token, Bytom, Zilliqa, Ethos, Loopring, Dragonchain, Golem, Nubulas, QASH, aelf, Polymath, Aion, Dent, Kyber Network, ChainLink, Dentacoin, IOStoken, FunFair, SALT, Kin, Power Ledger, Bancor, Enigma, Pillar, Request Network, Cindicator, MaidSafeCoin, TenX, Quantstamp, SinularityNET	0xBitcoin⋆

Table 6: Cryptocurrency Executable SHA256 Hashes, Code Repositories, Destination Ports and Relevant Lines of Code. In the Executable column, the name is represented in a case sensitive manner exactly as the installations are completed. We do not change the names to ensure that the SHA256 hashes are correct. Symbol: */ = shorthand to represent https://github.com/.

Cryptocurrency	Executable	Version	SHA256 Hash and Open Source Codebase (https://github.com/)	Firewall Dest. Ports and Code Lines
Bitcoin	Bitcoin-qt	0.16.2	23A3C2FD4C33EC5E4391E7E966CE1FCCA5EF374C5932DA352D9FC418468D6CEA	TCP: 8333, 9051
			DNS Seed */Bitcoin/Bitcoin/blob/master/src/chainparams.cpp	127–138
			Hard-Coded */Bitcoin/Bitcoin/blob/master/contrib/seeds/nodes_main.txt	1–1525
Eth. Classic	geth	5.5.1	F2FB63CFE81104F05CC3DBF42AA1FD0FA3AC0455362FD1E22519AEBF6847BCF3	TCP/UDP: 8545, 30301-30307
			Hard-Coded */ethereumproject/go-ethereum/wiki/Connecting-to-the-network	N/A
Bitcoin Gold	Bitcoin-qt	0.15.1	0EE01DB46D71A8C19B1540DA50D28D29C3CDD03899BFCFF79860E175C58CC6B3	TCP: 8338
			DNS Seed */BTCGPU/BTCGPU/blob/master/src/chainparams.cpp	174–193
ZCash	zcld	1.1.2	1636240D9781C9AA1AB143BD4CA93F09F4F5610A6D1DC7A33D8513A9954F9073	TCP: 8233
			DNS Seed */zcash/zcash/blob/8df0....6243/src/chainparams.cpp	137–141
Bytecoin	bytecoind	3.2.2	A3460AA6F536B0D8C1A8BB98C51592D44A5294B97C89A7ED17BBC22D8E0B4045	TCP: 8080
			Hard-Coded */bcndev/bytecoin/blob/master/src/CryptoNoteConfig.hpp	125–126
Verge	VERGE-qt	4.0.2.0	5EEFA37A574A4D36A0B5656A11297ED4A42FEFB80D3BF2B53079DC3A581402D3	TCP: 9050–Tor
			Tor */vergecurrency/VERGE/blob/master/src/net.cpp	1046–1059
Dogecoin	dogecoin-qt	1.10.0	D7050A58A53522AF8AA176262F24378B7F7B7330A2D697C0AFFF0A65244555B2	TCP: 22556
			DNS Seed */dogecoin/dogecoin/blob/master/src/chainparams.cpp	123–126
Siacoin	siad	1.3.3	DB37321482B0E2BB6161F42C95C245A5DB541EFD1EDB5B0EED261A50A3914625	TCP/UDP: 9981, 9982
			Hard-Coded */NebulousLabs/Sia/blob/fa3e....4366/modules/gateway.go	15–129
Electroneum	electroneumd	2.1.0.0	32AE2E24DF7265D3EF87067959E71B53371B00842773C6AE8B1ABDF090D862A1	TCP: 3333, 5555, 7777
			DNS Seed */electroneum/electroneum/blob/master/src/p2p/net_node.h	134–136
			Hard-Coded */electroneum/electroneum/blob/master/src/p2p/net_node.inl	407–430
HShare	hshare-qt	2.1.0.0	4C26861E991FF492E2430D2D7C6BCD0788E30D3ECD04587DBEB73F3BED7F3E00	TCP: 7433, 11616
			DNS Seed */HcashOrg/Hshare/blob/master/src/chainparams.cpp	115–117
			Hard-Coded */HcashOrg/Hshare/blob/master/share/seeds/nodes_main.txt	388–420
Zclassic	zcld	1.0.10	A36CF337560EBF096233F55335045982BD81F8E7A47B490B8B9C200840E7975E	TCP: 8133
			DNS Seed */z-classic/zclassic/blob/master/src/chainparams.cpp	100–114
			Hard-Coded */z-classic/zclassic/blob/master/contrib/seeds/nodes_main.txt	1–4
Komodo	komodod	1.0.15	6D48EFFB8172FD3C308A9D5268FF8A29C1DE47EE38531AD5C1585AE83684A231	TCP: 7771, 7778
			DNS Seed */KomodoPlatform/komodo/blob/master/src/chainparams.cpp	163–167
			Hard-Coded */KomodoPlatform/KomodoPlatform/blob/master/iguana/dpow/dpow_network.c	25–39
Syscoin	syscoin-qt	3.0.6.2	6C42EE3412311106C6618744AE4A351B69B72CBACD126149A6BC88D169EA0DFB	TCP: 8001, 8369
			DNS Seed */syscoin/syscoin/blob/master/src/chainparams.cpp	242–245
Digibyte	digibyte-qt	6.16.2	D640973832EDFDC1E41FEF9DB64612ED559C5CF6F6559624D7C7C4A5BC2DC950	TCP: 9051, 12024
			DNS Seed */digibyte/digibyte/blob/master/src/chainparams.cpp	207–214
			Hard-Coded */digibyte/digibyte/blob/master/src/chainparamsseeds.h	1–59
Cryptonex	cryptonex-qt	1.0.0.0	8B5670006E4FB395CC17DC396875CCCB360AF850CA54FFF52B1C0E625C3819E6	TCP: 20863
			DNS Seed */Cryptonex/source/blob/9a76....ae10/src/net.cpp	1227–1239
Monacoin	monacoin-qt	0.15.1	637DA8B48EB23E84DE5C48CBBA823A8D5322B8066C6F2A5C05F0D3A9C6DD6DFA	TCP: 9401
			DNS Seed */monacoinproject/monacoin/blob/master-0.15/src/chainparams.cpp	236–239
			Hard-Coded */monacoinproject/monacoin/blob/01c6....b504/src/chainparamsseeds.h	1–27
Bitcore	bitcore-qt	0.15.1.0	271A265C61B7F38A66C4C2D578BA760BD6B87F9E1FE107C81494340479787A8C	TCP: 8555
			Hard-Coded */LIMXTEC/BitCore/blob/0.15/src/chainparams.cpp	134–138
Zcoin	zcoin-qt	0.13.6.6	69E350F52D83CC59ECD9820793D546332A3160B8C509AF428BF5241863B2E507	TCP: 8168, 8888, 18168, 18888
			DNS Seed */zcoinofficial/zcoin/blob/master/src/chainparams.cpp	169–173
			Hard-Coded */zcoinofficial/zcoin/blob/master/contrib/seeds/nodes_main.txt	1–9
Vertcoin	vertcoin-qt	0.13.2	73C695F8C417097827C2B8121751FE9DF61E0D88CE40EDBE0636739077ACA016	TCP: 8333, 5889, 60021, 6970
			DNS Seed */vertcoin-project/vertcoin-core/blob/master/src/chainparams.cpp	127–133
			Hard-Coded */vertcoin-project/vertcoin-core/blob/master/src/chainparamsseeds.h	1–36
SmartCash	smartcash-qt	1.2.4	867A940D767E239DB2C937B3478C1F7A95E6D1A72C4D946831C9102850761BEA	TCP: 8333, 5889, 60021, 6970
			DNS Seed */SmartCash/Core-Smart/blob/032d....3df1/src/chainparams.cpp	148–157
			Hard-Coded */SmartCash/Core-Smart/blob/e0fe....6429/contrib/seeds/nodes_main.txt	1–10

Table 7: 92 DNS Seeds from source code. RCODES: 0 NOERROR, 1 SERVFAIL, and 3 NXDOMAIN. Symbols †= measurements started August 2018, unmarked = measurements started May 2018, ⋆ = resolves IPv4 and IPv6, ✓ = part of RIPE Atlas data collection as majority of probes return NOERROR, ∗ = RCODE in May 2018 different in August 2018. ● Yes ○ No ◐ Partial.

Cryptocurrency	DNS Seed	R:0	R:1	R:3	Cryptocurrency	DNS Seed	R:0	R:1	R:3
Bitcoin	seed.bitcoin.sipa.be⋆	●	○	○	Zclassic†	as1/2/3.zclassic.org	●	○	○
Bitcoin	dnsseed.bluematt.me⋆	●	○	○	Zclassic†	as2.zclassic.org	●	○	○
Bitcoin	dnsseed.bitcoin.dashjr.org⋆	●	○	○	Zclassic†	as3.zclassic.org	●	○	○
Bitcoin	seed.bitcoinstats.com⋆	●	○	○	Komodo†	seeds.komodoplatform.com	●	○	○
Bitcoin	seed.bitcoin.jonasschnelli.ch⋆	●	○	○	Komodo†	static.kolo.supernet.org	●	○	○
Bitcoin	seed.btc.petertodd.org	◐✓	◐	○	Komodo†	dynamic.kolo.supernet.org	●	○	○
Bitcoin	seed.bitcoin.sprovoost.nl⋆	●	○	○	Syscoin	seed1.syscoin.org	●	○	○
Bitcoin Cash	seed.bitcoinabc.org	○	●∗	○	Syscoin	seed2.syscoin.org	●	○	○
Bitcoin Cash	seed-abc.bitcoinforks.org⋆	◐✓	◐	○	Syscoin	seed3.syscoin.org	◐✓	◐	○
Bitcoin Cash	btccash-seeder.bitcoinunlimited.info⋆	●	○	○	Syscoin	seed4.syscoin.org	◐✓	◐	○
Bitcoin Cash	seed.bitprim.org	●	○	○	Digibyte	seed1.digibyte.io	●	○	○
Bitcoin Cash	seed.deadalnix.me⋆	●	○	○	Digibyte	seed2.digibyte.io	●	○	○
Bitcoin Cash	seeder.criptolayer.net	◐	◐	○	Digibyte	seed3.digibyte.io	●	○	○
Litecoin	seed-a.litecoin.loshan.co.uk⋆	◐✓	◐	○	Digibyte	seed.digibyte.io	●	○	○
Litecoin	dnsseed.thrasher.io	●	○	○	Digibyte	digihash.co	●	○	○
Litecoin	dnsseed.litecointools.com	●	○	○	Digibyte	digiexplorer.info⋆	●	○	○
Litecoin	dnsseed.litecoinpool.org⋆	●	○	○	Digibyte	seed.digibyteprojects.com	●	○	○
Litecoin	dnsseed.koin-project.com	●	○	○	Cryptonex	node-london.cryptonex.org	●	○	○
Dash	dnsseed.dash.org	◐✓	◐	○	Cryptonex	node-frankfurt.cryptonex.org	●	○	○
Dash	dnsseed.dashdot.io⋆	●	○	○	Cryptonex	node-amsterdam.cryptonex.org	●	○	○
Dash	dnsseed.masternode.io⋆	●	○	○	Cryptonex	node-toronto.cryptonex.org	●	○	○
Dash	dnsseed.dashpay.io	◐	○	◐	Cryptonex	node-singapore.cryptonex.org	◐∗	○	◐∗
Monero	seeds.moneroseeds.ch	◐	○	◐	Cryptonex	node-paris.cryptonex.org	●	○	○
Monero	seeds.moneroseeds.li	◐	○	◐	Cryptonex	node-bangalore.cryptonex.org	●	○	○
Monero	seeds.moneroseeds.ae.org	◐	○	◐	Monacoin	dnsseed.monacoin.org	●	○	○
Monero	seeds.moneroseeds.se	◐	○	◐	Zcoin	sf1.zcoin.io	●	○	○
Bitcoin Gold	eu-dnsseed.bitcoingold-official.org	○	○	●	Zcoin	sf2.zcoin.io	●	○	○
Bitcoin Gold	dnsseed.bitcoingold.org⋆	●	○	○	Zcoin	london.zcoin.io	●	○	○
Bitcoin Gold	dnsseed.btcgpu.org⋆	●	○	○	Zcoin	singapore.zcoin.io	●	○	○
Zcash†	dnsseed.z.cash	●	○	○	Zcoin	nyc.zcoin.io	●	○	○
Zcash†	dnsseed.str4d.xyz⋆	●	○	○	Vertcoin	useast1.vtconline.org∗	○	●	○
Zcash†	dnsseed.znodes.org⋆	●	○	○	Vertcoin	vtc.gertjaap.org∗	○	●	○
Dogecoin	seed.dogecoin.com	○	●	○	Vertcoin	seed.vtc.bryangoodson.org	○	●	○
Dogecoin	seed.multidoge.org	●	○	○	Vertcoin	dnsseed.pknight.ca	●	○	○
Dogecoin	seed2.multidoge.org⋆	●	○	○	Vertcoin	seed.oroftheta co.org∗	○	●	○
Dogecoin	seed.doger.dogecoin.com	○	●	○	Vertcoin	seed.alexturek.org∗	○	●	○
Electroneum	seeds.electroneum.com	○	○	●	Vertcoin	vertcoin.mbl.cash	●	○	○
HShare†	hshare-dns1.h.cash	◐	○	◐	SmartCash	seed.smrt.cash	●	○	○
HShare†	hshare-dns2.h.cash	◐	○	◐	SmartCash	seed1.smrt.cash	●	○	○
HShare†	hshare-dns3.h.cash	◐	○	◐	SmartCash	seed2.smrt.cash	●	○	○
Zclassic†	na1.zclassic.org	●	○	○	SmartCash	seed1.smartcash.org	◐	○	◐∗
Zclassic†	na2.zclassic.org	●	○	○	SmartCash	seed2.smartcash.org	◐	○	◐∗
Zclassic†	na3.zclassic.org	●	○	○	SmartCash	seed.smartcash.cc	●	○	○
Zclassic†	eu1.zclassic.org	●	○	○	SmartCash	seed2.smartcash.cc	●	○	○
Zclassic†	eu2.zclassic.org	●	○	○	SmartCash	seed3.smartcash.cc	●	○	○
Zclassic†	eu3.zclassic.org	●	○	○	SmartCash	seed4.smartcash.cc	●	○	○

C SOFTWARE FORKS AND HARD FORKS

We provide further details about the hard forks and software forks noted in Figure 6. A description of Genesis Blocks and Merkle Roots is given to understand the nature of cryptocurrency hard forks and how they differ from software forks. We recall that mapping the

Bitcoin

```
assert(consensus.hashGenesisBlock == uint256S
("0x000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f"));
assert(genesis.hashMerkleRoot == uint256S
("0x4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"));
```

Bitcoin Cash

```
assert(consensus.hashGenesisBlock == uint256S
("000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f"));
assert(genesis.hashMerkleRoot == uint256S
("4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"));
```

Komodo

```
assert(consensus.hashGenesisBlock == uint256S
("0x027e3758c3a65b12aa1046462b486d0a63bfa1beae327897f56c5cfb7daaae71"));
assert(genesis.hashMerkleRoot == uint256S
("0x4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"));
```

Figure 7: Genesis Blocks and Merkle Roots of Bitcoin, Bitcoin Cash and Komodo.

software forks on the cryptocurrencies allows us to highlight the lack of variance between the underlying source code. This allows other security researchers to expand the scope of vulnerability disclosure for individual cryptocurrencies which share a source code base with other coins. Hard forks of cryptocurrencies occur when a blockchain permanently diverges into two separate chains at a particular block number. Cryptocurrencies which are hard forks of a parent cryptocurrency must have the same Genesis Block which includes the initial Merkle Root of the parent [19]. The latter is a strict cryptographic requirement which ultimately relates to the inception point of the blockchain construction. We note that software forks have no strict requirements for cryptographic immutability. Instead, they are simply source code repository forks, most commonly seen on the GitHub platform. For a discussion on the different types of forks, please refer to [3].

C.1 Komodos Genesis Block and Merkle Root

Interestingly, we see that although Komodo is a software fork of Zcash, it seems to have created its Genesis Block prior to Zcash, and it is the only coin to have the property of front running its parent in this manner. In fact, Komodo has another interesting property not captured by Figure 6. Komodo opted to use the initial Merkle Root that Bitcoin used. A blockchain explorer actually indicates that because the initial Merkle Root for Komodos Genesis Block was Bitcoins, the timestamp is recorded as 2009-01-03. Therefore, we counted the subsequent block in the blockchain as its Genesis Block, which occurred on 2016-09-13. In Figure 7 we provide an extract from the chainparams.cpp files hosted on GitHub for Bitcoin, Bitcoin Cash and Komodo. We see that Bitcoin Cash is a hard fork of Bitcoin because the Genesis Block and Merkle Root are identical. However, we see that Komodo is not a hard fork of Bitcoin because only the Merkle Root is the same. The Bitcoin Genesis Block is made up from the double iterated SHA256 hash output of the original Merkle Root 0x4a5e1e4b...afdeda33b but is also concatenated with other fields such as the timestamp, nonce, and version. So, although Komodo shares the same Merkle Root as Bitcoin it does not share the same Genesis Block because the latter mentioned fields are not the same as those used by Bitcoin. Therefore, it is clear from the properties of hashing functions that when the different fields are concatenated onto the Bitcoin Merkle Root that the resulting Genesis Block for Komodo outputs a different hash value to Bitcoin.