New Trains, Same Rails: Fingerprinting Cryptocurrency Software via Network Requests

Muhammad Muzammil*, Oleksii Starov†, Zane Ma‡, Nick Nikiforakis*
*Stony Brook University †Palo Alto Networks ‡Oregon State University
{mmuzammil, nick}@cs.stonybrook.edu ostarov@paloaltonetworks.com zane.ma@oregonstate.edu

***Abstract*—Blockchain technologies are widely promoted as enabling privacy and censorship resistance through decentralization. In practice, however, these properties are shaped not only by the underlying cryptographic protocols, but also by the network-facing behavior of the client software through which users access blockchain networks. We present the first systematic study of DNS resolution behavior in cryptocurrency client software, showing that this behavior produces application-specific fingerprints observable by passive network intermediaries. We characterize these fingerprints across 303 versions of 40 applications and validate their real-world observability using global passive DNS records and enterprise URL filtering logs from a major network security provider. Our results show that DNS resolutions during routine cryptocurrency application use exhibit application-specific patterns that allow intermediaries to identify the application in use and infer sensitive user behavior. These findings reveal a gap between the privacy expectations commonly associated with blockchain-based applications and the passive visibility they afford to network operators.**

***Index Terms*—DNS, cryptocurrency, privacy, censorship**

I. Introduction

Digital services today predominantly rely on infrastructure controlled by centralized service providers. For example, Internet service providers (ISPs) provide Internet connectivity and payment processors mediate financial transactions. This concentration of control enables them to determine which services users can access, what data is collected about them, and how their behavior is monetized [26]. Web3 has emerged as an alternative to this model. It aims to shift this control away from centralized intermediaries and toward decentralized blockchain-based systems such as cryptocurrency. At its core, Web3 promises users the ability to transact directly on blockchain networks, free from the centralized intermediaries that collect and monetize their data [5].

Yet, Web3 users rarely interact with blockchains directly in practice. Instead, they rely on client-side applications installed on their devices to mediate access to blockchain networks. Despite their role in decentralized systems, these applications remain coupled to traditional Internet infrastructure, as their core functionality depends on access to external services such as RPC providers or price feeds.

Consequently, the effective privacy boundary for Web3 users extends beyond the cryptographic guarantees of blockchain protocols. It is also shaped by the network-facing behavior of the client software through which these systems are accessed. DNS plays a central role among these dependencies, as every domain an application contacts must first be resolved through third-party resolvers. These resolvers are operated by centralized entities such as ISPs, campus administrators, and enterprise network operators. As a result, these parties can track and block Web3-related activity without the user's awareness. This dependency thus reintroduces the very centralized control that Web3 seeks to eliminate.

These risks are particularly acute for users who rely on ISP-controlled DNS resolvers or are in countries or sensitive enterprise environments where DNS is subject to surveillance. Even when users configure their devices to use DNS-over-HTTPS (DoH) resolvers, their queries are often downgraded to plaintext DNS or forcibly redirected through resolvers controlled by network operators [6], [12], [18]. At the same time, this visibility may serve legitimate purposes, such as detecting unauthorized cryptocurrency mining on campus networks [23].

Prior work has examined related privacy risks at the application layer. Specifically, they report that website operators can fingerprint Web3 browser extensions to detect cryptocurrency users, and popular websites such as `tiktok.com` have been shown to engage in this practice at scale [24], [27]. However, the risks arising from Web3 applications' reliance on DNS infrastructure are not well understood.

In this paper, we investigate how DNS infrastructure enables the observation and attribution of activity generated by Web3-based applications. To this end, we curate an application suite comprising 303 versions of 40 popular applications spanning a range of user devices, with a primary focus on cryptocurrency-related functionality. We exercise these applications in network-isolated environments while recording all emitted DNS queries to extract their *DNS fingerprints*.

We show that these applications heavily rely on DNS infrastructure for routine operation. They produce distinctive DNS emissions for 34 (85%) applications that can be reliably attributed to active application use. In 18 (45%) cases, this attribution extends to specific application versions. We further observe that 90% of the IP addresses associated with representative domains used by these applications are routed through only five Autonomous Systems. These concentrated dependencies constitute both single points of failure and potential choke points for censorship of Web3 services.

We validate the real-world observability of these DNS fingerprints through a collaboration with Palo Alto Networks, by evaluating their prevalence in aggregate telemetry derived

ISBN 978-3-903176-82-9

from enterprise URL filtering logs. Among other findings, we infer over 74K instances of transaction-related activity using DNS alone. These instances are generated by the MetaMask browser extension and mobile application, the most widely used digital wallet, and originate from approximately 2K customer networks.

Our main contributions can be summarized as follows:

- We present the first study of DNS-level fingerprintability in cryptocurrency client software, demonstrating that passive network observers can identify applications in use and infer user behavior.
- We develop network-isolated experimental setups to extract application-specific DNS fingerprints across 40 applications and 303 versions, and validate their real-world observability using passive DNS and enterprise-scale URL filtering telemetry spanning thousands of customer networks.
- We release our DNS fingerprint dataset and measurement tools to support further research in this area [1].

II. DNS Observability and Threat Model

We consider a passive network observer operating at the DNS resolver level, with visibility into DNS queries issued by client devices during normal application use. In practice, this observer may correspond to an enterprise network operator, campus administrator, or ISP that operates or controls the DNS infrastructure used by those devices. In enterprise, educational, and government networks, DNS resolution is routinely centralized and logged for security monitoring and incident response [6], [18]. Similar observability conditions also arise in residential networks, where users typically rely on ISP-operated resolvers.

Let a denote an application executing over a time interval T, and let $\mathcal{Q}$ denote the set of domain names observed in DNS queries collected by a network observer. We define the *DNS footprint* of a as the set $\mathcal{D}_a = \{d_1, d_2, \ldots, d_n\}$ of fully qualified domain names resolved by a during T. We further define the *DNS fingerprint* $\mathcal{F}_a \subseteq \mathcal{D}_a$ as the subset of domains unique to a (Section V). The presence of domains in $\mathcal{F}_a \cap \mathcal{Q}$ enables the observer to attribute the observed traffic to a, to a specific version v of a, or to particular user actions within a.

Under this threat model, network operators may enforce the use of organization-controlled DNS resolvers and restrict or disable direct access to third-party DNS-over-HTTPS (DoH) services, such as those operated by Google or Cloudflare, in order to preserve resolver-level visibility for security monitoring [6], [18]. In some network environments, encrypted DNS traffic may also be intercepted and downgraded to plaintext DNS, as observed in prior measurement studies [12]. However, our analysis does not rely on downgrade behavior, as resolver-side visibility into queried domain names persists even when DNS traffic between clients and resolvers is encrypted. Network observers may also enforce short DNS cache TTLs to ensure that application-specific domain resolutions are repeatedly observable whenever applications invoke functionality that triggers them.

III. Application Suite

To obtain a comprehensive view of how cryptocurrency applications can be fingerprinted, we curate a diverse suite of applications spanning five categories listed in Table I. Applications are selected based on their popularity within the Web3 user community, as reflected by download statistics and recommendations from forums and review platforms.

The resulting dataset captures a broad range of functionality and use cases within the cryptocurrency ecosystem. It includes lightweight wallets that focus on basic asset storage and transfer for a single cryptocurrency (e.g., Eternl for ADA), as well as multi-currency wallets that support a wide range of assets and advanced features such as staking or token swapping (e.g., MetaMask). In addition to digital wallets, the dataset also includes specialized security-focused applications (e.g., Web3 Antivirus), which are designed to monitor user activity and warn against interactions with malicious contracts or addresses. We further include hardware wallet ecosystems, which consist of physical devices used to securely store cryptocurrency keys and are commonly paired with companion desktop applications for device configuration and asset management. Finally, our dataset covers cryptocurrency mining software, including firmware deployed on specialized mining hardware, which enables participation in blockchain consensus through computational work.

To study how application behavior evolves over time, we collect multiple versions of each application, ranging from the earliest available releases to the most recent versions as of mid-2025. While current releases are readily obtainable from official app stores and vendor websites, historical versions often require additional effort to source. We therefore rely on third-party archives, such as APKMirror for mobile applications, the Wayback Machine for desktop software, and GitHub release pages for open-source projects. Table I presents the complete set of applications included in our study, together with the earliest and latest versions analyzed for each application, the total number of versions examined, and user counts where available. In total, we study 40 distinct applications, encompassing 303 versions released between 2017 and 2025.

IV. Experimental Setup

To accurately capture the outbound DNS requests generated by each application, we design per-software-class experimental setups that enforce strict isolation from background processes and unrelated network activity. This isolation ensures that all observed traffic can be confidently attributed to the application.

After preparing the isolated environment, we launch each application and exercise its functionality using a combination of automated scripts and manual interactions. For cryptocurrency mining firmware, we deploy each firmware image on mining hardware, connect it to a mining pool, and initiate Bitcoin mining, during which we observe the resulting DNS activity. For the remaining four software classes, which are primarily user-facing, each experimental run is structured into a sequence of interaction phases. First, the application is launched without user input to capture initialization-related

Category	Application	Oldest	Latest	# Versions	# Users
Browser Extension	MetaMask	2.10.0	12.13.0	11	13M+
	Phantom Wallet	0.13.1.0	25.9.1		5M+
	Trust Wallet	0.0.1.0	2.32.1		1M+
	Keplr Wallet	0.9.8.0	0.12.201		1M+
	Coinbase Wallet	2.3.5.0	3.107.0		1M+
	Solflare Wallet	1.23.1.0	1.74.4		800K+
	Bitget Wallet	0.0.9.0	2.15.18		400K+
	Xverse Wallet	0.1.7.0	0.50.0		300K+
	Argent	4.8.3.0	5.23.1		300K+
	Leap Wallet	0.6.0.0	0.18.5		200K+
	Nightly Wallet	0.1.3.0	1.16.7		100K+
	Eternl	1.3.3.0	1.12.24		100K+
	Pocket Universe	0.0.47.5	0.6.5		100K+
	Aegis Web3	0.0.6.0	0.6.37		100K+
	Crypto.com	1.2.16.0	2.23.0.0		90K+
	Core Wallet	1.31.0.0	1.97.1		60K
	Taho Wallet	0.8.1.0	0.63.1		50K+
	AlphaOS	0.0.1.0	0.31.6		40K+
	Web3 Antivirus	0.2.1.0	0.17.3		20K+
	Typhon Wallet	0.0.9.0	3.2.16		10K+

Category	Application	Oldest	Latest	# Versions	# Users
Mobile Application	Trust Wallet	1.11.1	8.40.1	6	50M+
	Bitcoin.com Wallet	6.6.3	8.19.2	5	10M+
	Blockchain.com	1.2	8.5.4	8	10M+
	Crypto.com	1.71.0	1.78.0	2	10M+
	MetaMask	2.5.0	7.44.0	5	10M+
	OKX	6.0.40	6.114.0	3	10M+
	Phantom	24.24.0	25.12.0	2	10M+
	Kraken	1.37.0	3.11.0	3	5M+
Desktop Application	Atomic Wallet	2.11.0	2.91.5	4	5M+
	Exodus	24.1.29	25.13.3	3	1M+
	Coinomi	1.3.0	1.3.0	1	N/A
	Guarda	1.0.0	1.0.20	4	N/A
	Sparrow Wallet	0.9.2	2.1.3	6	N/A
Hardware Wallet Application	Ledger Live	2.47.0	2.106.0	8	7M+
	Trezor Suite	20.10.1	24.5.3	5	2M+
	Jade	0.0.1	2.0.23	6	N/A
Cryptomining Firmware	HiveOS	1.00	1.03	4	2M+
	Braiins OS	19.06	22.08.1	4	N/A
	cpuminer	2.4.1	2.4.5	3	N/A
	Antminer	2017	2019	3	N/A

TABLE I: *Cryptocurrency software analyzed in this study. For each application, we report the oldest and latest versions included in our measurements, the number of tested versions, and approximate user counts when available.*

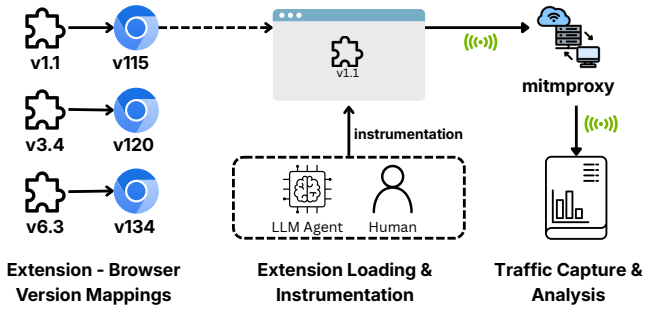

Fig. 1: *Experimental setup for analyzing browser extensions. Each extension version is executed in an isolated Chromium instance with all outbound traffic routed through an mitmproxy instance for traffic interception.*

DNS traffic. Second, we perform onboarding actions such as accepting terms of service or importing an existing wallet. Third, we navigate the application interface to trigger additional features and background functionality. Finally, for digital wallet applications, we initiate a small on-chain transfer (e.g., 0.001 ETH) to trigger transaction-related DNS requests. During each run, all outbound traffic generated by the application is recorded at the isolation boundary. These traces form the raw dataset from which we extract DNS fingerprints.

A. Browser Extensions

Figure 1 illustrates the execution environment used for our browser extension experiments. We maintain a local archive of versioned browser extension packages and Chromium/ChromeDriver binaries. As our dataset includes extension versions dating back to 2017, we require browser versions that are compatible with those historical builds. Newer Chromium releases may not load or behave correctly with older extension APIs, while older Chromium releases may not support newer extensions. For each extension version, we select the Chromium release whose publication date is closest to that of the extension version and pair it with the corresponding driver. Each extension version is then launched in its corresponding Chromium build with the extension preloaded. To further isolate experimental runs, we start Chromium with a fresh sandboxed user profile for each run, preventing state leakage across experiments and across extension versions. We also record and filter out baseline traffic from Chromium instances with no extensions installed, which separates browser-level background noise from extension-induced traffic.

To trigger extension activity in the instrumented browser, we employ an LLM-driven web automation agent integrated with a browser automation framework to execute high-level navigation instructions [17]. The agent uses OpenAI's GPT-4o model to automate onboarding workflows that are largely consistent across browser extensions, including account registration, login, acceptance of terms of service, and wallet creation or import. The system first activates the extension using OS-level GUI automation to click the browser's extension icon. If this interaction does not automatically open a full-page interface, the system directly navigates to a known extension-owned web page using its internal `chrome-extension://<id>/` URL specified in the manifest file. If the manifest does not declare a `default_popup` and the extension does not expose a stable, directly loadable UI page, we fall back to manual interaction. We observe that this situation occurs most often for older extension versions.

More complex actions, including sending transactions or performing cryptocurrency swaps, vary substantially across extensions and even across versions of the same extension. Automating these workflows is impractical due to nondeterministic execution behavior for each version. We therefore perform complex interactions manually, while relying on the automation agent to execute stable and repeatable onboarding steps. Outbound requests generated by browser extensions are intercepted using mitmproxy configured as a local forward proxy. Each Chromium instance is launched with its network traffic explicitly routed through mitmproxy via

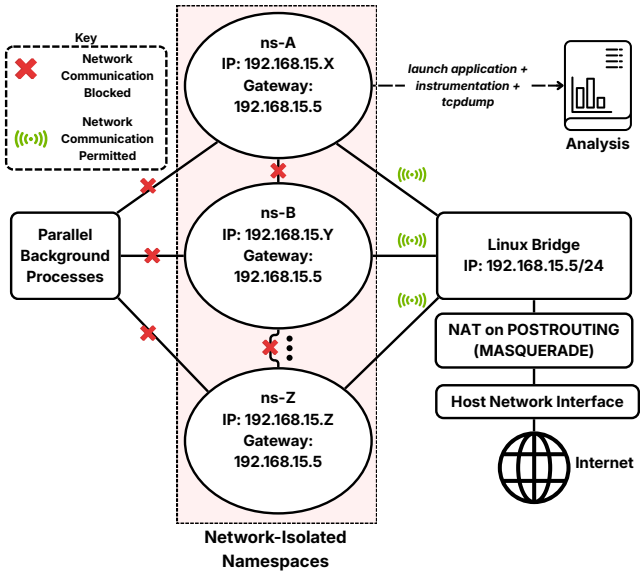

Fig. 2: *Linux network namespace setup for isolating desktop applications and capturing their outbound DNS traffic.*

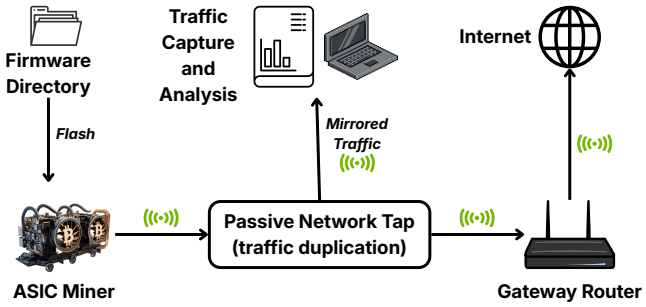

Fig. 3: *Experimental setup for analyzing cryptocurrency mining firmware. The mining device is connected to a personal router for Internet access, with a passive network tap inserted inline to mirror all outbound traffic to a monitoring host for DNS traffic capture.*

browser proxy settings. To enable HTTPS traffic interception, the browser profile is configured to trust the mitmproxy CA.

B. Desktop and Hardware Wallet Applications

Similar to browser extensions, we execute each desktop application in a network-isolated environment, systematically trigger its features, and capture the resulting traffic to extract DNS fingerprints. However, unlike browser extensions, we cannot rely on *mitmproxy* to intercept communications. Desktop applications establish connections at the operating system level and may use non-HTTP protocols, bypassing user-space proxies such as *mitmproxy*. Moreover, desktop applications may use certificate pinning or custom TLS stacks that prevent interception of HTTPS traffic using a proxy. To address this, we implement isolation and traffic capture directly at the network layer using Linux network namespaces and a custom network bridge configuration.

A Linux network namespace provides an isolated networking stack with its own interfaces, IP addresses, and routing tables, allowing applications to run in a logically separate network environment from the host. To connect these namespaces to the Internet, we attach them to a Linux bridge, which acts like a virtual Ethernet switch on the host. Each namespace is connected to the bridge via a virtual Ethernet pair, enabling traffic to flow between the namespace and the host's physical network interface. The bridge forwards packets and, together with iptables-based NAT rules, allows isolated applications to communicate externally while keeping their traffic separate from the host. This approach mirrors the mechanism used by containerization platforms such as Docker, which also rely on network namespaces and bridges to provide each container with an independent but routable network environment [21].

Our setup is illustrated in Figure 2. This design gives us fine-grained control over traffic isolation and capture. By placing each desktop application in its own namespace, we ensure that its network activity does not interfere with other applications or background processes on the host and that the captured DNS traffic solely represents the traffic generated by the underlying application. Within each namespace, we follow the same general execution strategy. Traffic is captured inside each namespace using `tcpdump`.

Hardware wallets are physical devices that store cryptocurrency private keys offline, providing enhanced security against online threats. To interact with these devices, users typically install companion desktop applications that facilitate wallet management and transaction signing. We focus on the desktop applications that interface with popular hardware wallets. These applications are installed and executed within isolated Linux network namespaces, following the same methodology as desktop wallets.

C. Mobile Applications

We analyze mobile wallet applications by running them within Android emulators, which provide a controlled environment for executing mobile apps on a desktop host. The emulators are themselves launched and exercised within Linux network namespaces to ensure network isolation, similar to the setup used for desktop wallets, using `tcpdump` to capture DNS traffic. Mobile applications under test are controlled on the emulated Android environment using `adb` commands, which enable programmatic interaction with the emulated device, including app installation and deletion.

D. Cryptocurrency Mining Firmware

Cryptocurrency mining can be performed using CPUs, GPUs, or application-specific integrated circuits (ASICs), with ASIC-based mining firmware offering the highest efficiency in practice. To study mining-related DNS behavior, we deploy mining firmware images on an Antminer S9 device, a widely used ASIC-based miner. The Antminer S9 natively runs a Linux-based mining firmware (denoted as *Antminer* in Table I), and multiple alternative firmware distributions are available that provide different performance optimizations and monitoring features. We select a set of popular mining firmware based on community recommendations and online

reviews. Each firmware image is obtained from its official source and flashed onto the Antminer S9 prior to experimentation. In addition to ASIC-based firmware, we include an open-source CPU-based mining application (*cpuminer*), which we execute within our aforementioned isolated namespaces.

Our experimental setup for mining firmware is illustrated in Figure 3. To capture network traffic from the ASIC device, we connect the Antminer to a dedicated router that provides Internet connectivity and insert a passive network tap inline between the miner and the router. The tap mirrors all packets to a separate monitoring host, allowing us to observe the miner's outbound network activity independently of the firmware in use. The miner is configured to connect to a mining pool and then started, while the monitoring host records all mirrored DNS traffic.

V. Analysis

A. Fingerprintability of Cryptocurrency Client Software

Each of the applications running on a user device emits its own DNS request patterns that are visible to the network observer described in Section II. We ask whether such an observer can infer that a user is running a particular application or mining firmware, even in the presence of DNS traffic from other concurrently running applications.

We categorize observed domains into three classes based on how strongly they reveal application identity, starting with the most distinctive signals. The *strong* category consists of application-specific domains that are tightly coupled to a particular application and typically support proprietary backend APIs and telemetry services. These domains provide high-confidence signals of application identity. Identity may be explicitly encoded in the domain name itself by embedding the application name or a known alias in the domain label. For example, `api.eternl.io` for the Eternl wallet or `doge-rpc.trustwalletapp.com` for the Trust Wallet Android application. Second, identity may be conveyed through application-specific telemetry and backend endpoints that are accessed exclusively by a single application and persist across versions. As an example, the domain `o1300201.ingest.sentry.io` is consistently resolved by every version of the Core Wallet application. Here, `o1300201` is a unique identifier [22] assigned by `sentry.io`, a third-party telemetry provider, and remains stable across Core Wallet releases. The presence of such domains enables a resolver-level observer to attribute DNS traffic to a specific cryptocurrency application.

The *medium* category consists of cryptocurrency-service domains that are specific to the cryptocurrency ecosystem but shared across multiple applications. These domains support common infrastructure, including blockchain RPC providers, wallet coordination services, block explorers, and exchange APIs that are used by many wallets to implement core functionality. For example, `mainnet.infura.io` is widely used by different wallet applications to relay JSON-RPC requests to the Ethereum mainnet. While these domains indicate cryptocurrency-related activity, they do not uniquely identify the application generating the traffic. We identify such domains using a curated list of known cryptocurrency infrastructure providers and classify them as medium associations.

Category	Domains

Weak	`android.googleapis.com`
	`static-cdn.hotjar.com`
	`star-mini.c10r.facebook.com`
	`cdn-settings.segment.com`

Medium	`mainnet.infura.io`
	`pulse.walletconnect.org`
	`bsc-dataseed1.binance.org`
	`min-api.cryptocompare.com`

Strong	`portfolio.api.cx.metamask.io`
	`swap.api.cx.metamask.io`
	`gas.api.cx.metamask.io`
	`tx-sentinel-ethereum-mainnet.api.cx.metamask.io`

TABLE II: *Domain categories observed in the latest version (v7.44.0) of the MetaMask mobile application*

Finally, the *weak* category includes general-purpose domains that are widely used across the Internet and commonly queried during routine web activity, but are not specific to cryptocurrency activity. Examples include domains supporting authentication services, such as Google login, or web analytics. As these domains are shared across many otherwise unrelated applications, their presence in DNS emissions does not provide meaningful information about application identity.

To systematically identify such domains, we develop a crawler that uses the Tranco top-10K websites [11] as seed URLs. The crawler visits each website and records all domains resolved during page loading. Since many resolved domains appear as subdomains, we canonicalize each hostname to its effective second-level domain using the Public Suffix List [14], aggregating domains that belong to the same underlying service provider. For instance, `ogads-pa.googleapis.com` and `update.googleapis.com` are both mapped to `googleapis.com`. Application-resolved domains whose effective second-level domains intersect with this baseline are classified as weak associations.

Some domains remained unclassified after applying these criteria: either they corresponded to cryptocurrency service providers too obscure to appear in our curated *medium* list, or they were too unpopular to be classified as *weak*. For this residual set, we queried OpenAI's GPT-5 model to assess whether each domain was associated with a cryptocurrency service. Domains identified as cryptocurrency-related were incorporated into the curated list and assigned a *medium* association, while the remainder were classified as *weak*. We validated the LLM-assisted classifications by confirming, through independent online documentation and service descriptions, the role attributed to *each* domain.

From the perspective of a network observer, domains in the *strong* category provide the highest confidence for attributing traffic to a specific application, whereas domains in the *medium* and *weak* categories offer progressively less certainty. Across the entire dataset, 27 (68%) applications exhibit the highest number of *strong* domains in their latest versions, and 35 (88%) applications have at least one *strong* domain in their latest version and are therefore uniquely identifiable at the DNS level under our threat model.

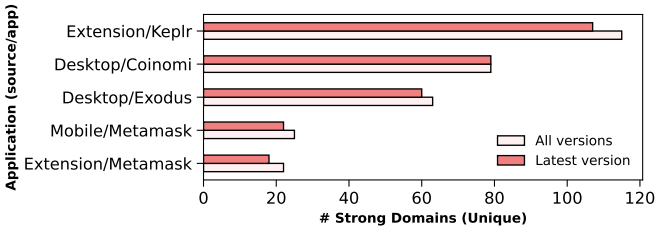

Fig. 4: *Top 5 applications by number of strong domains comparing the number of unique strong domains in all versions vs. latest version.*

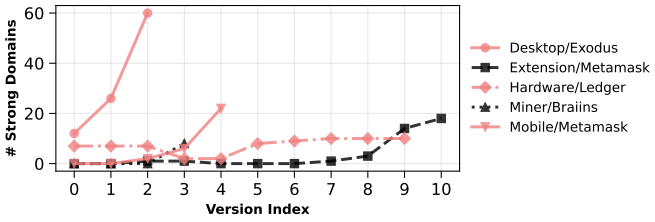

Fig. 5: *Trend of number of strong domains across versions for selected applications, one from each category.*

We observe a total of 528 unique *strong* domains and 644 unique *medium* domains, with a median of 5 *strong* domains per application. Furthermore, *strong* domains are exclusive to their originating applications and do not appear in the network footprints of other applications or in the crawler-collected *weak* domain set. Figure 4 presents the number of *strong* domains for the five applications with the largest counts in their latest versions, and additionally reports the total number of *strong* domains observed across all versions of each application. Figure 5 traces the evolution of *strong* domain counts across versions for a representative application from each software class. The observed upward trends suggest that domain-based fingerprintability persists across releases, and in some cases intensifies as applications mature and expand their network dependencies. Taken together, these results demonstrate that the large majority of cryptocurrency applications can be reliably identified based solely on their DNS emissions.

To further quantify attribution potential, we define two notions of fingerprintability for a given application version. Under the *Family* definition, the DNS fingerprint of an application version is non-empty, containing at least one domain not shared with any other application, although it may appear in other versions of the same application. Under the stricter *Global* definition, the fingerprint contains at least one domain that appears in no other application or version. Intuitively, *Family* fingerprintability allows a network observer to infer the presence of a particular application but not its exact version, whereas *Global* fingerprintability enables inference of both the application and its specific version.

Table III summarizes the number of applications that satisfy these two definitions, both when considering the latest version of each application and when considering all past versions in our dataset. We find that a substantial fraction of applications are fingerprintable based solely on their DNS fingerprints. Applications that do not exhibit a unique fingerprint lack *strong* domains in their DNS footprints. Our results suggest that for 34 (85%) applications, a network observer can determine that a user is running the latest application version.

Criterion	Past Version(s)		Latest Version	
	# Unique	# Not Unique	# Unique	# Not Unique
Family	32 (80%)	8 (20%)	34 (85%)	6 (15%)
Global	12 (30%)	28 (70%)	18 (45%)	22 (55%)

TABLE III: *Number of fingerprintable and non-fingerprintable applications under different uniqueness definitions.*

B. User Action Identification

We investigate whether DNS activity reveals information about user interactions with an application. As shown in Table II, domains in the *strong* category not only provide high-confidence evidence that the traffic originates from MetaMask, but also reveal the specific functionality being exercised, such as staking, portfolio management, or transaction-related activity. This behavior-level leakage stems from system design choices made by application developers, who commonly adopt microservice-based designs to support modular functionality and load balancing. As a result, different user interactions trigger requests to distinct backend services, which manifest as separate DNS requests. Consequently, DNS traces can distinguish between different modes of user engagement.

To understand what user actions can be inferred from DNS activity, we interacted with all versions of all applications in our dataset and recorded the DNS requests generated during each interaction type, following the methodology described in Section IV. We group user interactions into three categories. *Idle* corresponds to cases where the application or mining firmware is launched but not actively used, with no user interaction or cryptomining activity beyond startup. This captures scenarios in which a user has installed and opened the application or firmware but has not yet engaged with it. *Active* includes interactions such as account setup, wallet import, portfolio browsing, checking cryptocurrency prices, and cryptomining. *Transaction* refers to cases where the user initiates a cryptocurrency transaction, such as sending and receiving funds, or buying and selling assets. This category applies only to applications capable of performing on-chain operations and represents sensitive financial activity.

An application is considered to belong to a given interaction category if it resolves at least one domain that appears exclusively during that interaction. For example, if a domain is resolved during a transaction but does not appear during application startup or other interactions, it is assigned to the *Transaction* category. At the same time, if there is a domain that only resolves during idle usage, the application is also classified under the *Idle* category.

Table IV summarizes behavior-level fingerprintability per interaction category, broken down by historical and latest versions, with the average number of identifiable domains per

Interaction Group	Past Apps	Past Versions	Latest Versions	Past Avg. Domains	Latest Avg. Domains
Idle	22 (55%)	148 (56%)	17 (43%)	8.91	20.18
Active	20 (50%)	139 (53%)	19 (48%)	5.09	8.95
Transaction	9 (23%)	21 (8%)	9 (23%)	3.19	1.78

TABLE IV: *Behavior fingerprintability across interaction groups.*

ASN	Organization	# IP (%)
13335	Cloudflare	99 (47.6%)
14618	Amazon	51 (24.5%)
16509	Amazon	21 (10.1%)
396982	Google	11 (5.3%)
12876	Online SAS	6 (2.9%)

TABLE V: *Distribution of Web3-related domains across ASNs.*

version. We find that the latest versions of 9 (23%) popular wallet applications resolve domains that enable a network observer to infer when a user is performing a cryptocurrency transaction using the wallet. Specifically, this behavior is observed in the browser extension versions of Crypto.com, Keplr Wallet, Leap Wallet, MetaMask, Phantom Wallet, Solflare Wallet, Trust Wallet, and Typhon Wallet, as well as in the MetaMask mobile application. Across all of these applications, transaction-related DNS activity arises from interactions with backend APIs used to fetch gas fee estimates or to connect to blockchain nodes, making transaction execution distinguishable at the DNS level.

In the case of the Phantom wallet browser extension, we observe that during idle usage, the extension queries domains related to proxy services and error reporting. As the user becomes active and navigates through the application, additional APIs and data services are contacted. Finally, when the user initiates a transaction, the extension reaches out to `gas-price-oracle.phantom.app` to fetch current gas prices. If the user is sending assets to an Unstoppable Domain [2], the extension also queries `api.unstoppabledomains.com` to resolve the human-readable name to a cryptocurrency address. From this DNS query pattern, the network observer can infer not only that the user is performing a transaction but also that they are using an Unstoppable Domain for their transaction.

Similar patterns are observed in other applications, such as in the MetaMask mobile app, where the `tx-sentinel-[...].api.cx.metamask.io` domain is queried only when a transaction is initiated. In the case of mining firmware like Braiins OS, multiple *strong* domains are contacted to connect to BraiinsOS services on startup. Similarly, the HiveOS mining firmware connects to a different set of domains on startup (e.g., `api.hiveos.farm`) which are unique to HiveOS.

VI. Real-World Observability

In this section, we analyze the infrastructure and real-world usage of domains associated with the cryptocurrency applications under study using two vantage points.

1) Passive DNS: Understanding where cryptocurrency-related server endpoints are hosted is important for evaluating the practical decentralization guarantees of Web3 technologies.

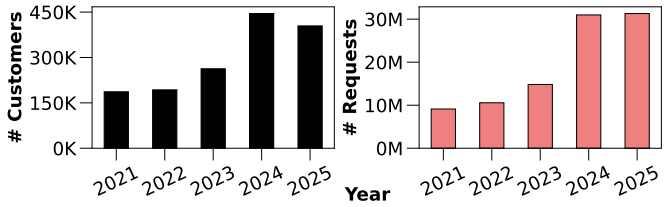

Fig. 6: *Aggregate number of customers and requests associated with domains from DNS fingerprints over time.*

While blockchain protocols themselves are designed to be decentralized, the supporting services, such as the domains discussed in Section V, often rely on centralized Internet providers, introducing potential chokepoints for censorship. We therefore examine the hosting infrastructure underlying these domains to assess the extent to which the backend services supporting cryptocurrency client software are distributed across Internet infrastructure providers.

To this end, we use DomainTools' passive DNS (pDNS) dataset [8] to collect the most recent IPv4 and IPv6 addresses for 197 representative *strong* and *medium* domains that are stable across application versions. This dataset is constructed from telemetry aggregated from recursive resolvers across global vantage points. We then map each observed IP address to its origin AS using a public IP-to-ASN service [7], which infers AS attribution from BGP routing announcements. This mapping captures the network operators that route traffic to the backend infrastructure at the time of observation.

We find that a majority of the infrastructure servicing users of cryptocurrency client software is operated by a concentrated set of network providers, placing these providers in a position to potentially censor these services and disrupt millions of users' ability to access the blockchain and their digital assets. Table V shows the distribution of the domains across the top 5 ASes, which together account for 90% of all observed IP addresses.

In addition, passive DNS records reveal sustained and widespread resolution of these domains, as evidenced by large numbers of RRset observations accumulated over extended time periods. Across the analyzed domains, recursive resolvers recorded an average of 11.5M RRset observations and a median of 1.4M observations, with `api.coinbase.com` having over 200M observed RRsets between 2014 and 2025.

2) Enterprise URL Filtering logs: While passive DNS provides a global view of the persistence and deployment of Web3-supporting domains, it does not capture how these domains appear within live operational networks or reflect actual user activity. To understand how Web3-supporting domains manifest in real usage environments, we therefore analyze aggregated DNS resolution activity derived from enterprise URL filtering logs collected by Palo Alto Networks. This data is collected through routine security monitoring across enterprise customer networks of varying scale. It allows us to quantify the practical observability of Web3 DNS fingerprints and characterize their prevalence across industry sectors.

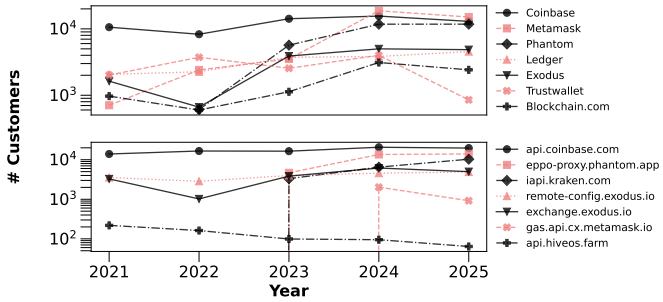

Fig. 7: *The top panel reports the number of customers resolving two or more domains belonging to the same application fingerprint within a 30-minute window, shown over the 2021 to 2025 period. The bottom panel shows the number of customers resolving the five most frequently observed hostnames in the dataset over time.*

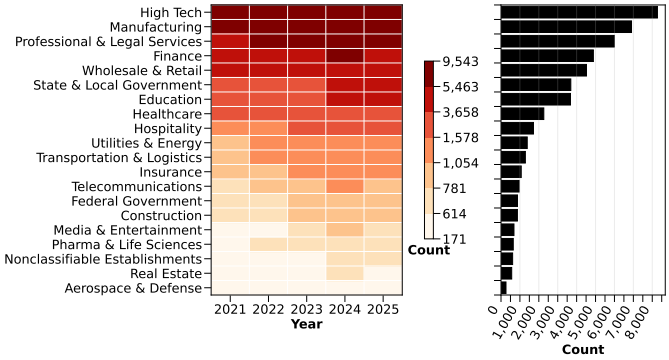

Fig. 8: *Number of customers resolving domains from application fingerprints spanning multiple industry categories over time.*

To perform this measurement, we analyze resolution activity associated with the same set of domains used in our passive DNS analysis that comprise the DNS fingerprints of the applications under study. These fingerprint-related resolutions appear alongside billions of other domains routinely processed within customer networks during normal security operations. To place these signals in a longitudinal context, we isolate fingerprint-related traffic collected during each October between 2021 and 2025. For each snapshot, we measure both the total number of DNS requests generated towards fingerprint domains and the number of distinct customer networks issuing those requests, where a single customer network may encompass multiple end users.

Overall, our results indicate substantial activity associated with our curated DNS fingerprints. We observe a general upward trend in both metrics over time, with DNS request volume increasing from 9.1M in October 2021 to 31.2M in October 2025. Figure 6 shows the total number of unique customer networks and the total volume of DNS requests generated by those networks for domains associated with selected representative fingerprints. The slower growth observed from 2024 to 2025 is likely due to coverage limitations, as fingerprints collected in mid-2025 may not capture domains introduced in the most recent application versions.

To further distinguish genuine application usage, we identify instances in which a single device within a customer network resolves two or more *strong* domains associated with the same application fingerprint within a 30-minute window. Using this criterion, we were able to identify user activity across 27 applications over time, with the number of users increasing from 27K in 2021 to 68K in 2025. The top panel of Figure 7 shows the number of user devices through which we detect such activity for 7 popular applications. In the bottom panel, we show the activity towards individual domains, which reveals high activity for *strong* domains in application fingerprints. For example, in October 2024, activity across 2,000 customers was observed for `gas.api.cx.metamask.io` totaling 74.5K requests. This domain is resolved by the MetaMask extension and mobile application to retrieve gas fee estimates for Ethereum transactions when the user is preparing a transaction.

Enterprise networks represent a particularly consequential vantage point for observing cryptocurrency-related activity, as they often operate under strict compliance requirements, and routinely monitor DNS traffic for security purposes. Understanding whether cryptocurrency client software appears in such environments is therefore critical for assessing whether the privacy assumptions users associate with Web3 systems hold in practice. At the same time, the presence of these signals may also reflect opportunities for legitimate defensive practices, such as the use of DNS-based monitoring to detect unauthorized cryptocurrency activity, such as cryptocurrency mining on campus networks.

Figure 8 supports this analysis by showing the number of domain resolutions per enterprise customer category associated with cryptocurrency application fingerprints. We observe that these DNS signals span a wide range of sectors, including regulated and security-sensitive industries such as *Healthcare*, *Federal Government*, and *State and Local Government*. This demonstrates that cryptocurrency client software is present not only in consumer or technology-centric settings, but also within organizational environments where network traffic is commonly monitored and retained. In such contexts, the visibility of DNS traffic increases the likelihood that cryptocurrency application usage and associated behavioral patterns can be inferred by network operators, even when users enable third-party DNS encryption mechanisms (Section II).

Together, these results highlight that the DNS dependencies of Web3 applications create practical points of observability and control, with implications for large-scale monitoring and censorship that users may not anticipate.

VII. Related Work

Prior work has shown that user activity on the web can be inferred through tracking mechanisms embedded in websites and third-party services, enabling the collection and correlation of sensitive information across browsing sessions [26]. Torres *et al.* [24] and Winter *et al.* [27] demonstrate that decentralized finance (DeFi) applications inherit these risks when cryptocurrency functionality is exposed through browser-based

front ends. They show how wallet extensions and DeFi interfaces integrate with the existing web tracking ecosystem and may leak sensitive information such as wallet addresses to third parties. These works assume a threat model in which privacy loss arises from interactions with websites or embedded third-party scripts. In contrast, our work shows that cryptocurrency-related activity can be inferred without any web-based tracking. We demonstrate that DNS resolution activity generated by cryptocurrency client software itself reveals application identity and behavior due to its reliance on application-specific Internet infrastructure.

DNS has been widely used as a signal for security monitoring [3], [9], [10], [13], [15], [16]. For example, prior work has shown that DNS can reveal connections to command-and-control infrastructure via DGAs [4], [20] and can be used to fingerprint IoT devices based on their DNS behavior [19]. To the best of our knowledge, this is the first work to quantify how DNS visibility alone enables direct, deterministic attribution of cryptocurrency application identity and version.

Related research has also shown that application usage can be inferred from encrypted network traffic by analyzing packet-level metadata such as sizes and timing, with applications to mobile app identification [25], VPN detection [28], and cryptomining activity [23]. These approaches rely on probabilistic classification over traffic features. In contrast, our work shows that DNS visibility enables direct attribution of cryptocurrency application identity and version based on DNS dependencies.

VIII. Conclusion

Privacy and censorship resistance in cryptocurrency applications are only as strong as the client software through which users access them. By analyzing the DNS behavior of 40 cryptocurrency applications across 303 versions, we show that a passive observer at the DNS resolver level can identify which application a user runs, determine its version, and detect when a financial transaction is initiated. The infrastructure supporting these applications is highly concentrated, with 90% of representative IP addresses routed through just five Autonomous Systems, creating practical chokepoints for censorship. Validation against enterprise DNS telemetry confirms these signals are observable across thousands of real-world networks, including regulated industries such as healthcare and government. We hope this work encourages Web3 developers to treat DNS-level privacy as a design concern.

Acknowledgment

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. We thank Alexandros Kapravelos and Kostas Solomos for sharing historical browser extension packages that are not publicly available.

References

[1] New trains, same rails: Fingerprinting cryptocurrency software via network requests, https://pragseclab.github.io/DNS-Web3/
[2] Unstoppable domains, https://unstoppabledomains.com/
[3] Ahmed, J.: Monitoring security of enterprise hosts via DNS data analysis. Ph.D. thesis (2022)
[4] Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., Dagon, D.: From {Throw-Away} traffic to bots: Detecting the rise of {DGA-Based} malware. In: USENIX Security (2012)
[5] AWS: What is web3? (2025), https://aws.amazon.com/what-is/web3/
[6] Cybersecurity, Agency, I.S.: Encrypted dns implementation guidance (2024), https://www.cisa.gov/sites/default/files/2024-05/Encrypted%20DNS%20Implementation%20Guidance_508c.pdf
[7] Cymru, T.: Ip to asn mapping service (2025), https://www.team-cymru.com/ip-asn-mapping
[8] DomainTools: Domaintools technical documentation (2025), https://docs.domaintools.com/dnsdb/
[9] Faulkenberry, A., Avgetidis, A., Ma, Z., Alrawi, O., Lever, C., Kintis, P., Monrose, F., Keromytis, A.D., Antonakakis, M.: View from above: Exploring the malware ecosystem from the upper dns hierarchy. In: ACSAC (2022)
[10] Kountouras, A., Kintis, P., Lever, C., Chen, Y., Nadji, Y., Dagon, D., Antonakakis, M., Joffe, R.: Enabling network security through active dns datasets. In: RAID. Springer (2016)
[11] Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczynski, M., Joosen, W.: Tranco: A research-oriented top sites ranking hardened against manipulation. In: NDSS (2019)
[12] Lee, J., Mohaisen, D., Kang, M.S.: Measuring dns-over-https downgrades: Prevalence, techniques, and bypass strategies. Proceedings of the ACM on Networking (2024)
[13] Lyu, M., Habibi Gharakheili, H., Russell, C., Sivaraman, V.: Mapping an enterprise network by analyzing dns traffic. In: PAM. Springer (2019)
[14] Mozilla: Public suffix list (2025), https://publicsuffix.org/
[15] Muzammil, M., Ansari, Z., Nikiforakis, N., Johnson, D.: Echoes of the Past: Detecting and Classifying Re-registered Domains in Enterprise DNS Traffic. In: MADWeb (2026)
[16] Muzammil, M., Pitumpe, A., Li, X., Rahmati, A., Nikiforakis, N.: The poorest man in babylon: A longitudinal study of cryptocurrency investment scams. In: The ACM WebConf (2025)
[17] Müller, M., Žunič, G.: Browser use: Enable ai to control your browser, https://github.com/browser-use/browser-use
[18] (NSA), N.S.A.: Adopting encrypted dns in enterprise environments (2021), https://media.defense.gov/2021/Jan/14/2002564889/-1/-1/0/CSI_ADOPTING_ENCRYPTED_DNS_U_OO_102904_21.PDF
[19] Perdisci, R., Papastergiou, T., Alrawi, O., Antonakakis, M.: Iotfinder: Efficient large-scale identification of iot devices via passive dns traffic analysis. In: EuroS&P. IEEE (2020)
[20] Plohmann, D., Yakdan, K., Klatt, M., Bader, J., Gerhards-Padilla, E.: A comprehensive measurement study of domain generating malware. In: USENIX Security Symposium (2016)
[21] Rana, S.: Docker: What's under the hood? (2025), https://blog.kubesimplify.com/docker-networking-demystified
[22] Sentry: Organization subdomains in dsns (2025), https://forum.sentry.io/t/organization-subdomains-in-dsns/9360
[23] Sun, H., Shi, R., Lan, L., Peng, Z., Wang, C.: A two-stage encrypted cryptomining traffic detection mechanism in campus network. In: ICBC. IEEE (2024)
[24] Torres, C.F., Willi, F., Shinde, S.: Is your wallet snitching on you? an analysis on the privacy implications of web3. In: USENIX Security Symposium (2023)
[25] Van Ede, T., Bortolameotti, R., Continella, A., Ren, J., Dubois, D.J., Lindorfer, M., Choffnes, D., Van Steen, M., Peter, A.: Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic. In: NDSS (2020)
[26] Vekaria, Y., Beugin, Y., Munir, S., Acar, G., Bielova, N., Englehardt, S., Iqbal, U., Kapravelos, A., Laperdrix, P., Nikiforakis, N., et al.: Sok: Advances and open problems in web tracking. arXiv (2025)
[27] Winter, P., Lorimer, A.H., Snyder, P., Livshits, B.: Security, privacy, and decentralization in web3. arXiv (2021)
[28] Xue, D., Ramesh, R., Jain, A., Kallitsis, M., Halderman, J.A., Crandall, J.R., Ensafi, R.: Openvpn is open to vpn fingerprinting. Communications of the ACM (2025)