Exploring the Ecosystem of DNS HTTPS Resource Records: An End-to-End Perspective

Hongying Dong*
University of Virginia
Charlottesville, Virginia, USA
hd7gr@virginia.edu

Yizhe Zhang*
University of Virginia
Charlottesville, Virginia, USA
yz6me@virginia.edu

Hyeonmin Lee
University of Virginia
Charlottesville, Virginia, USA
frv9vh@virginia.edu

Shumon Huque
Salesforce
McLean, Virginia, USA
shuque@gmail.com

Yixin Sun
University of Virginia
Charlottesville, Virginia, USA
ys3kz@virginia.edu

Abstract

The DNS HTTPS resource record is a new DNS record type designed for the delivery of configuration information and parameters required to initiate connections to HTTPS network services. In addition, it is a key enabler for TLS Encrypted ClientHello (ECH) by providing the cryptographic keying material needed to encrypt the initial exchange. To understand the adoption of this new DNS HTTPS record, we perform a longitudinal study on the server-side deployment of DNS HTTPS for Tranco top million domains, as well as an analysis of the client-side support for DNS HTTPS through snapshots from major browsers. To the best of our knowledge, our work is the first longitudinal study on DNS HTTPS server deployment, and the first known study on client-side support for DNS HTTPS. Despite the rapidly growing trend of DNS HTTPS adoption, our study highlights challenges and concerns in the deployment by both servers and clients, such as the complexity in properly maintaining HTTPS records and connection failure in browsers when the HTTPS record is not properly configured.

CCS Concepts

• **Networks** → **Naming and addressing**; **Network measurement**.

Keywords

Measurement, DNS, HTTPS resource record, HTTPS RR, Encrypted ClientHello, ECH

ACM Reference Format:
Hongying Dong*, Yizhe Zhang*, Hyeonmin Lee, Shumon Huque, and Yixin Sun. 2024. Exploring the Ecosystem of DNS HTTPS Resource Records: An End-to-End Perspective. In *Proceedings of the 2024 ACM Internet Measurement Conference (IMC '24), November 4–6, 2024, Madrid, Spain.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3646547.3688410

* Both Hongying Dong and Yizhe Zhang contributed equally to this work.

IMC '24, November 4–6, 2024, Madrid, Spain

ACM ISBN 979-8-4007-0592-2/24/11
https://doi.org/10.1145/3646547.3688410

1 Introduction

Transport Layer Security (TLS) plays a pivotal role in securing the Internet. Notably, Hypertext Transfer Protocol Secure (HTTPS) is an extension of Hypertext Transfer Protocol (HTTP) that employs TLS to protect web communications (e.g., communications between a web browser and a website).

However, upgrading a connection from HTTP to HTTPS typically incurs additional latency. Since a browser initially does not have knowledge of whether a website supports HTTPS, it typically attempts to first send a plaintext HTTP request.[1] The connection is only upgraded to HTTPS if the website responds with an HTTPS redirect. The plaintext nature of the HTTP to HTTPS redirection process presents a potential target for man-in-the-middle attackers to block or redirect clients to their own (malicious) HTTPS site. Although the HTTP Strict Transport Security (HSTS) [26] policy and Alt-Svc header [34] can mitigate these issues to some degree, they do not negate the need for the browser's first HTTP request and the HTTPS redirect. Additionally, while pre-configured HSTS preload lists can be used by browsers to unconditionally connect to websites on the list using HTTPS, the process of populating such lists is manual and does not cover the vast majority of websites.

The recently standardized SVCB and HTTPS DNS Resource Records (RR) [45] offer promising approaches to address these concerns. SVCB records provide clients with comprehensive information needed to access a service, including supported protocols, port numbers, and IP addresses by directly storing the information in the DNS record. In particular, the HTTPS record, a variation of SVCB tailored to the HTTPS protocol, informs clients about a website's HTTPS support, along with additional details such as supported HTTP versions. Therefore, a client can obtain all necessary information for accessing a website through a single HTTPS DNS query, thereby enabling it to directly establish a TLS session using this information. In contrast to the CNAME record, which aliases the entire domain and thereby excludes the inclusion of other record types, the HTTPS record supports coexistence with various record types and offers enhanced connection capabilities. Unlike the DNAME record that redirects an entire DNS subtree to another subtree, HTTPS records can redirect web traffic specifically at the DNAME owner while permitting distinct redirection policies for subdomains. Additionally, HTTPS

[1] This occurs if a user enters a domain name (e.g., a.com) in the browser's address bar without adding HTTPS prefix (e.g., https://a.com).

records may be employed within the DNAME subtree itself. Furthermore, another important aspect that HTTPS records could facilitate is the conveyance of TLS Encrypted Client Hello (ECH) [39] information, which encrypts the *ClientHello* message during the TLS handshake. Although ECH is not yet standardized, its integration with HTTPS records could enhance the privacy of TLS connections.

Given the performance and security benefits provided by the HTTPS record, it has been adopted by large cloud providers such as Cloudflare [22] and Akamai [2], even before its standardization in November 2023. Furthermore, popular web browsers (e.g., Chrome [11], Firefox [7], and Safari [36]) and DNS software (e.g., BIND 9 [6], PowerDNS [37], and Knot DNS [28]) support HTTPS records.

Nevertheless, the understanding of the current landscape of HTTPS record is still limited. While a preliminary study [51] provides brief statistics on server-side HTTPS records through one snapshot, it does not perform a longitudinal analysis or investigate client-side support. This research gap limits the understanding of the effectiveness and challenges of both server-side and client-side HTTPS record deployment.

In this paper, we present an *extensive* study of the DNS HTTPS ecosystem, by encompassing both server-side and client-side deployments. For server-side analysis, we take daily snapshots of HTTPS records from May 2023 to Mar 2024 for domains in the Tranco list [29], which ranks the top 1 million domains based on their popularity across various lists. Specifically, we scan the primary apex domains (e.g., a.com) and their corresponding www subdomains (e.g., www.a.com). We also collect other relevant data such as the NS and SOA records (from Aug 2023 to Mar 2024) and WHOIS information of name servers (from Oct 2023 to Mar 2024) to facilitate our analysis.

To examine client-side behavior, we investigate how popular web browsers support HTTPS records, including their failover mechanisms, by configuring our own DNS server with HTTPS records and performing active experiments with the browsers. To the best of our knowledge, there is no existing research on browsers' support of HTTPS records.

Key contributions. We present the first longitudinal study on the deployment of DNS HTTPS records by top domains and perform extensive testing of browser behavior in handling DNS HTTPS requests. Our measurements allow us to gain a deeper understanding of the DNS HTTPS ecosystem, and highlight potential obstacles and concerns that can help inform future deployment. Our key findings are:

- Despite its recent standardization, over 20% of Tranco top 1M domains have DNS HTTPS records and major browsers utilize HTTPS records in establishing connections. Notably, a major contributing factor is *Cloudflare*'s default HTTPS configuration, which accounts for over 70% of domains with HTTPS records.
- The adoption of HTTPS records inevitably incurs complexity and overhead in properly maintaining the validity of records to avoid connection failure. For example, the frequent ECH key rotation every 1 to 2 hours requires the server to properly implement retry mechanisms [39] to avoid breaking the connection due to expired keys.
- Connection failures from major browsers occur in various HTTPS misconfigurations due to the lack of proper failover mechanisms, exacerbating the challenge of utilizing HTTPS records for both servers and clients. Furthermore, the lack of support for ECH Split Mode leads to such failures across all major browsers, prohibiting clients from establishing connections to the server even when ECH is correctly configured by the server.

Artifact availability. We provide full availability, including our dataset and code to reproduce our results at https://github.com/yzzhn/imc2024dnshttps. In the long run, we plan to maintain a longstanding framework that continuously collects and releases HTTPS data periodically.

2 Background

We briefly discuss background on DNS HTTPS records and TLS Encrypted Client Hello (ECH) extension. Further details on DNS record types can be found in Appendix B.

DNS HTTPS records. The HTTPS (and SVCB) record [45] is designed to offer alternative endpoints for a service, along with parameters associated with each endpoint, within a single DNS record. The HTTPS record signals the use of HTTPS (rather than HTTP) for the specified host. Additionally, it can coexist with other record types, thus enabling name redirection at both zone apexes and any arbitrary location within a zone, a feature not supported by the CNAME record.

```
a.com. 300 IN HTTPS 0 b.com.
c.com. 300 IN HTTPS 1 . alpn=h3 ipv4hint=1.2.3.4
```

Figure 1: An example of HTTPS records.

Figure 1 illustrates example HTTPS records. Each HTTPS record consists of two required fields and one optional field:

- SvcPriority: the priority of the record (lower values preferred). A value of zero indicates AliasMode, aliasing a domain to the target domain (specified in TargetName). Other values indicate ServiceMode, which provides information specific to a service endpoint.
- TargetName: a domain name, which can be either the alias target in AliasMode or the alternative endpoint in ServiceMode. If ServiceMode specifies the value ".", the owner name of this record has to be used.
- SvcParams (optional): Utilized only in ServiceMode, a list of key-value pairs are included to provide details about the endpoint (in TargetName). The current specification defines seven parameter keys (port, alpn, no-default-alpn, ipv4hint, ipv6hint, ech, mandatory). The port parameter specifies additional ports supported by the endpoint, while alpn (Application-Layer Protocol Negotiation) specifies additional application protocols; by default, an HTTPS record indicates HTTP/1.1 support. The no-default-alpn key is used if the endpoint doesn't support the default protocol. ipv4hint/ipv6hint suggest IPv4/IPv6 addresses for reaching the endpoint. The ech parameter can be used to include the Encrypted Client Hello (ECH) information. The mandatory parameter specifies mandatory keys that must be used to connect to the endpoint.

One notable functionality of the HTTPS record is its ability to publish ECH information. In the current specification [45], ech is a reserved SvcParam, as the ECH specification has not yet been standardized.

Comparison with DNS CNAME and DNAME records: The DNS HTTPS record is expected to offer an improved replacement for the CNAME record commonly used today to redirect websites to a third party or alternate location. The CNAME record is more general purpose in nature since it completely aliases one domain name to another location. Since the CNAME aliases the entire domain name (including all the record types at that name), no other record types can exist at the origin domain name. This precludes its use as a web redirection mechanism at the apex of a DNS zone (since the apex necessarily includes other record types like NS and SOA). By contrast, the HTTPS record is application- and type-specific, which can coexist with record types, and offers additional connection level capability discovery and an extensible framework for new parameters. The SVCB record (the more generalized form of HTTPS records) is more similar to the existing DNS SRV record [23], where the record name identifies the service (and optionally other parameters like port number) via additional labels prepended to the domain name, but offers all the additional connection level capability indications that the SRV record does not.

The DNAME record [43] redirects an entire DNS subtree underneath the owner domain name to another DNS subtree, and is not directly comparable to the HTTPS record, although they can co-exist in various ways. For example, an HTTPS record could conceivably redirect web traffic at the owner of the DNAME, while subdomains of the DNAME are redirected elsewhere. HTTPS records could be placed at domains names in the redirected DNAME subtree.

TLS Encrypted Client Hello (ECH). ECH [39] is a TLS extension that allows a client to encrypt its initial *ClientHello* message in the TLS handshake. Normally, the *ClientHello* message is sent in plaintext, revealing information such as the server's domain name (SNI).[2] To enable ECH, a domain needs to publish key information (e.g., its public key for encrypting the *ClientHello* message). The HTTPS record provides a way for this publication, allowing a domain to include its key information as the ech parameter. Therefore, a client can retrieve the HTTPS record and use the ech parameter to encrypt its *ClientHello* message. Currently, major browsers like Chrome [10] and Firefox [32] have implemented ECH.

Tranco list. The Tranco List [29] offers a thorough and up-to-date ranking of the internet's most visited websites by aggregating data from a range of sources. This list synthesizes diverse traffic measurements to produce a unified ranking system that accurately reflects the relative popularity of websites. Its integration of multiple traffic metrics enhances both the precision and reliability of the rankings. In this study, we use the Tranco list as the basis for investigating trends in HTTPS deployment across popular domains.

3 Research Overview

We provide a brief overview of our research goals and agenda.

(1) Server-side HTTPS record deployment

- **Goal**: Our aim is to analyze the deployment trends of HTTPS records, as well as their characteristics.
- **Methodology**: We measure the deployment of HTTPS records, focusing on the popular domains within the Tranco top one million domain list. Our analysis includes details on the HTTPS record configurations, along with the support of other security protocols such as ECH and DNSSEC in conjunction with HTTPS records.
- **Datasets**: We collect HTTPS records as well as other DNS records, including A, AAAA, SOA, and NS records, every day from the Tranco 1 million domains. We also scan the IP addresses (A/AAAA records) of name servers used by domains that deploy HTTPS records. Additionally, we utilize the WHOIS database and perform DNSSEC record validation to further analyze the management of HTTPS records from these domains. The details of the server-side datasets are described in Section 4.1.

(2) Client-side HTTPS record support

- **Goal**: We aim to examine support of HTTPS records and identify associated behaviors in popular web browsers.
- **Methodology**: We focus on the top four browsers: Chrome, Safari, Edge, and Firefox. We analyze whether these browsers (i) perform HTTPS records lookup, (ii) utilize the information in the HTTPS records, and (iii) how they respond to incorrect/inaccurate HTTPS records. We set up our own DNS server and configure HTTPS records to perform controlled experiments. The setup and methodology are detailed in Section 5.

We provide insights into the adoption of DNS HTTPS records through quantitative server-side analysis and examine the impact of web browser functionalities on HTTPS record support via the client-side evaluation. These approaches together present a comprehensive view of the HTTPS record ecosystem and highlight the challenges in the current deployment.

4 Server-side HTTPS RR Deployment

We first describe our data collection. We then delve into the details of server-side HTTPS RR configurations.

4.1 Datasets

Scanning framework. Our scanning framework retrieves the top 1 million domain list from the Tranco [29] daily, and performs daily scans of HTTPS records along with other DNS records, as shown in Table 1. The Tranco list does not differentiate between apex domains (i.e., a.com) and www subdomains (i.e., www.a.com), and includes both types. We preprocess by retrieving the apex domains of all top million domains, and generate www subdomains by adding www prefix to the apex domains. We distinctly consider apex domains and www subdomains, where the latter have more specific usage for the web. This results in two lists: (1) a one-million list of apex domains, and (2) a corresponding one-million list of www subdomains.

Next, to scan DNS records of the domain lists, we implement a scanner by utilizing the *dnspython* [24] library. For each domain in the list, we initiate a DNS HTTPS query to two widely recognized public DNS resolvers: *Google* (8.8.8.8) as the main resolver and

[2] Although TLS 1.3 encrypts most handshake messages, the *ClientHello* message remains unencrypted.

Data Type		Measurement Period	Utilized in Section					
			4.2.1 Adoption rates	4.2.2 Name servers	4.2.3 Inconsistent use	4.3 HTTPS RR parameters	4.4 ECH deployment	4.5 HTTPS RR and DNSSEC
Domain (Apex, www)	HTTPS, A, AAAA	2023-05-08 – 2024-03-31	✓	✓	✓	✓	✓	✓
	SOA, NS	2023-08-16 – 2024-03-31		✓	✓	✓		
Name Server	A, AAAA, WHOIS	2023-10-11 – 2024-03-31		✓	✓			

Table 1: Overview of datasets for server-side analysis.

Cloudflare (1.1.1.1) as the backup resolver. If a domain produces a CNAME response, our inquiry extends to sending an HTTPS query to the domain in the resolved CNAME.[3] For HTTPS records, we also collect the corresponding RRSIG records if provided, along with the HTTPS record. Additionally, we gather information included in DNS responses of HTTPS records, such as the Authenticated Data (AD) [52] bit, which indicates that the response is authenticated with DNSSEC. If either the domain or the one in the CNAME has an HTTPS record, we proceed to perform additional queries for the domain, including A (IPv4 address), AAAA (IPv6 address), SOA, and NS records lookup.

Additionally, we perform daily scans of A/AAAA records for name servers used by domains that publish HTTPS records to analyze the distribution and trend. The name servers queried are obtained from the NS records collected through the daily apex and www domain scanning. Additionally, we perform WHOIS lookups for IP addresses found in the A/AAAA records of name servers, complemented by manual reviews (as described in section 4.2.2), to ascertain the registered owner of these IPs. This information is utilized to further analyze the operators of each name server. Our dataset on name servers covers the period from October 11th, 2023 to March 31st, 2024. Further discussion on ethics can be found in Appendix A.

Analyzing domains in the Tranco list. Starting from August 1st, 2023, Tranco changed its data sources by phasing out Alexa ranking for Chrome User Experience Report and *Cloudflare* Radar data [47], impacting the domain list composition. Therefore, our analysis is split into two phases, before and after August 1st, 2023, to account for the major change in the Tranco top 1M domain list.

Furthermore, since the Tranco list is updated *daily*, the composition of domains within the list may change each day. Consequently, relatively popular domains (e.g., those with higher rankings) are likely to be consistently included in the list, while domains with lower rankings may experience fluctuations in their presence on the list. Detailed observations of Tranco domain rankings can be found in Appendix C.

Therefore, our subsequent analysis distinctively considers two sets of domains, defined as follows:

(i) **Dynamic Tranco top 1M domains**: this set of domains includes the entire 1 million domains in the daily Tranco list. Analyzing this set allows us to observe general trends in HTTPS record deployment of top domains.

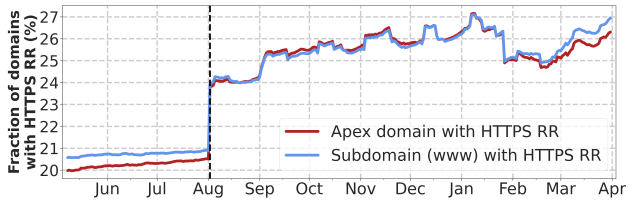
(ii) **Overlapping domains**: this set of domains consists of domains that appear in the Tranco list *every day* during our entire measurement period. Due to the changes in Tranco's data source, we further divide our analysis period into two parts. The first part, spanning from May 8th to July 31st, 2023, precedes Tranco's source change and includes 634,810 unique overlapping domains. The second part, from August 1st, 2023, to March 31st, 2024, follows the source change and includes 684,292 unique overlapping domains. By analyzing these overlapping domains, we aim to understand the behavior within a stable (and potentially more popular) set of domains.

[3]Note that CNAME records at the apex of a zone are technically not allowed [3], although some misconfigured DNS servers allow them to be installed.

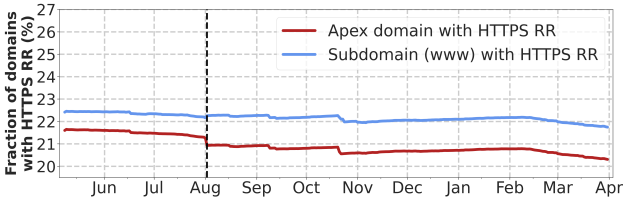
4.2 HTTPS RR Adoption

We analyze the adoption of HTTPS records from the perspective of domains and name servers.

4.2.1 Adoption rates. We start our analysis by examining the deployment of HTTPS records by domains in the Tranco list. Figure 2 shows the percentage of apex domains and their corresponding www subdomains that publish HTTPS records, from May 8th, 2023, to March 31st, 2024, ranging from 20% to 27%.



(a) Dynamic Tranco top 1M apex and www subdomains.



(b) Overlapping apex and www subdomains.

Figure 2: Percentages of apex/www domains that publish HTTPS records. Vertical dashed line (on August 1st, 2023) denotes the source change of the Tranco list.

We notice that the HTTPS adoption rates exhibit distinct trends between the dynamic Tranco 1M domains (Figure 2a) and the overlapping domains (Figure 2b). While an overall continuously increasing trend is observed in adopting HTTPS RR for the dynamic Tranco top domains, overlapping apex domains and corresponding www subdomains show a relatively stable ratio. This stability is disrupted only by a decrease in apex domains and a minor increase in www subdomains, which occur when Tranco updates its source feeds. This is followed by a gradual decline, likely attributable to changes in name servers (detailed in Section 4.2.2). Thus, the increase in HTTPS records deployment is more attributed to daily domains that are *not* always in the Tranco top 1M domain list (i.e., non-overlapping domains).

Name Server (NS) Category	Dynamic Tranco top 1M		Overlapping	
	Mean (%)	Std.	Mean (%)	Std.
Full *Cloudflare* NS	99.89	0.03	99.87	0.04
None *Cloudflare* NS	0.11	0.03	0.13	0.04
Partial *Cloudflare* NS	< 0.01	-	< 0.01	-

Table 2: The average and standard deviation of the percentage of apex domains (with HTTPS records) served by *Cloudflare* vs non-*Cloudflare* name servers.

4.2.2 Name servers. Considering that the name server providers could have a considerable impact on the adoption of HTTPS RR, we now take a closer look at the distribution of name servers utilized by apex domains that adopt the HTTPS RR. We utilize additional data, including A/AAAA records of name servers and WHOIS information for the corresponding IP addresses (detailed in Table 1) to pinpoint the host organization of each name server.[4] Additionally, we conducted a supplementary manual review of DNS providers. This involved a thorough analysis of their documentation to exclude instances where domain owners utilize cloud service providers for hosting while operating their own name servers (e.g., Amazon AWS). Such cases may result in AS information that misleadingly attributes the name servers to the cloud service provider. Note that a limitation with using WHOIS occurs when customers use their own IP addresses with cloud service providers (a practice known as BYOIP, or Bring Your Own IP). In such cases, the WHOIS information may reflect the original owner rather than the cloud service provider. This limitation likely affects the tail distribution of name server providers, but not top providers such as Google and GoDaddy.

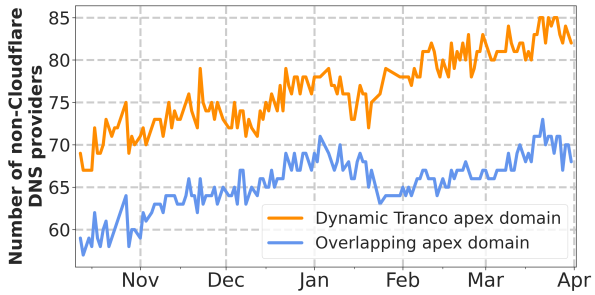


Figure 3: Number of non-*Cloudflare* DNS providers employed by domains that activate HTTPS records.

***Cloudflare* and non-*Cloudflare* name servers.** Interestingly, we notice that over 99% of both dynamic Tranco and overlapping apex domains that publish HTTPS records use *Cloudflare* name servers, as shown in Table 2. Note that we classify it as "Full *Cloudflare* name server" when an apex exclusively use *Cloudflare* name servers. On the other hand, we categorize it as "Partial *Cloudflare* name server" if an apex uses a combination of *Cloudflare* and non-*Cloudflare* name servers. The "None *Cloudflare* name server" indicates that an apex only utilizes name servers not provided by *Cloudflare*.

In total, there are 244 distinct non-*Cloudflare* DNS service providers used by dynamic Tranco apex domains and 201 by overlapping apex domains that activate DNS HTTPS records. Figures 3 illustrates an overall upward trend in the number of non-*Cloudflare* DNS providers with HTTPS records for dynamic Tranco and overlapping apex domains, respectively, ranging from 55 to 85. Table 3 shows the top non-*Cloudflare* DNS providers with HTTPS records.[5] Further details on domains with non-*Cloudflare* providers can be found in Appendix D.

Dynamic Tranco top 1M		Overlapping	
DNS provider	#. distinct domains	DNS provider	#. distinct domains
eName	185	GoDaddy	59
Google	159	Google	40
GoDaddy	105	NSONE	20
NSONE	79	Hover	11
Domeneshop	16	Domeneshop	6

Table 3: Top non-*Cloudflare* DNS providers from October 11th, 2023, to March 31st, 2024.

4.2.3 Inconsistent use of HTTPS *records.* During our NS measurement period (Oct 11th, 2023 to Mar 31st, 2024), the majority of apex domains with HTTPS records consistently maintained the HTTPS records once published. However, we observe 4,598 apex domains that intermittently activated HTTPS records. We investigate this intermittent use of HTTPS records by domains in relation to their name servers. Among these 4,598 apex domains, 2,719 (59.13%) domains consistently use the same name servers. Among the 2,719 domains, 2,673 (98.31%) exclusively utilize *Cloudflare* name servers, while 46 (1.69%) employ either non-*Cloudflare* name servers or both *Cloudflare* and non-*Cloudflare* name servers.

Employment of multiple name servers. We find that 1,593 apex domains exclusively utilize *Cloudflare* name servers for activating their corresponding HTTPS records. However, during deactivation, they utilize a mixture of *Cloudflare* and other name servers. Considering that public DNS resolvers may employ specific mechanisms for selecting the most suitable DNS authoritative servers to query, we conduct additional scans by directly querying HTTPS records from corresponding DNS authoritative servers for all apex domains. We then identify 6 apex domains with fewer returned HTTPS records than corresponding name servers, due to the fact that they employ a combination of DNS providers that both support and do *not* support HTTPS records. HTTPS records are consistently returned when querying name servers that support HTTPS records, while none are returned by those that do not support HTTPS records. These findings suggest that employing a mix of multiple DNS service providers, where not all provide support for HTTPS records, may lead to inconsistent activation of HTTPS records due to public resolvers' selecting mechanisms.

Change of name servers. We observe that 236 apex domains lose their HTTPS records when they switch their name servers from *Cloudflare* to non-*Cloudflare* name servers (172 unique ones in total). Additionally, 20 apex domains do not have corresponding name server records (i.e., NS records) when they deactivate HTTPS records, and these domains are all found to employ *Cloudflare* name servers when they have HTTPS records. These results suggest that one likely reason for having intermittent HTTPS records is due to the change of name server, e.g., where the new name server does not support HTTPS records by default.

[4]We use *ipwhois* [25] to parse WHOIS information.

[5]Akamai only supports HTTPS records on its Edge DNS service [38], primarily used for hosting TLD registries. This is consistent with our finding that no HTTPS RR is configured by Akaimai for top 1M domains.

Change in configuration. We observe 2,673 domains who exclusively use *Cloudflare* name servers despite having intermittent HTTPS records. One likely reason is that *Cloudflare* will automatically generate an HTTPS record for the domain when it has the "proxied" option turned on. Thus, if the domain turns the "proxied" option on and off, it will result in intermittent HTTPS records. We further investigate such default configuration by *Cloudflare* next in Section 4.3.1.

> ***(Takeaway)*** Though *Cloudflare* remains the major adopter of HTTPS records, there is a noticeable upward trajectory in support for HTTPS among other prominent DNS providers. However, using multiple DNS service providers where not all of them support HTTPS records may result in inconsistent and intermittent HTTPS records for a domain.

4.3 HTTPS RR Parameters

We now investigate how domains configure parameters in their HTTPS records. Given *Cloudflare*'s dominating presence, we examine *Cloudflare* name servers and non-*Cloudflare* name servers separately.

4.3.1 Cloudflare HTTPS RR practices. We investigate the HTTPS record configuration process of *Cloudflare* by registering our domain to *Cloudflare*'s DNS service in January 2024 using a free account (features and default settings may differ with paid plans). *Cloudflare* offers a `proxied records` feature [16]. This feature, when enabled by a service user (i.e., domain owner), redirects all traffic destined for the original host to *Cloudflare*'s proxy server, which then forwards it to the original host. For instance, if a domain owner enables the proxied function for its A record, DNS queries for that record will resolve to *Cloudflare*'s Anycast IP addresses instead of the original IP addresses set by the domain owner. This feature is enabled by default when a domain owner adds a new DNS record via *Cloudflare*'s DNS configuration dashboard.[6] Moreover, once the proxied feature is activated, *an* HTTPS *record is automatically added* for a domain with default parameters (IP hints are specified as the IP addresses of *Cloudflare*'s Anycast IPs):

```
a.com. 300 IN HTTPS 1 . alpn=h2,h3
ipv4hint=a.b.c.d ipv6hint=e:f::g
```

Furthermore, this default HTTPS record cannot be modified and other HTTPS records cannot be added if a domain is using the proxied feature. A domain can only set its own HTTPS records after disabling the proxied feature.

Based on this information, we divide domains employing *Cloudflare* name servers into two groups: those with default *Cloudflare* HTTPS RR configuration and those with customized HTTPS RR configuration. We identify each domain's category by comparing its HTTPS record configuration against *Cloudflare*'s default configuration for free users. The corresponding ratios are presented in Table 4.

We observe that over 79% and 72% of dynamic and overlapping domains, respectively, adhere to the default HTTPS record configuration provided by *Cloudflare*. On the other hand, the remaining 20% and 28% of dynamic and overlapping domains, which have customized HTTPS configuration parameters, may likely be aware of their HTTPS records. Thus, this portion might serve as a conservative estimate for the number of domains potentially aware of their HTTPS record usage. Hence, in our following analysis, we distinguish between domains that utilize *Cloudflare*'s default HTTPS record configuration and those with customized configuration.

HTTPS RR Configuration	Dynamic Tranco top 1M (%)	Overlapping (%)
Default	79.96	72.37
Customized	20.04	27.63

Table 4: The average ratio of domains using *Cloudflare* name servers that have default HTTPS record configuration v.s. customized configuration.

4.3.2 Other DNS providers' HTTPS RR practices. We investigate the configuration of HTTPS records associated with name servers from *Google* and *GoDaddy* as well, considering their popularity among DNS providers utilized by domains that support HTTPS records (See Table 3). We show HTTPS record configurations and corresponding ratios in Table 5.

	***Google* NS**	***GoDaddy* NS**
SvcPriority	1 (98.95%)	0 (99.19%)
TargetName	. (98.95%)	An alternative endpoint (99.19%)
alpn	- (95.11%)	- (99.19%)
ipv4hint	- (97.76%)	- (99.19%)
ipv6hint	- (98.90%)	- (99.19%)

Table 5: Common HTTPS configurations by *Google* and *GoDaddy* name servers. Percentages indicate the ratio of domains from each name server with the specified configuration parameter. Dashes denote empty fields.

While the majority of HTTPS records with *Google* name servers are in `ServiceMode`, most of them have empty `SvcParams` fields, therefore not offering additional domain information. Conversely, we observe that 558 HTTPS records specify `alpn`, predominantly supporting HTTP/2, and 172 records incorporate `ipv4hint`. Note that one domain may be associated with multiple HTTPS records. Among all these domains, aside from 153 that are owned by *Google*, we observe 10 not affiliated with *Google*.[7] Notably, `err.ee` is the sole apex domain that utilizes `AliasMode`, aliasing to its `www` subdomain.

In contrast, most HTTPS records utilizing *GoDaddy* name servers are configured in `AliasMode`, redirecting to alternative endpoints. Among the remainder, which amount to 44 apex domains, 36 support both HTTP/3 and HTTP/2, while 8 exclusively support HTTP/2. Furthermore, 42 out of these 44 domains incorporate both `ipv4hint` and `ipv6hint` in their HTTPS records.

4.3.3 SvcPriority and TargetName. As explained in Section 2, an HTTPS records can be deployed in two distinct modes: `AliasMode` (value 0) which aliases a domain to the other domains, and `ServiceMode` (other values) which provides information associated with a specific service endpoint.

[6] A domain owner can explicitly disable this feature by switching off the enabled toggle.

[7] `cromwell-intl.com`, `fetlife.com`, `err.ee`, `wakeuplaughing.com`, `pixelcrux.com`, `stvincenttimes.com`, `dukarahisi.com`, `miranajewels.com`, `americancensorship.org`, and `smalls.com`.

We observe that the vast majority of domains use SvcPriority value of 1 (i.e., ServiceMode) with both *Cloudflare* name servers (99.97% and 99.95% of overlapping apex and www domains) and non-*Cloudflare* name servers (96.65% of overlapping apex domains). However, 202 apex domains do not provide any SvcParams even though they are in ServiceMode. On the other hand, among the remaining domains in AliasMode, we find that 19 domains set themselves as the TargetName (i.e., by using "." as value), which does not appropriately provide a true alias. Further details in Appendix E.1.

4.3.4 ALPN. A domain can specify the application protocols it supports through the alpn parameters in its HTTPS records. We observe that the prevalent protocols in alpn are HTTP/2 and HTTP/3, which is attributed to *Cloudflare*'s default configuration (Section 4.2.2). For domains with non-*Cloudflare* name servers, we observe a much lower ratio of advertising HTTP/2 and HTTP/3, with an average of 64.09% and 26.79%, respectively. Furthermore, 8.44% of domains do not include alpn parameters, and 1 domain continuously advertise draft versions 27 and 29 of HTTP/3. Further details are in Appendix E.2.

4.3.5 IP Hints. A domain can include ipv4hint/ipv6hint parameters (i.e., IP hints) in its HTTPS record to specify the IP addresses that clients can use to access the service it provides. We observe that over 97% and 87% of both apex and www overlapping domains adopt ipv4hint and ipv6hint, respectively. We further examine the consistency between the IP addresses provided in the IP hints and those in the corresponding A/AAAA records of the domains. We find that 624 and 5,109 of apex and www domains, respectively, have exhibited inconsistency, with an average of 6.57 and 14.52 days in duration of the inconsistency. Further details can be found in Appendix E.3.

Connectivity of domains with mismatched IP hints. One question that naturally arises when there is a mismatch between IP hints and A/AAAA is whether connections can be successfully established to both IPs. To answer this question, we conduct an additional experiment from January 24th to March 31st, 2024. For any apex domain with HTTPS records, we perform another sequence of queries of its HTTPS, A and AAAA records. After receiving the DNS responses, we promptly examine the consistency of its IP addresses and initiate TLS handshakes with all IP addresses in the HTTPS, A and AAAA responses through an OpenSSL client, if the domain has mismatched IP hints. This allows for an immediate validation of connectivity.

We identify a total of 1,022 occurrences of domains exhibiting IP mismatches (a domain is counted multiple times if it displays mismatched IP addresses on different days), of which 317 are distinct domain names. 5 domains[8] consistently demonstrate IP mismatches throughout the entire observation period. Subsequent TLS handshake attempts to these domains after each occurrence reveal that 193 domains have at least one unreachable IP address in either their IP hints and/or A/AAAA records, most commonly with an unreachable network error. Among these 193 domains, 117 can only be reached through the IP addresses in their IP hints, whereas 59 domains are accessible solely via their A record.

[8] cf-ns.net, cf-ns.com, canva-apps.cn, cloudflare-cn.com, polestar.cn

Our observations regarding the inconsistencies between IP hints and A/AAAA records highlight the complexities involved in managing HTTPS records. Domains need to synchronously update both their IP hints (in HTTPS records) and A/AAAA records whenever they change their IP addresses. However, given that most inconsistencies are resolved within a few days, such mismatches may often stem from *DNS caching effects*. For instance, even though a domain simultaneously updates both its HTTPS and A records, recursive DNS resolvers may continue to serve cached records until their TTL (Time to Live) expires. Therefore, inconsistency issues can emerge when the expiration timings of these cached records differ.

Moreover, our connectivity experiments indicate that these inconsistencies can make domains unreachable for clients if clients fail to utilize both HTTPS records and A/AAAA records. In Section 5, we investigate the browsers' behaviors in response to these issues.

> (***Takeaway***) The use of IP hints (ipv4hint/ipv6hint) adds an additional layer of complexity in maintaining IP addresses in both HTTPS and A/AAAA records. Incorrect IPs could make domains unreachable for clients, especially if clients do not implement robust failover mechanisms.

4.4 ECH Deployment

An important feature of HTTPS records is their ability to deliver TLS Encrypted Client Hello (ECH) configurations, which enables clients to send encrypted *ClientHello* messages to the server (associated with the domain).

4.4.1 ECH support . We identify domains with ECH parameter specified in their HTTPS records, and uncover a sudden change in ECH support on October 5th, 2023.

Before October 5th, 2023. The overall trend remains relatively stable, with about 70% of apex domains and 63% of www subdomains utilizing HTTPS records having adopted ECH. Considering that *Cloudflare* automatically activates ECH for free zones prior to October 5th, 2023 [31], the substantial deployment of ECH for both apex domains and www subdomains is consistent with our earlier observations that a significant portion of domains using HTTPS RR employ *Cloudflare* name servers with the proxied option enabled (Section 4.3.1). Specifically, 99.95% of ECH-enabled apex domains and www subdomains use *Cloudflare* name servers.

ECH was also activated by 106 apex domains and 74 www subdomains in conjunction with 48 non-*Cloudflare* name servers.[9] Interestingly, we find that *all* ECH configurations used by these domains direct to the same *Cloudflare* server, regardless of their name servers' DNS providers. We discuss this in further detail in Section 4.4.2.

After October 5th, 2023. A notable drop in the number of domains with ECH is observed on October 5th, 2023, resulting in *zero domains with ECH*.[10] Our finding was confirmed by Cloudflare's announcement that ECH features were disabled for all domains using

[9] The top three most utilized non-*Cloudflare* name servers are ubmdns.com, domainactive.org, and informadns.com.

[10] Cloudflare continues to operate one or two domains for testing ECH beyond October 5th, 2023. We exclude these domains, cloudflare-ech.com and cloudflareresearch.com, from our daily counts.

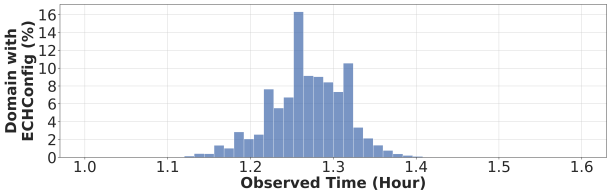


Figure 4: Percentage of domains based on the average duration of their ECH configuration.

their services [15], due to "a number of issues".[11] While we cannot directly confirm the exact reasons that lead to *Cloudflare*'s sudden disabling of ECH, we investigate potential challenges and issues in ECH usage—from both server-side (Section 4.4.2) and client-side (Section 5.3)—that can shed light on the future of ECH deployment.

4.4.2 Managing ECH with DNS caches. Since ECH configurations are delivered via HTTPS records and these records are cached by recursive DNS resolvers (or stub resolvers), domains have to carefully maintain their ECH configurations (i.e., ECH keys), taking into account *DNS caches (i.e., cached* HTTPS *records).* When the server changes its ECH configuration, it should maintain both the HTTPS record with the previous ECH configuration and the one with the new configuration to account for resolvers that may still have the old configuration in the cache. Otherwise, it could lead to inconsistencies between the ECH key delivered to a client via (cached) HTTPS records and the actual key recognized by the server. Similar types of key rotation (or key rollover) are known to be challenging [13, 30].

To further investigate the ECH key rotation frequency in HTTPS records, we conduct hourly scans on Tranco apex domains over seven days, from July 21 to July 27, 2023. As a result, we identify 169 unique ECH configurations, all directing to the same public client-facing server (`cloudflare-ech.com`), each associated with a distinct public key. Among these configurations, the majority (154) are consistently observed in two consecutive hourly scans; additionally, 13 configurations are presented in three consecutive hours and 2 are observed in only one hour. We notice that the TTL for over 99% of these HTTPS records is set to 300 seconds. Figure 4 depicts the distribution of domains based on the average duration their ECH configuration is observed. The duration periods range from 1.1 to 1.4 hours, with an overall average of 1.26 hours across all domains. This observation indicates that ECH keys (in these configurations) are rotated every one to two hours.

Such key rotation frequency implies the possibility of key inconsistency that may occur approximately every one to two hours, given the complexity of managing ECH keys in conjunction with DNS caches. As elaborated in Section 4.3.5, IP hints parameters (`ipv4hint`/`ipv6hint`) in HTTPS records face similar challenges with DNS caches. Therefore, to properly make use of ECH, it requires both servers to implement retry configuration and clients to correctly handle it (detailed in Section 5.3). Otherwise, it may prohibit encrypted TLS *ClientHello* messages between servers and clients, and even the failure of the connection.

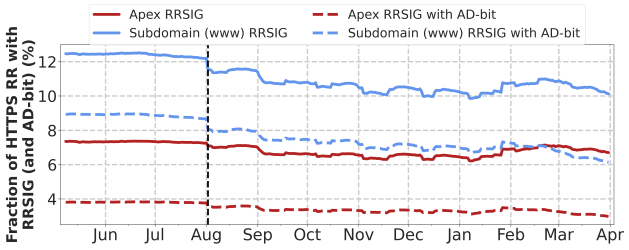
[11]While Cloudflare's announcement mentioned ECH re-enablement in early 2024, it has not been re-enabled at the time of this writing.

> (***Takeaway***) Managing ECH unavoidably adds additional complexity to the servers, especially considering the frequent key rotation. Both servers and clients need to handle ECH keys properly to avoid connection failures.

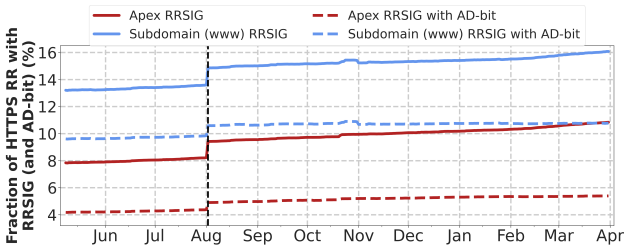
4.5 HTTPS RR and DNSSEC

DNSSEC [40–42] ensures the integrity and authenticity of DNS records. DNSSEC introduces three DNS record types; DNSKEY, RRSIG (Resource Record Signature), and DS (Delegation Signer) records. Although DNSSEC is optional in the current HTTPS record specification, it is critical to deploy DNSSEC for HTTPS records. Otherwise, it is susceptible to being dropped or forged by attackers, posing similar security risks to those encountered in traditional HTTP to HTTPS redirection (due to their plaintext transmission).

4.5.1 Signed HTTPS *records.* In our daily scan of HTTPS records, we simultaneously collect the corresponding RRSIG records, if provided by the servers. Using the collected RRSIG records, we examine the DNSSEC support of domains with HTTPS records. Figure 5 depicts the ratio of *signed* HTTPS records (i.e., HTTPS records that have the corresponding RRSIG records) in solid lines, and the ratio of *validated* HTTPS records (i.e., the `Authenticated Data (AD)` bit [52] is set in the DNS response) in dashed lines. We note that the observed ratio of RRSIG aligns with the trends in DNSSEC deployment [49].



(a) The fraction for dynamic Tranco top 1M domains.



(b) The fraction for overlapping domains.

Figure 5: Percentages of HTTPS records with RRSIG (solid line), RRSIG and AD bit (dashed line). Vertical dashed line (on August 1st, 2023) denotes Tranco source change.

Prevalence of signed HTTPS RR. The ratio of signed HTTPS records shows a decreasing trend for both apex and www subdomains (solid lines in Figure 5a) when using dynamic domain list. In contrast, the overlapping domains show increasing trends (solid lines in Figure 5b), indicating that the deployment of DNSSEC for HTTPS RR is growing for domains that are consistently in Tranco 1M. On the other hand, the relatively lower-ranked and possibly newly-added domains in the dynamic domain list are less likely to have

DNSSEC deployed already, contributing to the overall decreasing trend.

Interestingly, this trend is reversed for the overall HTTPS records deployment (in Figure 2); there is an increasing trend in HTTPS records deployment for dynamic domains but a decreasing trend for overlapping domains. This suggests that relatively higher-ranked domains are more likely to support DNSSEC than HTTPS records, possibly due to the recency of HTTPS records.

Validity of signed HTTPS RR. We examine the validity of the RRSIG records of HTTPS records by utilizing the AD bit [52] in the DNS responses for HTTPS records. The AD bit is set by a recursive resolver only if the DNSSEC chain of the corresponding DNS record has been verified. Figure 5 illustrates the ratio of validated HTTPS records in dashed lines, which is much lower than that of the signed records. For example, for overlapping apex domains, 47.8% of signed HTTPS records cannot be validated likely due to issues in their DNSSEC chain, leaving the authenticity of these HTTPS records in question. Upon further analysis, we find that the lack of validation is likely due to the well-known issue where domains use a third-party DNS operator instead of their registrar's DNS service, and consequently fail to upload necessary DS records themselves [14]. In our dataset, only 26% of apex domains with signed HTTPS records use the same DNS operator and registrar. Further details are in Appendix G.

4.5.2 ECH with DNSSEC. The low adoption rate of DNSSEC among domains publishing HTTPS records could additionally pose security concerns to the use of ECH. It is ironic that ECH delivers the server's public key to the client, and yet in the absence of DNSSEC, such key cannot be fully trusted. Unfortunately, our data shows that the vast majority of HTTPS records with ECH parameters are not signed, Before October 5th, 2023 (the date *Cloudflare* disabled ECH), less than 6% of overlapping domains with HTTPS and ECH are signed, and only half of them can be validated.

> ***(Takeaway)*** We observe very limited (<10%) DNSSEC deployment for HTTPS records, nearly half of which cannot be validated due to missing DS records. The lack of proper DNSSEC deployment leaves the majority of HTTPS records susceptible to attacks such as DNS cache poisoning.

5 Client-side HTTPS RR Support

So far, we have observed that over 20% of the top 1 million domains have adopted HTTPS records, with the majority employing various parameters, such as SvcPriority, alpn and IP hints. However, the effectiveness of these records depends significantly on their proper utilization by clients. Thus, an important question arises: how do clients, particularly web browsers, support the HTTPS records? In this section, we aim to examine the support of HTTPS records across popular web browsers [17], including Chrome, Safari, Edge,[12] and Firefox, to evaluate the potential impact on domains that have adopted these records. Specifically, we investigate (1) whether the browsers support HTTPS record lookup, (2) whether the browsers utilize the fetched HTTPS records in establishing connections, and (3) how they respond to misconfigured HTTPS records (i.e., failover behavior). Furthermore, we perform an in-depth analysis to assess the ECH support from the popular browsers.

[12]Both Edge and Chrome are built upon the Chromium [12] framework. However, we perform separate experiments on these browsers, as variations in their custom implementations may result in differing behaviors.

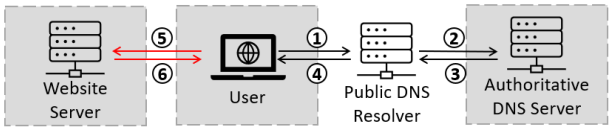


Figure 6: Experimental setup for client-side browser behavior analysis. Shaded regions denote controlled environments, namely the testing client, authoritative DNS server, and the target domain website server.

Experimental setups. To test web browsers' HTTPS record support, we set up a testbed, as illustrated in Figure 6. We configure our own domain and operate an authoritative name server using BIND9 [27] to configure various HTTPS record settings for our domain. Additionally, we set up a web server (accessible through our domain name) that supports ECH, utilizing the publicly available ECH (draft version 13) implementation of OpenSSL [19] and Nginx [18]. Both the name server and web server are hosted on Amazon Web Services (AWS) instances. Target web browsers run with default settings on separate machines with Microsoft Windows 11 (Home edition) and macOS Sonoma; Safari tests are omitted on Windows due to its lack of support. For DNS resolution, we utilize Google DNS resolver. For Firefox, we set its DoH server as: https://dns.google/dns-query. We choose to use a public resolver due to its widespread use and responsiveness, allowing us to focus on evaluating the browsers' HTTPS record support without the added complexity of varying resolver behaviors. Note that our test environment is configured for IPv4 addresses, and the interaction of browsers with IPv6 addresses remains as future work.

To eliminate the cache effect, in each round of the experiment, we begin by clearing the local DNS cache and resetting the browser history. We set the DNS record's Time-to-Live (TTL) to 60 seconds to ensure frequent refreshes of DNS records by the public resolver. Between experiments, we ensured that the interval exceeded the 60-second TTL to prompt public resolvers to fetch fresh DNS records (e.g., HTTPS records) from our authoritative DNS server. The test flow is as follows, as illustrated in Figure 6. Upon instructing the browser to access our domain, the browser sends DNS queries (e.g., for A or HTTPS record) to the DNS resolver (①), which then sends queries (②) to our authoritative name server (associated with our domain). Subsequently, the corresponding DNS response will be returned to our test browser (③~④). Finally, the browser starts the TLS handshake process with our web server (⑤~⑥). We repeat this process 5 times for each target browser and for each parameter setting.

5.1 Support by Popular Browsers

We first measure the overall support for HTTPS RR across four web browsers. One objective of HTTPS RR is to signal that the HTTPS protocol should be employed instead of the HTTP protocol when a client (e.g., browser) connects to a host (e.g., web server). To evaluate browser support for HTTPS record, we directed browsers to access three types of URLs: (i) a.com, (ii) http://a.com and (iii) https://a.com. The purpose of the first two URLs is to assess the

		Chrome		Safari	Edge		Firefox	
OS		macOS	Windows	macOS	macOS	Windows	macOS	Windows
Browser Version		120.0.6099		17.2.1	120.0.2210		122.0.1	
HTTPS RR Utilization	{apex}	●	●	◐	●	●	●	●
	http://{apex}	●	●	◐	●	●	●	●
	https://{apex}	●	●	●	●	●	●	●
AliasMode	TargetName	○	○	●	○	○	○	○
ServiceMode	TargetName	○	○	●	○	○	●	●
	port	○	○	●	○	○	●	●
	alpn	●	●	●	●	●	●	●
	IP Hints	○	○	●	○	○	●	●

Table 6: The HTTPS RR support from four major browsers. A full circle (●) means that the record or parameter being utilized. A half circle (◐) suggests that while the record or the parameter is being utilized, some essential function related to it is missing. An empty circle (○) denotes that there is no support for the feature.

utilization of the HTTPS record. For instance, following the retrieval of the HTTPS record, if a browser directly initiates an HTTPS connection when accessing the first two URLs, it indicates that the browser utilizes the HTTPS record as a signal of the web server's support for HTTPS. Therefore, this experiment aims to verify (1) whether browsers issue DNS queries for HTTPS RR and (2) whether they leverage the response to initiate an HTTPS connection as per the standard [45].

Our test domain has the following HTTPS record: ServiceMode (SvcPriority set to 1), TargetName pointing to itself (value "."), and alpn specifying support of h2 (HTTP/2). We also publish the A record that points to our web server.

```
a.com. 60 IN HTTPS 1 . alpn=h2
a.com. 60 IN A 1.2.3.4
```

The results are summarized in Table 6, inside "HTTPS RR Utilization". First, we observe that all the tested browsers request two distinct DNS queries, i.e., for HTTPS records and for A records, when visiting all three types of URLs. An HTTPS RR query is issued even when the target server lacks a corresponding HTTPS record, as the browser is unable to ascertain this beforehand. However, only Chrome, Edge, and Firefox[13] proceed to initiate HTTPS connection with web server across all URL variations, indicating their actual use of the HTTPS records as a signal for HTTPS support of the server. In contrast, Safari still establishes HTTP connections (i.e., via port 80) for the first two URL types (a.com and http://a.com), implying that it does not utilize the fetched HTTPS records (half circle in Table 6 indicating fetching HTTPS RR but not utilizing it).

> (***Takeaway***) While all four browsers send queries for HTTPS records, one browser, Safari, does not utilize the fetched HTTPS records to initiate secure connections.

5.2 Resolution of HTTPS RR Parameters

We next investigate how browsers utilize the parameters specified in HTTPS records. The HTTPS record specification [45] states that the resolution of HTTPS records relies on the client rather than the recursive resolver, thus we do not consider interventions from resolvers. Given that Safari only utilizes HTTPS records to initiate HTTPS connection when visiting https://a.com, we examine the behavior of all browsers by visiting our test domains with the https:// prefix.

5.2.1 AliasMode. AliasMode (SvcPriority value of zero) serves a crucial purpose by enabling aliasing at the zone apex, which cannot be conducted with CNAME records. In this mode, a domain specifies another domain it wishes to point to in the TargetName field. Consequently, with an AliasMode HTTPS record, a browser is expected to access the domain specified in the TargetName field (i.e., by issuing A record queries). To verify if browsers adhere to this anticipated behavior, we configure our DNS zone as follows:

```
a.com. 60 IN HTTPS 0 pool.a.com.
pool.a.com. 60 IN A 1.2.3.4
```

We observe that only Safari issues subsequent DNS queries for or establishes a connection with the server (at 1.2.3.4) that hosts pool.a.com., as indicated in Table 6 (AliasMode section). In contrast, the other three browsers simply try to access a.com. but fail due to the absence of an associated IP address with a.com.

5.2.2 ServiceMode. A domain can provide information about its alternative service endpoint accessible to browsers through ServiceMode. Specifically, an HTTPS record configured with ServiceMode can include parameters such as port, IP hints, and alpn. Here, we examine how browsers utilize these parameters provided in an HTTPS record. The ech parameter, another critical feature of the HTTPS record, will be discussed in the subsequent Section 5.3. Table 6 (ServiceMode section) summarizes our findings.

TargetName. In ServiceMode, the TargetName field specifies the domain name of the alternative service endpoints. We first examine if browsers utilize the domain name specified in this TargetName field. In the following setup, the HTTPS-RR-aware browser is expected to establish an HTTPS connection with the server (at 2.2.3.4) which hosts pool.a.com using the provided information (i.e., alpn).

```
a.com. 60 IN HTTPS 1 pool.a.com. alpn=h2
a.com. 60 IN A 1.2.3.4
pool.a.com. 60 IN A 2.2.3.4
```

We notice that both Safari and Firefox adhere to the domain name in the TargetName by issuing additional queries or accessing the

[13] Firefox only queries HTTPS records if using DNS over HTTPS (DoH) [8], which is enabled by default, while Chrome does not require DoH for HTTPS. Note that DoH is needed to preserve the privacy of the domain name (in addition to ECH), and DNSSEC is still needed to provide full protection.

server that hosts the domain. However, Chrome and Edge fail to utilize TargetName, resulting in failure of obtaining the right service. Therefore, in the subsequent experiments, we explicitly specify the owner name (i.e., the original domain that publishes HTTPS records) as the TargetName by setting its value to ".".

(1) port. Browsers are anticipated to connect to an alternative endpoint (specified in TargetName) using the port included in the port field. The following configuration directs browsers to use port 8443 for the HTTPS connection to the target server. Incorrect handling or ignoring of this parameter could lead to failure of establishing the connection.

```
a.com. 60 IN HTTPS 1 . alpn=h2 port=8443
```

Our observation reveals that neither Chrome nor Edge utilizes the port parameter; both browsers initiate connections directly to port 443, the default port number for HTTPS services, leading to a failure to access the web service. In contrast, Safari and Edge successfully initiate connections using the designated port (i.e., 8443).

port failover. To examine browser behavior in response to connection failures, we set up three server configurations: one accessible only on port 443, another only on port 8443, and a third on both ports. Chrome and Edge experience a hard failure when they cannot establish a connection on their initially attempted port (port 443). Safari and Firefox, however, demonstrates a fallback mechanism to port 443 if it fails to connect via the port suggested in the HTTPS record.

(2) IP hints. Browsers can bypass the additional A or AAAA records lookups by leveraging the IP hints (ipv4hint or ipv6hint). The HTTPS record specification states that if local A/AAAA records for TargetName exist, clients should ignore these hints. Otherwise, clients are encouraged to conduct A and/or AAAA queries for TargetName and use the retrieved IP addresses (in the A/AAAA records) for future connections. We interpret this as recommending clients prioritize IPs from A and AAAA records and configure our DNS zone as below to test how browsers choose between these IPs.

```
a.com. 60 IN HTTPS 1 . alpn=h2 ipv4hint=1.2.3.4
a.com. 60 IN A 2.2.3.4
```

As a result, we observe that both Safari and Firefox directly leverages the IP hints, while the Chrome and Edge prefer the IP addresses in A records.

IP hints failover. To explore how browsers respond when they fail to establish connections with their preferred IP addresses, we set up servers exclusively accessible via either the IP address specified in the ipv4hint or the one in the A record. Safari makes an initial connection attempt with the first available IP address. If this attempt fails, it immediately retries with the IP address specified in the alternate record type. In contrast, Firefox waits for a longer period before attempting to connect with a different IP address. Chrome and Edge experience a hard failure if unable to connect using the IP addresses in the A record.

(3) alpn. To ensure successful access to the service endpoint, browsers should employ an application protocol advertised in the alpn parameter. We configure two server setups, each exclusively advertising either the HTTP/2 (h2) or HTTP/3 (h3) protocol.

```
[1] a.com. 60 IN HTTPS 1 . alpn=h2
[2] a.com. 60 IN HTTPS 1 . alpn=h3
```

We observe that all four browsers successfully establish connections using the protocol specified in each HTTPS record. When the server exclusively advertises the HTTP/3 protocol, Firefox sends an HTTP/2 connection request shortly after initiating the correct protocol, possibly to ensure better compatibility. When the server supports only HTTP/2, Firefox does not initiate an additional HTTP/3 connection.

Code corroboration. We examine the Chromium code and note that as of February 8th, 2024, it does not accommodate subsequent queries in AliasMode and ServiceMode, indicating a lack of processing for follow-up queries for domain names in TargetName beyond the apex. Moreover, Chrome does not initiate service on a separate port and disregards HTTPS RR with an empty alpn parameter. Furthermore, our investigation confirms that Firefox interprets HTTPS parameters to establish connections in ServiceMode. These findings are consistent with our experimental results.

> (***Takeaway***) Although major browsers query HTTPS RR and fully support the alpn parameter, they often fail to properly utilize other associated HTTPS parameters. We identify several scenarios where inconsistent handling of HTTPS record parameters can result in divergent connection behaviors. In particular, such inconsistencies may direct connections to different IP addresses when there are mismatches between the A or AAAA record and the IP hint parameter in the HTTPS record, potentially leading to connection failures.

5.3 Browsers Support of ECH

We now investigate browser support for ECH. As discussed in Section 4.4.2, managing ECH presents challenges due to its reliance on DNS. Due to DNS caching, inconsistencies may arise between the ECH key (in the HTTPS records) and the actual key used by servers. Therefore, we examine how browsers respond to this issue, as well as other ECH misconfigurations. Ultimately, our goal is to offer insights into challenges in ECH deployment, and help inform and improve future ECH deployment.[14]

ECH workflow. We briefly describe the ECH workflow:

(i) To enable ECH, domain private-example-ech.com. publishes its ECH configuration via HTTPS records, including a public key for encrypting the *ClientHello*, an SNI extension directing to the client-facing server (that hosts public-example-ech.com.), and other metadata.
(ii) To access the target domain (i.e., private-example- ech.com.), a client fetches HTTPS records, parses the ECH configuration (in the ech parameter), and sends a *ClientHello* to the client-facing server (public-example-ech.com.). This *ClientHello* includes a private *ClientHello* (termed *ClientHelloInner*) that has an SNI pointing to the intended domain (i.e., private-example-ech .com.) and is encrypted using the key advertised in ech.
(iii) Upon receiving the *ClientHello*, the domain's client-facing server decrypts the encrypted *ClientHelloInner* using its private

[14] Recall that Cloudflare rolled back ECH deployment and has not re-enabled it at the time of this writing (Section 4.4).

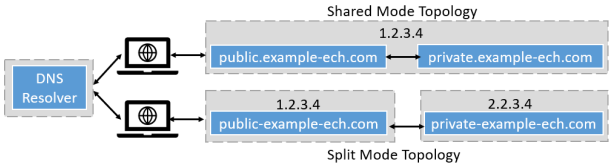


Figure 7: `Shared` and `Split` Mode of ECH operation.

	Chrome	Edge	Firefox
`Shared` Mode Support	●	●	●
(1) Unilateral ECH	●	●	●
(2) Malformed ECH	○	○	●
(3) Mismatched key	●	●	●
`Split` Mode Support	○	○	○

Table 7: Browser support and failover mechanisms of ECH. Same behavior is observed in Mac and Windows OS. Safari is excluded due to lack of any ECH support.

key associated with the ECH configuration. Successful decryption allows the client-facing server to forward the connection to the intended domain (i.e., `private-example-ech.com.` specified in an SNI of the *ClientHelloInner*). If decryption fails, the client-facing server either rejects and terminates the connection or sends "retry ECH configurations" to the client, which prompts the client to attempt the connection again using the newly provided ECH configuration.

There are two modes of ECH operation, `Shared Mode` and `Split Mode`, as shown in Figure 7). Each mode requires distinct configurations of `HTTPS` records. In `Shared Mode`, the client-facing server and the back-end server (e.g., web server) can be hosted on the same IP address, typically within the same apex zone (e.g., the same second-level domains). In `Split Mode`, the client-facing and back-end servers are hosted by a different apex zone and operate on separate IP addresses. We will examine the support for both modes.

5.3.1 `Shared Mode` *ECH support.* We set up `HTTPS` records as follows. The domain associated with the client-facing server is `cover.a.com.` and the domain associated with the back-end server is `a.com.`. Both domains' `A` records point to the same IP address (2.2.2.2). The ECH configuration is specified in the `ech` parameter.

```
a.com. 60 HTTPS 1 . alpn=h2 ech=......
a.com. 60 A 2.2.2.2
cover.a.com. 60 A 2.2.2.2
```

Upon directing the browsers to visit `https://a.com`, we note that three of the four browsers (except Safari) exhibit support for ECH, by initiating a handshake with the client-facing server and encrypting the SNI in the *ClientHelloInner*.

We next explore how browsers handle failover in the presence of misconfigured ECH through three experiments. Note that Safari, lacking ECH support, is omitted from our analysis. Our findings are summarized in Table 7.

(1) Unilateral ECH deployment. In a scenario where a server no longer supports ECH but the associated domain's `HTTPS` record continues to advertise ECH configuration, we examine the browsers' response. This situation could arise if a server discontinues ECH support without updating its `HTTPS` records to reflect this change. Alternatively, even if the ECH configuration is removed from the domain's DNS zone file, clients might still attempt to connect using the cached ECH configuration. Our findings indicate that three browsers successfully fallback to standard TLS connections.

(2) Malformed ECH configuration. We generate a malformed ECH configuration (e.g., due to typographical errors during copy-and-paste to zone files) that the browser cannot successfully parse. Chrome and Edge exhibit hard failure in the presence of malformed ECH configuration, terminating the connection after the initial SYN packet. This disrupts user access to the domain. In contrast, Firefox ignores the malformed ECH configuration and proceeds with a standard TLS handshake with the target server (i.e., `a.com`).

(3) ECH key mismatch. We publish a correct ECH configuration where the public key in the ECH diverges from the one utilized by the target server (i.e., the server that hosts `a.com`). Such inconsistencies can arise from a failure to account for DNS cache effect in `HTTPS` record management, as discussed in Section 4.4.2. The current ECH specification [39] outlines a server retry process (i.e., retry configuration) that can mitigate this problem. This process entails the client-facing server offering a valid ECH configuration for retry, thereby allowing a client to reinitiate the TLS handshake with the valid configuration. Our findings reveal that all three browsers support the retry mechanism and successfully establish connections with the target server (i.e., `a.com`) using the provided retry configuration. On the server-side, disabling retry is discouraged in the ECH specification [39] and is also not supported in the current ECH implementation of Nginx. We plan to further explore this aspect in future work.

5.3.2 `Split Mode` *ECH Support.* In the `Split Mode` topology, the client-facing and back-end servers may be hosted by separate entities, such as the ECH service provider and the website owner, across different apex zones and IP addresses. This introduces complexity in handling ECH configurations.

```
a.com. 60 HTTPS 1 . ech=..public_name=b.com..
a.com. 60 A 1.1.1.1
b.com. 60 A 2.2.2.2
```

Consider a scenario where the client-facing server, `b.com.` (specified as `public_name` in the `ech` parameter), operates on IP `2.2.2.2`, while the web server, `a.com.` (the domain the client intends to visit), is hosted on IP `1.1.1.1`. To establish connections successfully, the client must interpret the ECH configuration, conduct subsequent DNS queries to locate the IP address of the client-facing server associated with `b.com.`, and then initiate a *ClientHello* to this server.

However, our experiments reveal that all three browsers fail to execute follow-up queries (i.e., `A` records for `b.com.`) and incorrectly initiate connections directly to the back-end server (i.e., 1.1.1.1) associated with `a.com.`, using the incorrect SNI of `b.com.`. Consequently, the certificate validation process fails, leading to website loading failures across all three browsers; Chrome and Edge display an "ERR_ECH_FALLBACK_CERTIFICATE_INVALID" error, while Firefox shows a "We're having trouble finding that site." message.

(***Takeaway***) Although browsers (except for Safari) support ECH by default, certain essential features are still absent, especially in the `Split Mode` where all three browsers hard fail on the connection. Such lack of support significantly harms servers with ECH configurations where their connectivity can be disrupted.

6 Related Work

`HTTPS` RR Deployment. While there are numerous studies on the deployment of DNS record types (e.g., DNSSEC records), `HTTPS` (as well as `SVCB`) records have received scant attention, due to their recent introduction. There has been research [50] primarily examining the interaction between `HTTPS` records and `QUIC` deployment. However, this study used `HTTPS` records as a means to analyze the deployment of the `QUIC` protocol, rather than performing an analysis of the `HTTPS` records ecosystem. In 2023, Zirngibl and colleagues [51] performed scans on over 400 million domains within a 15-day timeframe to examine the deployment of `SVCB` and `HTTPS` records. They uncovered that about ten million domains support `HTTPS` records, with a majority hosted by *Cloudflare*. Jan Schaumann [44] released a brief analysis on the adoption and usage of `HTTPS` RRs in Tranco top 1M domains.

Our study diverges from these prior works in several ways. First, to the best of our knowledge, this is the first study to dissect browser support for `HTTPS` RR, including their failover mechanisms. Furthermore, we examine how browsers handle `ech` in various configurations (including misconfigurations). Second, although our datasets cover a smaller number of domains compared to [51], our focus extends to the longitudinal analysis of `HTTPS` records, including inconsistent use of `HTTPS` records, changes in DNS providers, IP address inconsistency, and domain connectivity. We also perform in-depth analysis to understand the implications of `HTTPS` parameters through additional experiments, such as the connectivity experiments to examine the reachability of mismatched IPs, additional scans to measure the `ech` key rotation frequency, and the conjunction with DNSSEC to understand the security protection. Last but not least, our analysis distinguishes between popular domains (overlapping) and relatively less popular domains (dynamic Tranco). This approach allows us to explore trends based on domain popularity more thoroughly.

ECH Adoption. Several studies investigated ECH. Chai *et al.* [9] examined Encrypted SNI (ESNI, a precursor to ECH) in the context of censorship circumvention. Bhargavan *et al.* [5] conducted an assessment of the security, privacy, and performance aspects of TLS 1.3 with and without ECH by employing automated verification tools. Tsiatsikas *et al.* [48] analyzed the adoption rates of both ESNI and ECH but found only one domain that supported ECH. Most recently, Zirngibl *et al.* [51] reported 20 domains utilizing ECH among 400 millions domains scanned in 2023.

We observed a marked increase in the usage of ECH as compared to previous research. Our longitudinal study indicates that *Cloudflare* began adopting ECH as early as May 2023, four months prior to their announcement [31]. Furthermore, while earlier studies do not address the service providers associated with ECH, our study not only examines these providers but also explores ECH key rotation and its integration with DNSSEC. Additionally, we assess ECH support across popular web browsers.

7 Discussion

Automation tool for `HTTPS` record management. Managing `HTTPS` records involves several complexities due to their DNS-based nature, including the coordination across multiple DNS service providers, potential inconsistencies between IP hints and `A`/`AAAA` records, and the risks associated with improper handling of ECH, which can lead to connection issues. We believe the DNS `HTTPS` ecosystem could borrow experiences learned from the management of digital certificates, where automating the certificate issuance and renewal process through `ACME` and `Certbot` [1, 46] has significantly reduced the barriers to obtaining and maintaining a digital certificate, demonstrating the potential benefits of automation in managing web security features.

Limitations in major browsers. Our assessment indicates that all four leading web browsers currently support querying `HTTPS` records, which points to a promising trend in industry adoption. However, several crucial functionalities remain unimplemented. Notably, both Chrome and Edge lack support for IP hints—a parameter utilized by 97% of apex domains and 87% of www domains that have adopted `HTTPS` records. Furthermore, the absence of support for ECH `Split Mode` could potentially lead to service disruptions. Cloud providers and domain administrators should take these limitations into account when integrating `HTTPS` records into their systems.

8 Conclusion

In this study, we present a comprehensive analysis of the DNS `HTTPS` record ecosystem, uncovering the deployment challenges and complexities from both server-side and client-side perspectives. Specifically, our server-side analysis shows that over 20% of domains in the Tranco list support `HTTPS` records, with *Cloudflare* playing a crucial role in this adoption and a noticeable increase in support from other major DNS providers as well. However, a significant concern is the lack of DNSSEC protection for many `HTTPS` records, particularly those utilizing ECH, which renders them vulnerable to potential attacks. We also explore the complexities of managing `HTTPS` records, including issues related to IP hints and ECH configurations. On the client side, while the four major web browsers support `HTTPS` record lookups, they do not fully utilize the capabilities offered by `HTTPS` records. Our analysis reveals that improper handling of `HTTPS` records can lead to connection failures, shedding light on the obstacles that we need to overcome to move towards a more widespread `HTTPS` deployment. We plan to reach out to DNS providers and web browsers regarding our findings.

Acknowledgments

We thank anonymous reviewers for their insightful and constructive suggestions and feedback. This work is supported by National Science Foundation CNS-2154962 and CNS-2319421, and the Commonwealth Cyber Initiative.

References

[1] Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, et al. 2019. Let's Encrypt: an automated certificate authority to encrypt the entire web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2473–2487.

[2] Akamai. 2020. New SVCB & HTTPS Resource Records in the wild. https://community.akamai.com/customers/s/article/NetworkOperatorCommunityNewSVCBHTTPSResourceRecordsinthewild20201128135350?language=en_US (accessed Aug 26, 2024).

[3] David Barr. 1996. Common DNS Operational and Configuration Errors. RFC 1912. https://doi.org/10.17487/RFC1912

[4] David Belson and Lucas Pardue. [n. d.]. Examining HTTP/3 usage one year on. https://blog.cloudflare.com/http3-usage-one-year-on (accessed Aug 26, 2024).

[5] Karthikeyan Bhargavan, Vincent Cheval, and Christopher Wood. 2022. A symbolic analysis of privacy for tls 1.3 with encrypted client hello. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 365–379.

[6] BIND9. 2021. BIND9 v9.16.21 Release notes. https://bind9.readthedocs.io/en/v9_16_21/notes.html#new-features (accessed Aug 26, 2024).

[7] Bugzilla. 2020. Implement HTTPSSVC. https://bugzilla.mozilla.org/show_bug.cgi?id=1623126 (accessed Aug 26, 2024).

[8] Bugzilla. 2023. Allow resolving HTTPS RR with native DNS. https://bugzilla.mozilla.org/show_bug.cgi?id=1852752 (accessed Aug 26, 2024).

[9] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. 2019. On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention. In *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*.

[10] Chrome Platform Status. 2020. Feature: TLS Encrypted Client Hello (ECH). https://chromestatus.com/feature/6196703843581952 (accessed Aug 26, 2024).

[11] Chrome Platform Status. 2021. Feature: HTTP->HTTPS redirect for HTTPS DNS records. https://chromestatus.com/feature/5485544526053376 (accessed Aug 26, 2024).

[12] Chromium. 2024. The Chromium Projects. https://www.chromium.org/chromium-projects/ (accessed Aug 26, 2024).

[13] Taejoong Chung, Roland van Rijswijk-Deij, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. 2017. A Longitudinal, End-to-End View of the DNSSEC Ecosystem. In *26th USENIX Security Symposium (USENIX Security 17)*. 1307–1322.

[14] Taejoong Chung, Roland van Rijswijk-Deij, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. 2017. Understanding the role of registrars in DNSSEC deployment. In *Proceedings of the 2017 Internet Measurement Conference*. 369–383.

[15] Cloudflare Community. 2023. Early Hints and Encrypted Client Hello (ECH) are currently disabled globally. https://community.cloudflare.com/t/early-hints-and-encrypted-client-hello-ech-are-currently-disabled-globally/567730 (accessed Aug 26, 2024).

[16] Cloudflare Docs. 2024. Proxy status. https://developers.cloudflare.com/dns/manage-dns-records/reference/proxied-dns-records/ (accessed Aug 26, 2024).

[17] Cloudflare Radar. 2023. Browser Market Share Report for 2023 Q3. https://radar.cloudflare.com/reports/browser-market-share-2023-q3 (accessed Aug 26, 2024).

[18] DEfO. 2024. Nginx, ECH-draft-13c branch. https://github.com/sftcd/nginx/tree/ECH-experimental (accessed Aug 26, 2024).

[19] DEfO. 2024. OpenSSL, ECH-draft-13c branch. https://github.com/sftcd/openssl/tree/ECH-draft-13c (accessed Aug 26, 2024).

[20] David Dittrich, Erin Kenneally, et al. 2012. *The Menlo Report: Ethical principles guiding information and communication technology research.* Technical Report. US Department of Homeland Security.

[21] Patrick R. Donahue. 2021. Upgrading the Cloudflare China Network: better performance and security through product innovation and partnership. https://blog.cloudflare.com/upgrading-the-cloudflare-china-network (accessed Aug 26, 2024).

[22] Alessandro Ghedini. 2020. Speeding up HTTPS and HTTP/3 negotiation with... DNS. https://blog.cloudflare.com/speeding-up-https-and-http-3-negotiation-with-dns (accessed Aug 26, 2024).

[23] Arnt Gulbrandsen and Dr. Levon Esibov. 2000. A DNS RR for specifying the location of services (DNS SRV). RFC 2782. https://doi.org/10.17487/RFC2782

[24] Bob Halley. 2020. DNSPython. https://www.dnspython.org/ (accessed Aug 26, 2024).

[25] Philip Hane. 2015. Ipwhois. Retrieve and Parse WHOIS Data for IPv4 and IPv6 Addresses. https://pypi.org/project/ipwhois/ (accessed Aug 26, 2024).

[26] Jeff Hodges, Collin Jackson, and Adam Barth. 2012. HTTP Strict Transport Security (HSTS). RFC 6797. https://doi.org/10.17487/RFC6797

[27] ISC. 2024. BIND9, Versatile, classic, complete name server software. https://www.isc.org/bind/ (accessed Aug 26, 2024).

[28] Knot DNS. 2021. Knot DNS Version 3.1.0. https://www.knot-dns.cz/2021-08-02-version-310.html (accessed Aug 26, 2024).

[29] Victor Le Pochat, Tom Van Goethem, S Tajalizadehkhoob, and Wouter Joosen. 2019. TRANCO: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Network and Distributed Systems Security (NDSS) Symposium 2019*.

[30] Hyeonmin Lee, Md Ishtiaq Ashiq, Moritz Müller, Roland van Rijswijk-Deij, Taejoong Chung, et al. 2022. Under the Hood of DANE Mismanagement in SMTP. In *31st USENIX Security Symposium (USENIX Security 22)*. 1–16.

[31] Achiel van der Mandele, Alessandro Ghedin, Christopher Wood, and Rushil Mehra. 2023. Encrypted Client Hello - the last puzzle piece to privacy. https://blog.cloudflare.com/announcing-encrypted-client-hello (accessed Aug 26, 2024).

[32] Mozilla Wiki. 2022. Security/Encrypted Client Hello. https://wiki.mozilla.org/Security/Encrypted_Client_Hello (accessed Aug 26, 2024).

[33] NLnet Labs. 2024. Unbound. https://nlnetlabs.nl/projects/unbound/about/ (accessed Aug 26, 2024).

[34] Mark Nottingham, Patrick McManus, and Julian Reschke. 2016. HTTP Alternative Services. RFC 7838. https://doi.org/10.17487/RFC7838

[35] Craig Partridge and Mark Allman. 2016. Ethical considerations in network measurement papers. *Commun. ACM* 59, 10 (2016), 58–64.

[36] Tommy Pauly. 2020. DNS HTTPS/SVCB record type support in iOS 14. https://mailarchive.ietf.org/arch/msg/quic/sFgifP9vOY9xsmogVqiq-qtxPiQ/ (accessed Aug 26, 2024).

[37] PowerDNS. 2024. Using SVCB and derived records. https://doc.powerdns.com/authoritative/guides/svcb.html (accessed Aug 26, 2024).

[38] Sam Preston. 2022. Akamai Blog. Edge DNS and the Top-Level Domain Hosting. https://www.akamai.com/blog/edge/edge-dns-and-the-top-level-domain-hosting (accessed Aug 26, 2024).

[39] Eric Rescorla, Kazuho Oku, Nick Sullivan, and Christopher A. Wood. 2023. *TLS Encrypted Client Hello.* Internet-Draft draft-ietf-tls-esni-17. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-tls-esni/17/ Work in Progress.

[40] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends. 2005. DNS Security Introduction and Requirements. RFC 4033. https://doi.org/10.17487/RFC4033

[41] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends. 2005. Protocol Modifications for the DNS Security Extensions. RFC 4035. https://doi.org/10.17487/RFC4035

[42] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends. 2005. Resource Records for the DNS Security Extensions. RFC 4034. https://doi.org/10.17487/RFC4034

[43] Scott Rose and Wouter Wijngaards. 2012. DNAME Redirection in the DNS. RFC 6672. https://doi.org/10.17487/RFC6672

[44] Jan Schaumann. 2023. Use of HTTPS Resource Records. https://www.netmeister.org/blog/https-rrs.html?utm_source=pocket_saves (accessed Aug 26, 2024).

[45] Benjamin M. Schwartz, Mike Bishop, and Erik Nygren. 2023. Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records). RFC 9460. https://doi.org/10.17487/RFC9460

[46] Christian Tiefenau, Emanuel von Zezschwitz, Maximilian Häring, Katharina Krombholz, and Matthew Smith. 2019. A usability evaluation of Let's Encrypt and Certbot: usable security done right. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1971–1988.

[47] Tranco. 2024. Methodology. https://tranco-list.eu/methodology (accessed Aug 26, 2024).

[48] Zisis Tsiatsikas, Georgios Karopoulos, and Georgios Kambourakis. 2022. Measuring the Adoption of TLS Encrypted Client Hello Extension and Its Forebear in the Wild. In *European Symposium on Research in Computer Security*. Springer, 177–190.

[49] Masanori Yajima, Daiki Chiba, Yoshiro Yoneya, and Tatsuya Mori. 2021. Measuring adoption of DNS security mechanisms with cross-sectional approach. In *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.

[50] Johannes Zirngibl, Philippe Buschmann, Patrick Sattler, Benedikt Jaeger, Juliane Aulbach, and Georg Carle. 2021. It's over 9000: Analyzing Early QUIC Deployments with the Standardization on the Horizon. In *Proceedings of the 21st ACM Internet Measurement Conference*. 261–275.

[51] Johannes Zirngibl, Patrick Sattler, and Georg Carle. 2023. A First Look at SVCB and HTTPS DNS Resource Records in the Wild. In *International Workshop on Traffic Measurements for Cybersecurity 2023*.

[52] Ólafur Guðmundsson and Brian Wellington. 2003. Redefinition of DNS Authenticated Data (AD) bit. RFC 3655. https://doi.org/10.17487/RFC3655

A Ethics

Throughout our scanning activities, we strictly adhered to ethical standards [20, 35]. Our study potentially impacts two entities: DNS resolvers and authoritative name servers hosting Tranco Top 1M domains. To mitigate any negative effects, we take the following

precautions. First, we conduct our scans at a controlled pace to ensure that we never overwhelm a single resolver with numerous concurrent requests simultaneously. Next, we limit our data retrieval to only the necessary DNS records for our analysis (as outlined in Table 1), which we collect once daily. For specific analyses, we conduct additional scans (e.g., for DNSSEC records). We consider the load placed on the name servers of domains within the Tranco list due to our DNS scans is negligible, especially considering their popularity (and high levels of traffic). Additionally, we clearly identified our measurement vantage point through DNS and WHOIS information, and a maintained testbed including a hosted domain and an operated authoritative name server using BIND9. Notable, we did not encounter any inquiries regarding our scans throughout this endeavor.

B Background

DNS record types. DNS records, known as resource records, are entries allowed in DNS zone files that serve to associate domain names with IP addresses and offer additional information about domains. We introduce DNS record types used in this work as follows (HTTPS records, the main focus of this study, will be detailed in the subsequent paragraph):

- A maps a domain name to its IPv4 address.
- AAAA maps a domain name to its IPv6 address.
- CNAME (Canonical Name) creates an alias from one domain name to another. When a DNS resolver encounters a CNAME record, it replaces the original domain name with the canonical domain name specified in the record and then performs a new DNS lookup using the canonical name.
- SOA (Start of Authority) holds information about a DNS zone. This record is crucial for managing DNS zones and ensuring the proper functioning of DNS services.
- NS (Name Server) stores information of the authoritative name servers for a domain.
- RRSIG (Resource Record Signature) stores cryptographic signatures of DNS records, used to authenticate records in accordance with DNSSEC (DNS Security Extensions) [40–42]. If the RRSIG record passes validation, the integrity of the given DNS record is ensured.
- DNSKEY (DNS Public Key) holds a cryptographic public key used to verify an RRSIG record (of a given DNS record).
- DS (Delegation Signer) contains the hash of a key (in DNSKEY) and is uploaded to the parent DNS zone to create a chain of trust across the DNS hierarchy.

C Domains in the Tranco List

Given the daily updates to the Tranco list, the list's domain composition can change daily. Therefore, relatively popular domains (e.g., those with higher rankings) are likely to consistently appear in the list, while less popular domains (i.e., those with lower rankings) may not consistently included in the list. Figure 8 shows the distribution of average popularity rankings for (apex) domains that are consistently included in the Tranco list throughout the entire first measurement period (i.e., prior to the list's source change), compared to those that are not; say overlapping and non-overlapping domains, respectively.

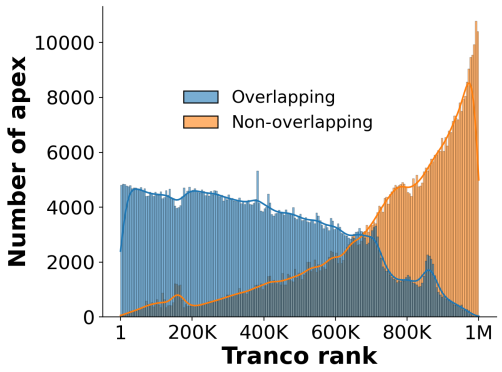


Figure 8: The distributions of Tranco rankings for each group of apex domains (overlapping or non-overlapping). The rank of each domain is averaged over the period from May 8th, 2023 to July 31st, 2023.

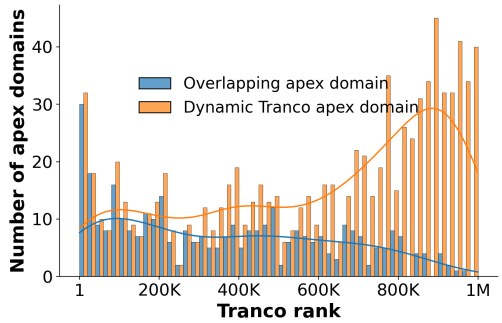


Figure 9: Ranking of apex using non-*Cloudflare* name servers. The rank of each apex is shown by its mean ranking value across the period from Oct 11th, 2023, to Jan 21st, 2024.

We observe that the overlapping domains tend to include domains with higher rankings compared to non-overlapping domains.

D Domains with Non-*Cloudflare* Name Servers

D.1 Ranking of Domains with Non-*Cloudflare* Name Servers

We show the ranking of these apex domains in Figure 9.

D.2 Domains with HTTPS records

We show the number of apex domains utilizing non-*Cloudflare* name servers during HTTPS record activation in Figure 10.

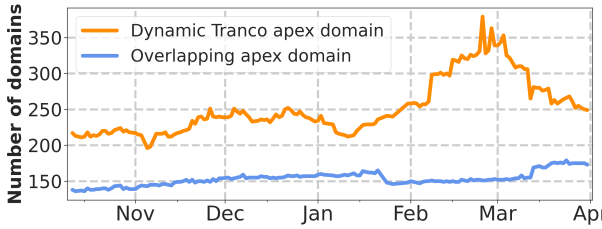


Figure 10: Number of domains that both activate HTTPS records and use non-*Cloudflare* name servers.

Protocols		% of domains with HTTPS RR	
		Apex	www
HTTP/2		99.64	99.61
HTTP/3		78.42	75.67
HTTP/3-29	< May 31st, 2023	77.43	74.32
	≥ May 31st, 2023	< 0.01	< 0.01
HTTP/3-27		< 0.01	0
HTTP/1.1		< 0.01	< 0.01

Table 8: Application layer protocols specified in the alpn parameter of overlapping domains, on a daily average. Protocols highlighted in red are those in *Cloudflare*'s default HTTPS record configuration.

E Details on HTTPS RR Parameters

We present additional details of the HTTPS RR parameters.

E.1 SvcPriority and TargetName

During our measurement period, the SvcPriority value of 1 (i.e., ServiceMode) is adopted by 99.97% and 99.95% of HTTPS RR on average, for overlapping apex domains and www subdomains respectively. On the other hand, the SvcPriority value of 0 (AliasMode), meanwhile, is used by approximately 39 HTTPS records for apex domains and 7 for www subdomains, on a daily average, respectively.

The dominance of ServiceMode, indicated by the value 1, is largely due to domains utilizing *Cloudflare* name servers with the default HTTPS record configurations, as we discussed in the previous section (Section 4.3.1).

Domains with *Cloudflare* name servers. We observe that on a daily average, approximately 12 apex domains using *Cloudflare* name servers have customized HTTPS record configuration (i.e., setting their SvcPriority or TargetName differently from *Cloudflare*'s default configuration). Among these domains, 5 apex domains use SvcPriority value of 0 (i.e., AliasMode). However, one domain, newlinesmag.com, sets itself as the TargetName (i.e., by using "." as value), despite using the AliasMode. Other three apex domains (unze.com.pk, idaillinois.org, and pokemon-arena.net) diverge from standard practices by using IP addresses as their TargetName. Lastly, gachoiphungluan.com uses an HTTPS URL as its TargetName. Additionally, we note that 14 distinct apex domains specify the same TargetName, geo-routing.nexuspipe.com, with multiple SvcPriority values in their corresponding HTTPS RR. The SvcPriority values for all these HTTPS records are a list including values ranging from 1 to 12, with each corresponding to a specific port. Interestingly, domain host-ir.com and pionerfm.ru keep only one HTTPS RR and they use priority 443 and 1800, respectively, for the record.

Domains with non-*Cloudflare* name servers. When examining apex domains that utilize non-*Cloudflare* name servers, we discover 2,884 such domains, with 2,755 (95.53%) of them using a SvcPriority value of 1 (i.e., ServiceMode) and setting their TargetName to point to themselves (by using the value of "."). We observe 9 domains using a SvcPriority value of 1 (i.e., ServiceMode) but setting their TargetName to point to alternatives. There are 108 domains that utilize the AliasMode (with SvcPriority value of 0), among which 22 apex domains set themselves as the TargetName; note that 21 out of 22 apex domains are with domaincontrol.com name servers and 1 employ {he.net, shaunc.com, and shat.net} as its name server hosts. Additionally, we again observe 7 domains with a list of priority values using non-*Cloudflare* name servers. These domains also specify the same TargetName (geo-routing.nexuspipe.com) with SvcPriority ranging from 1 to 12, with each assigned to a specific port. These domains are using sone.net name servers.

While SvcParams is an optional field reserved for ServiceMode, our observation reveals that the majority of domains employing ServiceMode include at least one key-value pair. In contrast, 232 apex domains utilize SvcPriority value of 1 (i.e., ServiceMode) but do not provide any SvcParams. This encompasses 42 DNS service providers, with the most prevalent being google.com, domaincontrol.com, netclient.no, icsn.com, nsone.net, {d-53.jp, d-53.net, and d-53.info}. The results for www domains are largely similar to the apex domains.

E.2 ALPN

The prevalent protocols in HTTPS RR are HTTP/2 and HTTP/3, garnering support from almost 100% for overlapping domains (and near 80% for www subdomains), as shown in Table 8. These substantial percentages align with our earlier observation that the majority of apex domains employ *Cloudflare* name servers and maintain an HTTPS RR configuration identical to *Cloudflare*'s default settings (see Section 4.2.2).

It is noteworthy that we observe massive support of the implemented draft version 29 of HTTP/3 prior to May 31st, 2023; starting from May 31st, 2023, we merely observe several support of this draft version on a daily basis. This aligns with the fact that as of late May 2023, *Cloudflare* no longer advertises this draft version for zones that have HTTP/3 enabled [4].

Among overlapping apex domains employing *Cloudflare* name servers with customized HTTPS records, HTTP/2 is supported by approximately 98.57% domains on a daily basis. In contrast to domains with *Cloudflare*'s default HTTPS record configurations, only 0.28% advertise their support for HTTP/3, and 1.13% do not include alpn in their SvcParams.

For apex domains utilizing non-*Cloudflare* name servers, we observe a lower ratio of advertising HTTP/2 and HTTP/3 as compared to *Cloudflare*'s default configuration, with an average of 64.09% and 26.79%, respectively. About 8.44% domains do not include alpn in their HTTPS records. Moreover, we observe 1 domains continuously advertise both draft version 27 and 29 of HTTP/3. This apex domain gentoo.org is with the gentoo.org name server, suggesting that it hosts its own apex zone. Specifically, we observe 6 apex domains exclusively advertise HTTP/1.1, of which 2 utilize a combination of name server jpberlin.de and cloudns.net, 2 employ jpberlin.de, 1 use a mix of gandi.net and trash.net, and the remaining 1 hosts it own apex zone.[15]

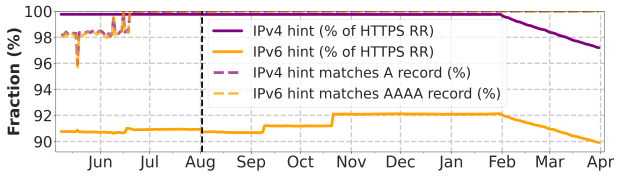
Among www subdomains, 1.63% and 0.18% do not indicate support for any alpn, for subdomains with customized HTTPS records using *Cloudflare* name servers and with non-*Cloudflare* name servers, respectively.

Additionally, starting from Feb 11th, 2024, we observe a consistent 0.003% of domains supporting *Google* QUIC version Q043, Q046, and Q050. These domains are all with *Cloudflare* name servers.

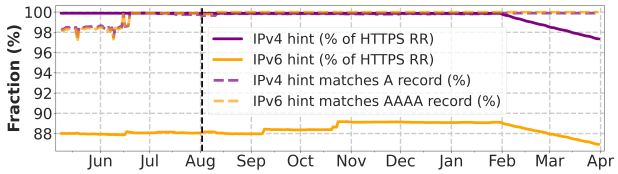
[15] he.net, shaunc.com, shat.net.

E.3 IP Hint Mismatching Analysis

Figure 11 show the ratio of overlapping domains that specify `ipv4hint`/`ipv6hint` in their HTTPS records (solid lines), as well as the consistency between the IP addresses provided in the IP hints and those in the corresponding A/AAAA records of the domains (dashed lines). Before June 19th, 2023, the matching rates fluctuated around 98% for both apex domains and www subdomains. However, starting from June 19th, 2023, the matching rates (for both apex domains and www subdomains) increased to over 99.8% for `ipv4hint` and `ipv6hint`, aligning with the corresponding A/AAAA records.



(a) IP hints utilization and consistency with A/AAAA records in overlapping apex domains.



(b) IP hints utilization and consistency with A/AAAA records in overlapping www subdomains.

Figure 11: The ratio of domains with HTTPS records that utilize IP hints (solid lines) and their matching ratio with IP addresses in A/AAAA records (dashed lines). The vertical dashed line (August 1st, 2023) denotes the source change of the Tranco list.

To take a closer look into the mismatched IP addresses in IP hints v.s. the corresponding A/AAAA records (Figure 11), we examine name servers utilized by these domains. Unfortunately, we lack information on name servers utilized by domains before August 16th, 2023 (as our NS records scan started on this date). To address this gap, we estimate their name servers by cross-referencing the name servers these domains utilized after August 16th, 2023. Based on this approach, we then continue our analysis on name servers.

IP hints and name servers with cross-referencing. Before June 19th, 2023, 40,578 of apex domains and 36,825 www domains (both about 2% of domains utilizing HTTPS RR) exhibit inconsistencies between their IP hints and A/AAAA records. As a result, we can estimate the name server for 88.08% of apex domains and 86.46% of www subdomains before June 19th, 2023. We observe that 99.97% of domains with mismatches between their IP hints and A/AAAA records utilize *Cloudflare* name servers; the remaining domains use `cf-ns.com` and `cf-ns.net` name servers.

Among these domains, about 64% exhibit inconsistency in both `ipv4hint` and `ipv6hint` (for both apex domains and www subdomains). While the majority of these domains display inconsistency in just a few days, we observe that 14 apex domains and 17 www subdomains consistently show such discrepancies for over 10 days. Furthermore, 5 apex domains and 8 www subdomains consistently

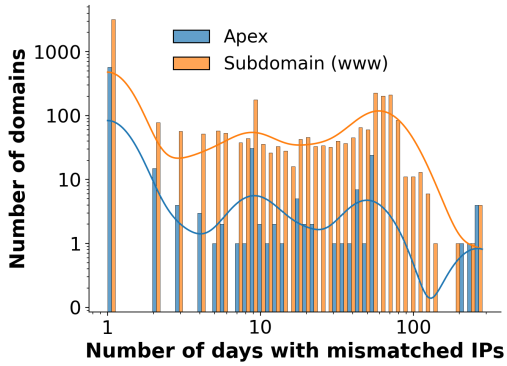


Figure 12: Duration of domains with mismatched IP hints and A/AAAA records.

present mismatched IP hints and A/AAAA records throughout the entire observation period, all of which are associated with *Cloudflare* name servers.

After the matching rate increase on June 19th, 2023, we identify discrepancies in IP hints and A/AAAA records for 178 apex domains and 814 www subdomains, through the cross-referencing of name servers, with daily discrepancies ranging from 30 to 80 domains. Among these, 94.94% and 98.89% are associated with *Cloudflare* name servers, while the remainder utilize `cf-ns.com`, `cf-ns.net`, `peavey.com`, and `upclick.com` name servers. Interestingly, 52 apex domains and 116 www subdomains are also found to advertise mismatched IP addresses before June 16th, 2023. Once again, the majority of these domains use *Cloudflare* name servers, with the rest employing `cf-ns.com` and `cf-ns.net` name servers.

IP hints and name servers after August 16th, 2023. Since we start scanning the name servers of domains on August 16th, 2023, we directly utilize the collected data for our analysis. We observe a total of 482 apex domains and 4,508 www subdomains advertising mismatched IP addresses, involving 91 and 34 DNS service providers, respectively. Notably, while `cf-ns` name servers are predominantly utilized by apex domains exhibiting such inconsistencies, www subdomains with mismatched IP addresses primarily employ *Cloudflare* name servers. Furthermore, we identify 4 apex domains and 4 www subdomains consistently advertising mismatched IP addresses from May 8th, 2023, to January 21st, 2024; all are associated with `cf-ns` name servers.

Mismatch duration. We monitor domains exhibiting discrepancies between IP hints and A/AAAA records commencing on June 19th, 2023, and illustrate the duration of these mismatches in Figure 12. In particular, we find 4 apex domains and 4 www subdomains that consistently provide mismatched IP addresses throughout our data collection period from May 2023 to March 2024; among these, 66.67% and 93.22% are associated with `cf-ns` (name servers for *Cloudflare* China Network, partnered with Chinese registrars [21]) and *Cloudflare* name servers, respectively.

F ECH Deployment by Domains

Figure 13 illustrates the ratio of domains that have deployed ECH (by publishing ech parameters) among those that publish HTTPS records.

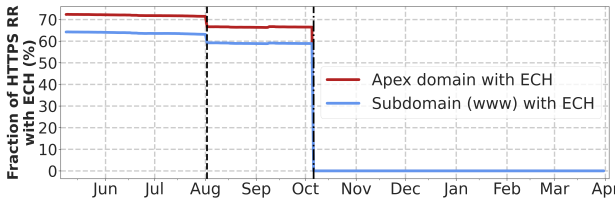


Figure 13: The percentage of overlapping domains with HTTPS records that support ECH. The vertical dashed line (near August 1st, 2023) denotes the source change of the Tranco list, and the vertical dash-dotted line (on October 5th, 2023) shows the date that *Cloudflare* disabled ECH from all its domains.

G DNSSEC Analysis of Apex Domains and Name Servers

Comparison with domains without HTTPS RR. We perform an additional data collection on January 2, 2024, where we fetch and validate the DNSSEC chain (i.e., DNSKEY, DS, and RRSIG records) of top 1M apex domains, using the Unbound library [33]. Table 9 shows the number of domains with signed records and their corresponding DNSSEC validation results. Interestingly, we find that 49.4% of signed HTTPS records are `insecure` (i.e., missing the DS records in their parent zone) [40]. This ratio is considerably high compared to the commonly known `insecure` ratio of domains that support DNSSEC; in our data, only 23.7% of signed domains (i.e., have a DNSKEY record) that do not publish HTTPS records are `insecure` (as indicated in the first row of Table 9), and this finding aligns with the similar `insecure` ratio of around 30% reported in [13]. We also examine the `insecure` ratio for both overlapping and non-overlapping domains and find no significant difference between them. Both groups exhibit high `insecure` ratios; 48.4% for overlapping domains (6,666 out of 13,762) and 53.6% for non-overlapping domains (1,656 out of 3,087), respectively.

Insecure HTTPS records and Name servers. However, when considering name servers, we found that *domains served by Cloudflare name servers exhibit a significantly higher* `insecure` *ratio* compared to those that do not. We find that 16,784 (99%) domains are served by *Cloudflare* name servers, and only 64 (1%) domains are served by other entities' name servers (e.g., other hosting providers).[16] Specifically, domains using *Cloudflare* name servers show a 49.5% `insecure` ratio, while those not using *Cloudflare* name servers have a 14.1% `insecure` ratio, as shown in Table 9. This notable discrepancy indicates that the high `insecure` ratio of domains with HTTPS records is primarily associated with domains using *Cloudflare* name servers.

Given the well-known issue that domains using a third-party DNS operator instead of their registrar's DNS service, often fail to upload necessary DS records themselves [14],[17] we further investigate the registrars of those domains. We specifically examine the congruence between DNS operators and registrars for domains supporting DNSSEC. To this end, we extract registrar information

[16]We are unable to retrieve the name server information for 11 domains.
[17]Conversely, if a domain uses its registrar as the DNS operator, the registrar is capable of autonomously generating and uploading the domain's DS records.

Category	Domains		
	Signed	Secure	Insecure
without HTTPS RR	46,850	35,688 (76.2%)	11,121 (23.7%)
with HTTPS RR	16,849	8,527 (50.6%)	8,322 (49.4%)
- *Cloudflare*	16,784	8,471 (50.5%)	8,313 (49.5%)
- Non-*Cloudflare*	64	55 (85.9%)	9 (14.1%)

Table 9: The number of domains with signed records and their DNSSEC validation results, as of January 2nd, 2024. Domains with HTTPS records are broken down based on their name servers (i.e., *Cloudflare* or Non-*Cloudflare*). We validate the DNSSEC chain of HTTPS records if a domain published HTTPS records, and the DNSSEC chain of DNSKEY records for a domain without HTTPS records. Note that bogus validation results are omitted as there are no bogus HTTPS records.

from additional Whois database searches.[18] We then analyze this congruence for two domain groups based on HTTPS record support: (i) signed domains *without* HTTPS records (i.e., the first row in Table 9) and (ii) signed domains *with* HTTPS records (i.e., the second row in Table 9). Here, we simply determine congruence by checking whether a domain uses name servers known to be associated with a registrar; for instance, if a domain's registrar is *Cloudflare* and it uses `amir.ns.cloudflare.com.` as a name server, this is classified as congruent. First, we observe that the (i) DNSSEC-supporting domains that do not publish HTTPS records have a 58% alignment between their DNS operator and registrar. Next, focusing on the (ii) DNSSEC-supporting domains with HTTPS records, we find that the top 10 popular registrars cover only 61.6% of these domains. This indicates a varied distribution of registrars of these domains, particularly given that 99% of the domains with HTTPS records are served by *Cloudflare* name servers. Consequently, only 26% of domains with signed HTTPS records use the same DNS operator and registrar (among these 99% use *Cloudflare* for both services), which potentially accounts for their higher `insecure` ratio.

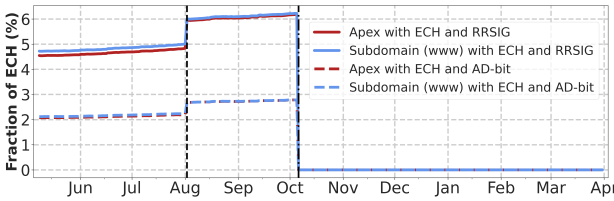


Figure 14: The percentage of overlapping domains with signed HTTPS records and ECH parameter. The vertical dashed line (August 1st, 2023) denotes the source change of the Tranco list, and the vertical dash-dotted line (on Oct. 5, 2023) shows the date that *Cloudflare* disabled ECH from its domains.

ECH with DNSSEC trend. Before October 5th, 2023 (the date Cloudflare disabled ECH), less than 6% of overlapping domains with HTTPS and ECH are signed, and only half of them can be validated, as shown in Figure 14. Note that the y-axis ticks represent percentages ranging from 0% to 7%.

[18]We are able to gather Whois information for 88% of domains with signed HTTPS records.