POLITECNICO DI TORINO
Repository ISTITUZIONALE

Encrypted Client Hello Is Coming: A View from Passive Measurements

Original
Encrypted Client Hello Is Coming: A View from Passive Measurements / Merlach, G., Trevisan, M., Giordano, D.. - In: NETWORK. - ISSN 2673-8732. - ELETTRONICO. - 5:3(2025). [10.3390/network5030029]

Availability:
This version is available at: 11583/3002952 since: 2025-09-11T10:55:18Z

Publisher:
MDPI

Published
DOI:10.3390/network5030029

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

19 November 2026

network

MDPI

Article

Encrypted Client Hello Is Coming: A View from Passive Measurements

Gabriele Merlach [1,*], Martino Trevisan [1] and Danilo Giordano [2]

[1] Department of Engineering and Architecture, University of Trieste, 34127 Trieste, Italy; martino.trevisan@dia.units.it
[2] Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 10129 Torino, Italy; danilo.giordano@polito.it
* Correspondence: gabriele.merlach@dia.units.it

Abstract

The Encrypted Client Hello (ECH) extension to Transport Layer Security (TLS) and the new type of Domain Name System (DNS) records called HTTPS represent the latest efforts to improve user privacy by encrypting the server's domain name during the TLS handshake. While prior studies have assessed ECH adoption from the server perspective, little is known about its usage in the wild from a passive network standpoint. In this paper, we present the first passive analysis of ECH and HTTPS DNS adoption using a month-long dataset collected from an operational network. We find that HTTPS DNS queries already make up approximately 8% of total DNS traffic, although responses to those queries are often incomplete, leading to increased query volume. Furthermore, 59% of QUIC flows include ECH, although only a negligible fraction is directed to servers supporting it. The remaining ECH flows are composed of GREASE values, intended to prevent protocol ossification. Our findings provide new insights into the current state and challenges in deploying privacy-enhancing protocols at scale.

Keywords: Encrypted Client Hello; HTTPS DNS; passive measurements; QUIC

check for updates

Academic Editor: Chin-Tser Huang

Received: 2 July 2025
Revised: 31 July 2025
Accepted: 5 August 2025
Published: 8 August 2025

Citation: Merlach, G.; Trevisan, M.; Giordano, D. Encrypted Client Hello Is Coming: A View from Passive Measurements. *Network* **2025**, *5*, 29. https://doi.org/10.3390/network5030029

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Passive monitoring refers to the process of observing and analyzing traffic within operational networks. While it has proven useful for applications such as traffic engineering, cybersecurity, and behavioral analysis, it also involves processing personal data. In particular, client IP addresses can often be linked to individuals, and the content of their network traffic may reveal sensitive information, such as visited websites, which can, in turn, expose user habits and characteristics. As such, passive monitoring poses significant privacy concerns for end users [1].

Over the past two decades, various efforts have aimed to enhance network privacy through encryption. These include the widespread adoption of Transport Layer Security (TLS) and Hypertext Transfer Protocol Secure (HTTPS) [2], as well as the more recent deployment of the QUIC protocol [3], which encrypts nearly all transport layer fields. However, these technologies do not fully protect user privacy. Although they encrypt application-layer payloads, domain names remain exposed in Domain Name System (DNS) queries and the Server Name Indication (SNI) field of the TLS Client Hello message.

To mitigate this issue, recent standards have introduced encrypted versions of DNS, namely DNS over TLS (DoT) [4] and DNS over HTTPS (DoH) [5], the latter of which is increasingly supported by major web browsers. Despite these advancements, encrypted DNS

only partially addresses the problem. However, in many networks, users are still forced to use local DNS resolvers, and DoH/DoT traffic can often be detected and blocked [6,7]. Moreover, even with encrypted DNS, domain names continue to be visible in the SNI field. To fully hide domain names from passive eavesdroppers, it is necessary to also encrypt the SNI field, which presents significant technical challenges. The SNI is essential for servers hosting multiple virtual hosts, as it allows them to select the correct TLS certificate. Simply removing the SNI is, therefore, not feasible.

The recent Encrypted Client Hello (ECH) draft [8] addresses this issue by proposing encryption of the SNI and other TLS handshake fields using a public key retrieved via DNS. The current ECH draft is an evolution of the earlier Encrypted SNI (ESNI) proposal, with the key difference being that ECH can encrypt any Client Hello field (or even the entire message), while ESNI only encrypted the Server Name Indication (SNI) within it. Specifically, the client queries a new type of DNS record called "HTTPS", which contains the server's public key and other connection metadata [9]. The server uses the corresponding private key to decrypt the ECH. These HTTPS records are designed to eventually replace traditional A and AAAA records, as they can also contain server IP addresses and application-layer protocol information (superseding ALPN negotiation).

In this paper, we present the first study of ECH and HTTPS DNS adoption from a passive monitoring perspective. Using traffic traces collected over one month from a campus network, we analyze DNS and QUIC traffic to evaluate the real-world deployment and use of these protocols. Our findings reveal that HTTPS DNS queries already account for approximately 8% of all DNS traffic. However, the responses often lack fundamental information, such as server IP addresses, forcing clients to issue additional A and AAAA queries. As a result, overall DNS traffic increases. We also observe that ECH is already used in a substantial share of connections, that is, 59% of QUIC flows include an ECH extension. Notably, clients often send ECH even when the destination server does not support it. Only a small subset of servers—all uniquely hosted on Cloudflare—actually support ECH. Among connections to these servers, 1.96% include a valid ECH. In the other cases, clients send an ECH containing random values called GREASE (Generating Random Extensions and Sustain Extensibility), which carries no meaningful data and has the sole purpose of preventing protocol ossification.

The remainder of the paper is organized as follows. Section 2 offers a background on ECH and summarizes related work, while Section 3 describes the methodology for data collection and the dataset we use. Section 4 presents our results regarding HTTPS DNS and ECH adoption, while, finally, Section 5 concludes the paper.

2. Background and Related Work

The Encrypted Client Hello (ECH) extension [8] enhances privacy in TLS by encrypting the entire Client Hello message, including sensitive fields such as the Server Name Indication (SNI), which would otherwise be visible in plaintext during the handshake. To achieve this, the client constructs two Client Hello messages—an outer Client Hello and an inner Client Hello. The outer Client Hello acts as a syntactically valid decoy that can be processed by servers that do not support ECH. It also includes a placeholder, in-clear SNI, called "ECH Public Name". The inner Client Hello contains the actual parameters of the TLS connection, including the real server name and any sensitive extensions. This inner Client Hello is then encrypted using the Hybrid Public Key Encryption (HPKE) scheme and embedded within the outer Client Hello as the payload of the `encrypted_client_hello` extension.

In addition to real ECH usage, clients may also send GREASE ECH values to help prevent protocol ossification. GREASE (Generate Random Extensions And Sustain Extensibility) was originally introduced in TLS to ensure that middleboxes and servers do not

incorrectly assume that certain protocol features will always be absent or follow a fixed structure. In the context of ECH, a GREASE ECH is a syntactically valid but semantically meaningless ECH extension, typically containing a random encrypted payload. These are deliberately constructed so that they cannot be decrypted by any server, and they serve no functional purpose in the handshake. Instead, their role is to keep the network ecosystem flexible by normalizing the presence of ECH extensions in Client Hello messages, even in cases where true ECH is not used. This reduces the risk of network interference or broken connections when ECH is eventually deployed at scale.

The server's ECH public key and configuration parameters are advertised through a DNS HTTPS resource record associated with the target domain. The HTTPS DNS resource record (RR), defined in RFC 9460 [9], is a new type of DNS record designed to allow clients to discover configuration details about a service endpoint, including those needed for Encrypted Client Hello (ECH). Unlike traditional A or AAAA records that simply map domain names to IP addresses, the HTTPS RR can carry metadata in the form of service parameters. These parameters can specify endpoint addresses (IPv4 or IPv6), supported protocols (e.g., QUIC), port numbers, and the ECH configuration. The ECH configuration includes the public key, cipher suites, and other encryption parameters necessary for clients to encrypt the inner Client Hello. This information is embedded as a base64-encoded blob within the HTTPS record.

When a client wishes to initiate a TLS connection using ECH, it first performs a DNS resolution for the HTTPS RR of the target domain. If the record is available and contains valid ECH configuration data, the client can construct the inner Encrypted Client Hello using HPKE. When a server that supports ECH receives a Client Hello with the ECH extension, it attempts to decrypt the inner Client Hello using its private HPKE key. If decryption is successful, it proceeds with the handshake using the decrypted values. If decryption fails, the server falls back to processing the outer Client Hello to ensure backward compatibility.

Adopting ECH and HTTPS DNS records requires significant infrastructural changes on the part of content providers, as well as updates to client implementations. The research community has thus begun to explore this transition. A number of articles tackled the impact of ECH on traffic classification [10–12], showing that machine learning models can, under some circumstances and provided an adequate training set, nullify the privacy benefits of ECH. Bbhargavan et al. [13] analyze the cryptographic correctness of the ECH proposal, while Niere et al. [14] study the relationship between ECH and state-level censorship systems.

The first study regarding ECH adoption was carried out by Tsiatsikas et al. [15] in 2022. The authors used a web crawler to verify ESNI and ECH support in the top 1M domains in terms of popularity. They found that only a small portion, less than 19%, supported the former ESNI extension, and practically no domain supports ECH. In 2023, Zirngibl et al. [16] studied ECH adoption by measuring the availability of ECH public keys in HTTPS DNS records. Across more than 400 M domains, they found 10.5 M HTTPS records. Cloudflare hosted and served most of those domains, and most records contained only Application-Layer Protocol Negotiation (ALPN) and IP address hints while missing the ECH public key. More recently, in 2024, Dong et al. [17] performed a similar study, offering a longitudinal perspective on the server-side deployment of DNS HTTPS for the Tranco top million domains. They observed a growing trend in DNS HTTPS adoption, and, despite its recent standardization, over 20% of probed domains had DNS HTTPS records. All the aforementioned works used active measurements in the form of web crawlers and DNS resolutions; thus, they targeted server-side deployment of ECH and HTTPS DNS. In contrast, in this paper, we use passive measurements, monitoring the operational traffic

of a population of real users. Thus, we offer a different and complementary perspective, measuring client-side adoption and watching ECH and HTTPS DNS in operation.

3. Dataset and Methodology

Our dataset was passively collected from a medium-sized university campus with approximately 35,000 students. The campus network is connected to the internet via two bidirectional 10 Gbit/s access links, providing connectivity to students and staff via Wi-Fi and Ethernet cable access. The campus organization operates an internal DNS resolver, which (i) is used as the local resolver by internal users and (ii) is the authoritative name server for the campus DNS zones (the campus operates two second-level DNS domains, and this DNS server is authoritative for both). The university does not support IPv6 addressing; therefore, all observed traffic is IPv4-based. While we expect client behavior regarding ECH to be consistent across IP versions, due to its origin in the application layer, this remains an open area for further study. Notice that our dataset, collected from a single university campus network, reflects the behavior of a specific user demographic—primarily students, faculty, and staff—whose device types, application usage, and network configurations may differ from those in residential or corporate environments. As such, our findings may not fully generalize to other network contexts, particularly where DNS policies or middlebox deployments differ significantly.

We deploy a passive measurement infrastructure depicted in Figure 1. We collected data using a passive monitor deployed at the campus edge and connected to the router via four span ports that mirror the router's traffic. The monitor uses the DPDK library (https://www.dpdk.org/) for packet capture and runs Tstat [18], a passive monitoring tool that records details of observed TCP and UDP flows. Tstat is open-source and available online (http://tstat.polito.it/). It is highly configurable and can produce log files regarding a number of events observed in the network. Specifically, for each TCP or UDP connection, Tstat tracks IP addresses and port numbers, as well as several metrics extracted from the Deep Packet Inspection (DPI) of the flow packets. It can produce other types of log files regarding HTTP, DNS, or RTP traffic. In the following paragraphs, we describe the fields relevant to our analyses. We gather data for the full month of February 2025 and analyze the log file related to DNS and QUIC traffic. We run queries on the logs using DPMon, a differentially private query engine for passive network measurements developed in our previous work [19] and released as open source at https://github.com/marty90/DPMon, accessed on 4 August 2025. DPMon runs directly on the probe. We leverage the following three datasets:

- *Outgoing DNS:* We record all DNS transactions initiated by the campus resolver to answer queries by internal clients. To this end, Tstat has been configured to produce a "DNS log" which records all the DNS transactions observed in the traffic. Indeed, all internal clients are configured to use the internal DNS resolver, which answers the client's recursive queries by contacting the required authoritative DNS servers. Our setup allows us to observe DNS traffic between the internal resolver and the internet, but we cannot record the individual DNS requests/responses issued by clients. Thus, we can collect the whole set of domains accessed by internal clients, but we cannot link them to the corresponding user. Moreover, some requests might be handled by the DNS server thanks to the cache, without the need for issuing any query to the external authoritative name server. For DNS transactions, Tstat exports the endpoint IP addresses and port numbers as well as the queries and resource records carried within packets. For each one, Tstat reports the domain name, DNS record type (A, AAAA, HTTPS, etc.), and the content of the record. Specifically, in the case of HTTPS records, it records the contained information if present—i.e., ECH keys, the

Application-Level Protocol Negotiation (ALPN) preferences, and the possible IPs in both 6 and 4 versions. We use this dataset to study the prevalence of HTTPS DNS queries in client traffic and the extent to which they are successfully resolved. Over the month of February, we collected 879 million DNS queries, resulting in 615 million responses to 105,000 unique domains. We refer to this dataset as DNS-Outgoing.

- *Incoming DNS:* We collect all the DNS transactions initiated by any external node and directed to the internal DNS resolver for the campus. This traffic includes (almost) unique requests issued by ISP/corporate/open resolvers operating on behalf of world-wide users accessing the domains belonging to the campus's two DNS zones, i.e., the campus websites. We record the same kind of data as for DNS-Outgoing, but we never observe any answer to DNS HTTPS queries, as the internal resolver does not hold any resource record of this type. Over the month of February 2025, we collected 220 million DNS queries and 64 million responses. We use this dataset to study the volume of HTTPS DNS traffic that an authoritative DNS resolver must expect as of 2025 and its adoption across different Internet service providers (ISPs). We refer to this dataset as DNS-Incoming.
- *QUIC traffic:* We also collect logs related to all QUIC connections by internal clients to external web servers. Indeed, Tstat identifies QUIC traffic thanks to Deep Packet Inspection, matching the QUIC headers contained in the first packets of each UDP flow. For each UDP flow identified as a QUIC connection, it records several metrics, such as size and timing of the first packets and the type of QUIC frames they carry. Moreover, in the case of "Initial" QUIC frames, Tstat decrypts the packet payload using the encryption keys mechanically derived from the packet headers. Thus, it obtains and records various fields of the contained TLS Client Hello message, such as the Server Name Indication (that is, the server's domain name) and the presence (and size) of an Encrypted Client Hello, which, obviously, we cannot decrypt. Notice that, when a client does not possess a valid ECH configuration, it generates a GREASE ECH extension—a randomized but syntactically correct placeholder—to maintain protocol consistency [20]. These GREASE values do not represent actual Encrypted Client Hello messages but serve to ensure that middleboxes and servers properly ignore unknown extensions. Over February 2025, we collected 106 million QUIC connections to 95,500 unique domain names. We refer to this dataset as QUIC-Outgoing.

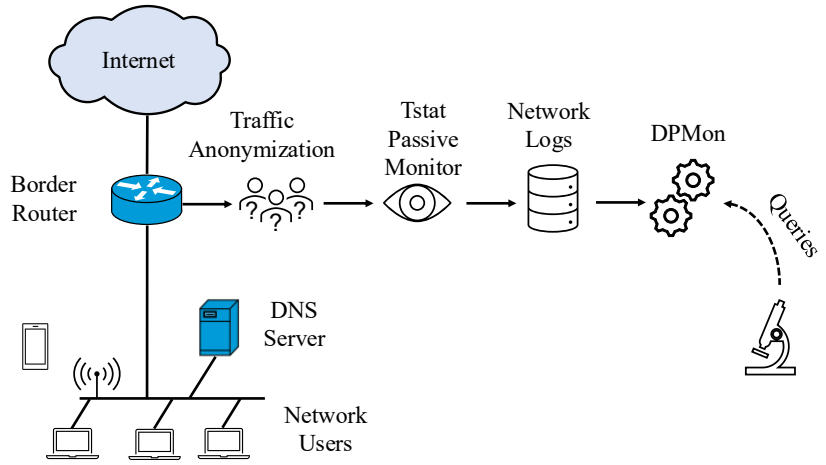


Figure 1. Network measurement setup.

Ethical Considerations

Passive monitoring involves capturing and processing traffic generated by human beings; thus, we need to take proper actions to protect as much as possible the individual's

privacy. Indeed, the IP address is considered Personally Identifiable Information (PII) and it can be used to identify and track individuals. The characteristics of traffic, such as the list of visited websites, can be considered Sensitive Personal Information (SPI), as they can reveal personal aspects and habits of an individual.

For this work, we take all possible countermeasures to properly handle our measurements. We configured the data collection to limit the exposed information as much as possible. We process packets in real time and save only strictly required information in flow logs. In detail, we do not store any information present in packets that can be associated with a single user, such as the HTTP headers of the packet payload. Clients' IP addresses are anonymized in real time using the CryptoPan algorithm [21], which preserves the subnet structure of the original IP addresses.

The anonymized logs are stored uniquely on the probe, which deletes all records older than two months. We run queries directly on the probe, exporting only the aggregate metrics that we use to produce the figures in the paper. The IP address anonymization with CryptoPan is performed directly by Tstat during packet capture. Our queries are run using DPMon [19], which enforces differential privacy on the results it provides to the user and runs directly on the probe. The probe is protected by a custom firewall and has strictly controlled physical access. The operating system and software are kept up to date to avoid possible vulnerabilities, and strict user access policies limit access to the data only to authorized users.

4. Results

In this section, we present the results of our analyses. We start by studying HTTPS DNS queries and responses, showing that they already represent a non-negligible fraction of the overall DNS traffic. We also dissect the content of HTTPS DNS responses, finding they are often incomplete. We then focus on ECH adoption, discussing the features observed from passive measurements.

4.1. HTTPS DNS Traffic

We first characterize the prevalence of HTTPS queries and responses in DNS traffic at the campus network border. In Table 1, we present a breakdown of the observed DNS packets, distinguishing between DNS-Outgoing and DNS-Incoming datasets. Starting with DNS-Outgoing, we observe that HTTPS DNS queries already represent a non-negligible proportion of 7.89% of total DNS traffic. Interestingly, AAAA requests represent 19.18% of the total, although internal clients are not assigned an IPv6 address (Likely, browsers issue AAAA queries even if they run on an IPv4-only host). Regarding responses, 48% are of type A, confirming that IPv4 is still prevalent. HTTPS-type responses account for only 0.89% of the total, as the vast majority of HTTPS queries do not receive any response—only 3.8% of HTTPS queries are answered. This is particularly interesting, as, at the current state, although HTTPS queries represent a sizable fraction, they typically remain unanswered as DNS servers do not yet support them, thus generating a useless load on clients and local/authoritative resolvers.

Passing to the DNS-Incoming dataset, overall, we again observe a large prevalence of type A queries (53.34%) and a smaller fraction of AAAA (17.68%). We find 5.48% of queries for HTTPS records, which, despite being lower than in DNS-Outgoing, represents a sizable fraction. Recall that DNS-Incoming includes traffic destined to the campus resolver by all external nodes—mostly local resolvers (ISP or open ones) operating to answer users' queries. Interestingly, the HTTPS fraction in DNS-Incoming is smaller than in DNS-Outgoing, as, for such a query to reach the campus resolver, the following two conditions must hold: (i) a client must issue an HTTPS DNS query and (ii) its local resolver must support HTTPS

DNS records. In the following, we show that not all local resolvers are already updated to support it.

Table 1. Distribution of DNS query types and responses.

	DNS-Outgoing		DNS-Incoming	
	Query	**Response**	**Query**	**Response**
A	38.67%	48.28%	53.34%	64.04%
AAAA	19.18%	5.69%	17.68%	0.1%
HTTPS	7.89%	0.89%	5.48%	0%
NS	29.7%	2.76%	3.03%	0.7%
Other	4.56%	42.38%	20.47%	35.16%

We now dissect HTTPS DNS adoption by infrastructure, thus showing the response rate to HTTPS queries in DNS-Outgoing separately by the DNS server's autonomous system. We use the MaxMind GeoLite database to map an IP address to the corresponding Autonomous System (https://dev.maxmind.com/). Recall that DNS-Outgoing includes traffic from the campus local resolver to the authoritative name servers for the domains being queried by the campus internal clients. Thus, in these DNS transactions, the server IP represents the authoritative DNS server for the queried domain, under the control of the corresponding organization. The top AS in terms of HTTPS queries is Google, which accounts for 27% of all HTTPS queries, followed by Amazon at 21%. Cloudflare ranks 5th with 6%, while other ASes, such as Apple and Fastly, each account for less than 0.9%. Interestingly, among the top 100 ASes in terms of HTTPS queries, only 26 responded at least once, meaning that most operators do not support HTTPS DNS records at all.

To quantify HTTPS deployment within operators, we measure the percentage of queries that are successfully answered. In Figure 2, we break down this number by the top 10 ASes in terms of HTTPS response ratio, among those receiving at least 10,000 HTTPS queries per day. Cloudflare leads with an approximately 48% response rate, meaning that 48 out of every 100 HTTPS queries are answered by the Cloudflare authoritative DNS server, while the rest go unanswered. This aligns with Cloudflare's unique role in supporting HTTPS DNS standardization and deployment as a means of offering ECH for hosted services [15]. Facebook ranks second, but the response rate is as low as 15%, followed by Google and Apple at 4.2% and 1.5%, respectively. The remaining ASes, such as Amazon and Akamai, exhibit response rates below 0.1%. Interestingly, some of the most frequently targeted ASes by HTTPS DNS queries—such as Amazon (21%), Microsoft (18%), and Akamai (9%)—have response rates below 0.1%. In some cases, as with Microsoft AS, no responses are observed at all despite millions of daily queries. In summary, clients support HTTPS DNS issue queries regardless of the target; however, only Cloudflare offers partial support, while other players answer those queries in only a minimal (or null) fraction.

In Figure 3, we further analyze HTTPS DNS responses in DNS-Outgoing in terms of the content they carry. Recall that such responses may contain a variety of information, including (i) the server IPv4 or IPv6 addresses (similar to A and AAAA responses), (ii) the Application Layer Protocol Negotiation (ALPN, i.e., the Layer 7 protocols supported by the server), and (iii) the public key to be used by the client to produce an ECH. Figure 3 shows the fraction of responses containing such information, separately for the top 5 ASes per response rate. We find that in most cases, the responses do not contain the server's IP address(es)—information essential for the client for establishing any connection to the required server. Thus, the clients need to rely on additional A or AAAA queries to obtain this information. An extreme case is that of Apple, whose responses are always empty, i.e., the DNS server returns a non-error HTTPS DNS response with no data. Only Fastly DNS servers always return a complete response, containing IP addresses and ALPN. Facebook

and Google responses, conversely, contain (almost) uniquely the ALPN, indicating that servers can be reached using the HTTP/3 and QUIC protocols. Focusing on the presence of the ECH public key, among the five ASes in the Figure 3, only Cloudflare includes it in the 16% of its responses. Over the entire DNS-Outgoing dataset, we find 8 ASes returning at least one response with a valid key. However, they are all related to Cloudflare domains, as we determined by decoding the base64-encoded key parameter (these keys are all associated with the ECH public name *cloudflare-ech.com*). This suggests that the rare occurrences of ECH keys in responses from non-Cloudflare ASes are likely the result of domains that are part of the Cloudflare infrastructure, but registered under different DNS zones. In summary, even when a client obtains a valid response to an HTTPS query, in most cases, it is incomplete, lacking the server IP address(es). This suggests that HTTPS DNS deployment is at an early stage, and any client willing to use it still needs to issue A and AAAA queries as well.

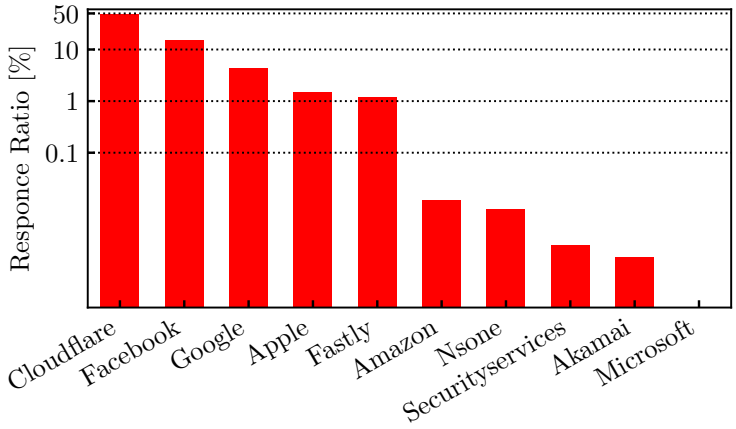


Figure 2. Response rate for the HTTPS DNS queries (DNS-Outgoing dataset).

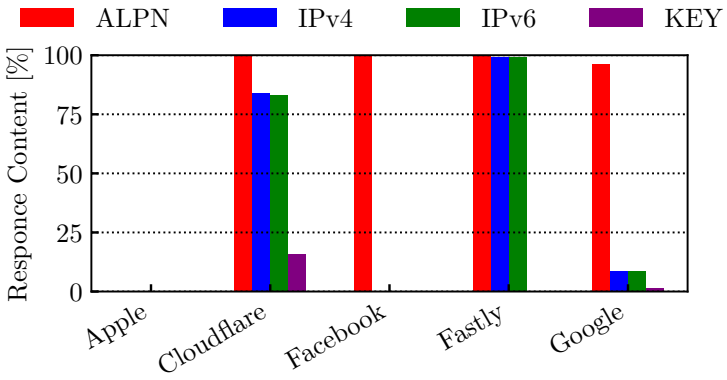


Figure 3. Breakdown of the HTTPS DNS response content (DNS-Outgoing dataset).

We now analyze the frequency of DNS HTTPS queries across resolvers from different ISPs, thus considering the DNS-Incoming dataset. Figure 4 presents the distribution of DNS query types among the top 10 ISPs in terms of packet volume. The predominant ISPs are Italian, which aligns with the fact that our organization is an Italian university serving students and staff primarily from Italy. An exception is *Free*, a French telecommunications provider affiliated with Iliad, which operates in Italy through its subsidiary Iliad Italia. Consequently, Iliad's DNS queries are resolved by Free's DNS infrastructure in Italy. Other exceptions are 1&1, a German ISP, and Renater, which provides internet access to French universities. Overall, all ISPs but 1&1 issue HTTPS DNS queries, meaning that the ISP

resolvers do support it as well as a portion of their clients (recall that in the DNS-Incoming we observe traffic from the local ISP resolver to the campus authoritative one). Vodafone and TIM show the highest proportion of HTTPS queries (21% each), which coincides with the lower percentage of AAAA requests—12% and 8%, respectively. This suggests a potential shift from AAAA requests toward HTTPS queries among users implementing HTTPS. Eolo and Tiscali follow with 15% and 13% of HTTPS requests, respectively. Non-Italian ISPs exhibit the lowest share of HTTPS requests, with 1&1, Free, and Renater at 0.3%, 2%, and 2.8%, respectively. Among Italian ISPs, Fastweb has the lowest percentage of HTTPS requests (4%), generating the second highest proportion of A-type queries (82%). GARR follows with only 8% of HTTPS requests. In short, most ISP resolvers already support HTTPS DNS and, even with strong differences, their users are starting to issue such queries.

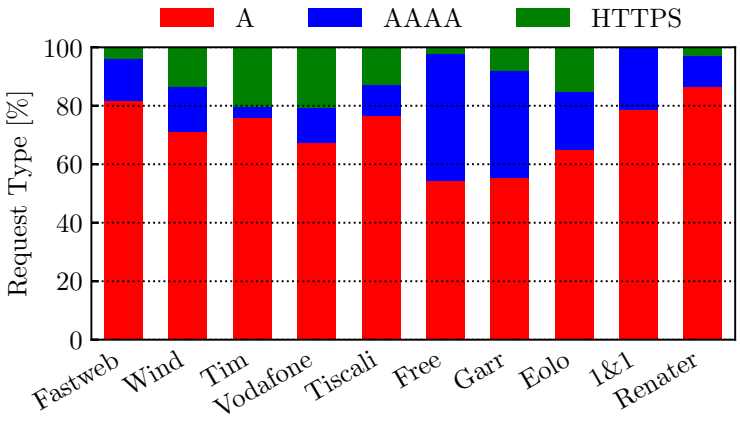


Figure 4. DNS request type from the top 10 ISPs (DNS-Incoming dataset).

4.2. HTTPS DNS and Query Volume

As Figure 3 shows, HTTPS DNS queries alone cannot be used as a replacement for A and AAAA queries, as they are typically incomplete. Thus, the clients supporting HTTPS DNS must issue two or three queries at the same time (of type HTTPS, A, and potentially AAAA) to accomplish a resolution. We now quantify this phenomenon by investigating how often an HTTPS DNS query is followed (or preceded) by another A or AAAA query for the same domain issues by the same IP address. Figure 5 quantifies these fractions in the DNS-Incoming dataset, considering windows of different durations. The x-axis represents the size of the time window in milliseconds, centered on the HTTPS DNS query, within which we search for other DNS queries. The y-axis shows the percentage of HTTPS requests associated with at least one subsequent DNS query. The results distinguish between queries of type A (blue line), AAAA (red line), and cases where both A and AAAA queries are observed (green line). The figure shows a growth in the occurrence of duplicate DNS queries as the time window increases. For a 1 ms window, 5.3% of HTTPS requests are followed/preceded by a repeated A query, rising to 18.8% at 10 ms and reaching 26.4% at 100 ms. AAAA duplicates are less frequent—starting from 1.6% at 1 ms and growing to 8.4% at 100 ms—consistent with the still-limited adoption of IPv6. Combined A and AAAA duplicates start at just 0.4% for a 1 ms window and reach 6.2% at 100 ms. Overall, A duplicates are approximately three times more frequent than AAAA across all windows. Notably, at the 100 ms mark, the share of combined A+AAAA duplicates (6.2%) nearly matches that of AAAA alone (8.4%), suggesting that clients issuing AAAA queries often also perform A queries—possibly due to dual-stack fallback strategies. We repeat the same analysis on the DNS-Outgoing dataset, while we do not report the figure for the sake of brevity. In DNS-Outgoing, we observe significantly higher proportions of duplicate A

queries—up to 46.2% within 100 ms, while AAAA duplicates remain extremely low, never exceeding 2.7% (The low presence of AAAA queries is explained by the lack of IPv6 support in the campus). The x-axis is truncated after 100 ms, as the Round Trip Time we measure between the passive monitor and the campus DNS servers is in the order of 2 ms; thus, we do not expect correlated queries to be captured with a temporal separation larger than 100 ms. Indeed, all three curves significantly flatten after 50 ms. We conclude that although HTTPS DNS queries are already a 5–10% fraction of DNS traffic, they cannot be used alone; thus, they are accompanied by A or AAAA queries to the same domain, increasing DNS traffic volume and imposing extra load on DNS servers.

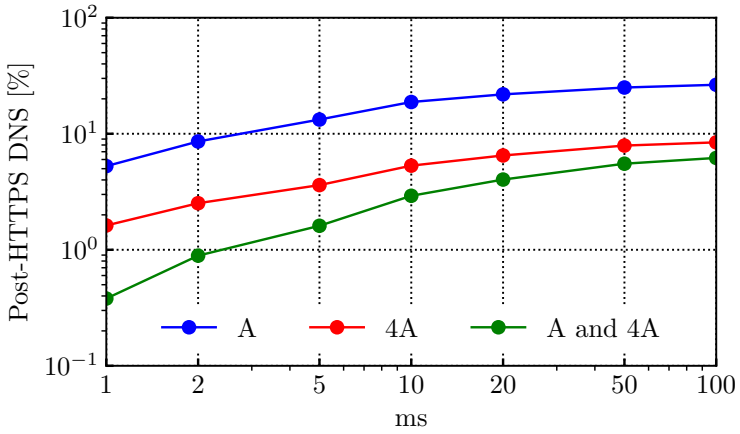


Figure 5. Percentage of HTTPS DNS requests followed or preceded by A and/or AAAA queries for the same domain by the same client (DNS-Incoming dataset).

4.3. Encrypted Client Hello in QUIC Flows

We now focus on the Encrypted Client Hello feature and study its adoption using the QUIC-Outgoing dataset. Specifically, for each QUIC connection in the data, the Tstat passive meter seeks the ECH extension in the "Initial" packet of each QUIC connection and records its in-clear fields. Among all QUIC flows, 59% carry an ECH extension, which is quite a surprising proportion. Not only do most clients support ECH, but they also include an ECH extension in most of their QUIC initial packets. Note that the ECH standard mandates clients to include an ineffective ECH, even when the server does not support it—called GREASE ECH (see [20]). Its goal is to avoid protocol ossification and traffic differentiation by middleboxes. In the following, we investigate the features of ECH extensions and the frequency of genuine and GREASE ones. Indeed, despite QUIC's full encryption since the initial packet, the structure and size of the ECH extension in the Client Hello remain accessible for passive observation. This makes it possible to distinguish between genuine ECH and GREASE ECH. As detailed in Section 2, ECH Client Hello packets include an outer, in-clear SNI called "ECH Public Name" that acts as a placeholder. Cloudflare, the only current adopter of ECH, publicly states that flows directed to its servers use the cloudflare-ech.com ECH Public Name (https://developers.cloudflare.com/ssl/edge-certificates/ech/, accessed on 23 July 2025).

To study the use of ECH in real-world traffic, Figure 6 shows the distribution of ECH extension size in QUIC flows, focusing on the top 10 ASes by traffic volume. We identify 6 dominant extension sizes, each corresponding to a GREASE value (in decimal format). These 6 values are consistently used across all ASes, except for Cloudflare, which additionally employs value 154, which is, however, still one of the 16 GREASE values recommended by the standard. Indeed, recall that the only AS (partially) supporting ECH is Cloudflare, while all ECH extensions in flows to other ASes are certainly GREASE.

Less frequent extension sizes are grouped into the "Other" category in black. This group includes the subset of actual valid ECH exchanges, which account for 1.5% of all ECH flows and are uniquely directed to the ECH Public Name cloudflare-ech.com, exclusively served by the Cloudflare AS. Overall, although most QUIC flows include an ECH, only a fraction (1.96%) of those directed to Cloudflare do include a valid one, while all the other are GREASE ones.

To conclude our analysis of the ECH extension, in Figure 7, we present the cumulative distribution of the size (in bytes) of the QUIC ECH extensions. The x-axis indicates the size of the ECH extension, while the y-axis shows the cumulative distribution function (CDF). The solid red line represents flows carrying a valid ECH extension, which we identify from the outer SNI, which reports the domain cloudflare-ech.com, whereas the dashed blue line refers to flows with a GREASE ECH extension. We observe a clear separation between the two distributions. Flows with actual ECH usage have sizes tightly concentrated between 150 and 180 bits. In contrast, flows without ECH exhibit a much broader and more irregular distribution, ranging from values below 50 bits up to 250 bits. Thus, we observe how the size of the GREASE ECH does not resemble that of the real ones; for instance, many real ECHs have a size of 154 Bytes, while this value never appears for GREASE ECHs. This entails that, although GREASE ECH can help prevent protocol ossification, middleboxes may still leverage the extension size to distinguish between real and GREASE ECH.

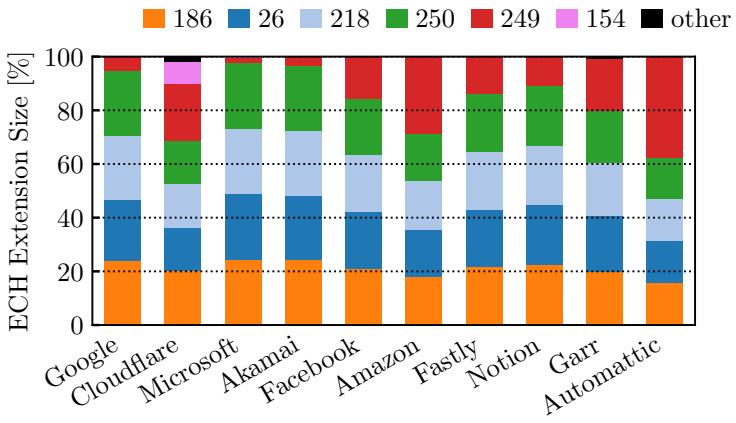


Figure 6. GREASE ECH values in the QUIC connections (QUIC-Outgoing dataset).

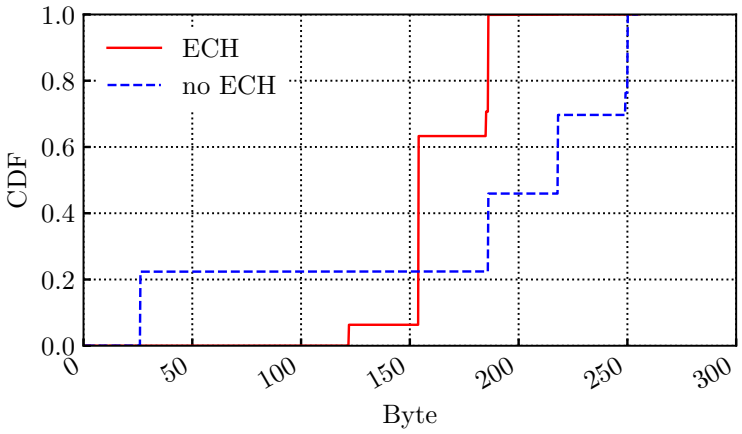


Figure 7. ECH payload dimension (QUIC-Outgoing dataset).

5. Conclusions

This paper presented the first study of Encrypted Client Hello and HTTPS DNS records using passive measurements. Although these are recent standards (ECH is still an IETF draft by the TLS Working Group), we observe that many client devices already use such features, likely due to popular browsers' support for these privacy-enhancing technologies. Specifically, HTTPS DNS represents 5–10% of all DNS queries, but fewer than 5% of those queries receive a valid response. Not all providers support HTTPS DNS; only Cloudflare and Fastly provide complete and correct answers. ECH deployment is at a later stage from the clients' perspective; 56% of QUIC flows include an ECH, meaning the respective clients already support the standard. Interestingly, adoption on the server side is almost nonexistent. Only Cloudflare partially supports ECH; thus, in almost all cases, the observed ECH is "GREASE", meaning it does not include any encrypted domain name but rather verifies possible traffic differentiation at middleboxes.

As ECH and HTTPS DNS continue to evolve, further research is necessary to fully understand their implications across different segments of the internet ecosystem. It remains uncertain whether major internet providers and platforms will adopt these technologies at scale, given the non-trivial operational and infrastructural costs they entail. Continuous passive monitoring is, therefore, essential to track real-world deployment and usage trends. Additionally, the computational impact of ECH and HTTPS DNS on both clients and servers—particularly in terms of handshake processing and DNS resolution—has yet to be thoroughly evaluated and represents a promising direction for future work. These technologies also shift the balance between privacy and network observability, potentially conflicting with the security and policy enforcement requirements in corporate networks. As a result, their deployment may face resistance or outright blocking in environments relying on firewalls, intrusion detection systems, or parental control middleboxes, raising further questions about compatibility, detectability, and circumvention.

Author Contributions: Conceptualization, M.T. and D.G.; methodology, M.T. and G.M.; investigation, G.M.; resources, M.T.; data curation, M.T. and G.M.; writing—original draft preparation, M.T. and G.M.; writing—review and editing, M.T. and G.M.; supervision, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results was funded by projects 20228FT78M "DREAM" and 2022M2Z728 "COMPACT" under the Italian Ministry of University and Research 2022 PRIN program. This work received funding from Next Generation EU, Mission 4 Component 1, CUP: D53D23001340006.

Data Availability Statement: The datasets presented in this article are not publicly available because of privacy or ethical restrictions. Requests for access to the datasets should be directed to the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Farrell, S.; Tschofenig, H. Pervasive Monitoring Is an Attack. RFC 7258. 2014. Available online: https://www.rfc-editor.org/rfc/rfc7258.html (accessed on 4 August 2025).
2. Naylor, D.; Finamore, A.; Leontiadis, I.; Grunenberger, Y.; Mellia, M.; Munafò, M.; Papagiannaki, K.; Steenkiste, P. The cost of the" s" in https. In Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies, Sydney, Australia, 2–5 December 2014; pp. 133–140.
3. Iyengar, J.; Thomson, M. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000. 2021. Available online: https://www.rfc-editor.org/info/rfc9000 (accessed on 4 August 2025).
4. Hu, Z.; Zhu, L.; Heidemann, J.; Mankin, A.; Wessels, D.; Hoffman, P.E. Specification for DNS over Transport Layer Security (TLS). RFC 7858. 2016. Available online: https://www.rfc-editor.org/info/rfc7858 (accessed on 4 August 2025).
5. Hoffman, P.E.; McManus, P. DNS Queries over HTTPS (DoH). RFC 8484. 2018. Available online: https://www.rfc-editor.org/info/rfc8484 (accessed on 4 August 2025).

6. Casanova, L.F.G.; Lin, P.C. Generalized classification of DNS over HTTPS traffic with deep learning. In Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, 14–17 December 2021; pp. 1903–1907.
7. Vekshin, D.; Hynek, K.; Cejka, T. Doh insight: Detecting dns over https by machine learning. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Virtual, 25–28 August 2020; pp. 1–8.
8. Rescorla, E.; Oku, K.; Sullivan, N.; Wood, C.A. TLS Encrypted Client Hello. Internet-Draft draft-ietf-tls-esni-24, Internet Engineering Task Force. Work in Progress. 2025. Available online: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/24/ (accessed on 4 August 2025).
9. Schwartz, B.M.; Bishop, M.; Nygren, E. Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records). RFC 9460. 2023. Available online: https://www.rfc-editor.org/info/rfc9460 (accessed on 4 August 2025).
10. Trevisan, M.; Soro, F.; Mellia, M.; Drago, I.; Morla, R. Does domain name encryption increase users' privacy? *ACM SIGCOMM Comput. Commun. Rev.* **2020**, *50*, 16–22. [CrossRef]
11. Trevisan, M.; Soro, F.; Mellia, M.; Drago, I.; Morla, R. Attacking DoH and ECH: Does Server Name Encryption Protect Users' Privacy? *ACM Trans. Internet Technol.* **2023**, *23*, 19. [CrossRef]
12. Shamsimukhametov, D.; Kurapov, A.; Liubogoshchev, M.; Khorov, E. Is encrypted clienthello a challenge for traffic classification? *IEEE Access* **2022**, *10*, 77883–77897. [CrossRef]
13. Bhargavan, K.; Cheval, V.; Wood, C. 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, Los Angeles, CA, USA, 7–11 November 2022; pp. 365–379.
14. Niere, N.; Lange, F.; Heitmann, N.; Somorovsky, J. Encrypted Client Hello (ECH) in Censorship Circumvention. *Free. Open Commun. Internet* **2025**, *2*, 64–73.
15. Tsiatsikas, Z.; Karopoulos, G.; Kambourakis, G. Measuring the adoption of TLS encrypted client hello extension and its forebear in the wild. In *European Symposium on Research in Computer Security*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 177–190.
16. Zirngibl, J.; Sattler, P.; Carle, G. A first look at SVCB and HTTPS DNS resource records in the wild. In Proceedings of the 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Delft, The Netherlands, 3–7 July 2023; pp. 470–474.
17. Dong, H.; Zhang, Y.; Lee, H.; Huque, S.; Sun, Y. Exploring the Ecosystem of DNS HTTPS Resource Records: An End-to-End Perspective. In Proceedings of the 2024 ACM on Internet Measurement Conference, Madrid, Spain, 4–6 November 2024; pp. 423–440.
18. Trevisan, M.; Finamore, A.; Mellia, M.; Munafo, M.; Rossi, D. Traffic analysis with off-the-shelf hardware: Challenges and lessons learned. *IEEE Commun. Mag.* **2017**, *55*, 163–169. [CrossRef]
19. Trevisan, M. Dpmon: A Differentially-Private Query Engine for Passive Measurements. Preprint. Available online: https://ssrn.com/abstract=4901748 (accessed on 4 August 2025).
20. Benjamin, D. Applying Generate Random Extensions And Sustain Extensibility (GREASE) to TLS Extensibility. RFC 8701. 2020. Available online: https://www.rfc-editor.org/info/rfc8701 (accessed on 4 August 2025).
21. Fan, J.; Xu, J.; Ammar, M.H. Crypto-pan: Cryptography-based prefix-preserving anonymization. *Comput. Netw.* **2004**, *46*, 253–272. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.