Remaining potential weaknesses in ECH (Encrypted Client Hello)

The consequences of the adoption of ECH and its related protocols

L. Aaron Kaplan, Sebastian Wagner

2025-06-30

Contents

1 About this report

The following security analysis tries to identify weaknesses in the ECH protocol (especially as it also interfaces with lots of other protocols). It is part of the DEfO project - Developing ECH for OpenSSL[1].

In the DEfO project, task 9.1 deals with:

> Deployment Scenarios Analysis: there are many variations in how ECH can be deployed and the varying relationships between the client and server entities involved. There therefore remains a need to map out residual privacy leaks in such scenarios and how to plug those, given the existence of additional privacy mechanisms such as Qname Minimization, Oblivious DNS-over-HTTPS, and MASQUE.
>
> This task will accumulate practical documentation (for deployers) and analyses covering these issues. It is clear ECH reduces metadata leakage, but it is not yet clear how censors might react. This task will audit and review the metadata in a properly functioning ECH interaction and explore remaining avenues for de-anonymisation, filtering, blocking and censorship. This touches relevant protocols required in an ECH setup, such as: DNS-over-HTTPS (DoH), DNS-over-TLS (DoT), OCSP (Online Certificate Status Protocol), certificate revocation list (CRLs), etc. We will also focus on "medium scale" web sites such as found on a University campus, smaller hosters or NREN member organisation – CDNs already see the benefits of ECH and have a clear path to deployment, but it is important to enable smaller, but still significant, scale organizations to enjoy the benefits of ECH. The task will result in one report.

2 Methodology

The authors read the RFCs and -Drafts and brainstormed after each how we might break or downgrade the security/privacy promises. In addition, we gave a presentation on the topic in front of approx. 30 IT security professionals in order to elicit feedback and further ideas on how to attack the protocols. The result is this report.

3 How to read this report

This report intentionally takes a skeptical perspective. Not in order to criticize ECH or the approach but rather to challenge the protocol authors, implementers and deployments to think about certain aspects we mention in the report. This "skeptical"/outside view helps to ask hard questions and identify potential issues which might have been overlooked.

[1] https://defo.ie/

By releasing this report we aim to improve future versions.

It is clear that - given such widely used code and protocol stacks as with HTTP/HTTPS, changing things is very hard. ECH tries to achieve the maximum possible, given lots of constraints by the protocol landscape, implementers, etc. Hence, ECH has to live with all the legacy issues. It's probably not possible to find a quick, elegant and 100% compatible solution for the problem which ECH is trying to address. ECH being a complex solution is inherent - RFC8744 provides lots of background for these matters.

Finally, we acknowledge that ECH is an incremental update step and we assume there will be an incremental roll-out of ECH globally. This has multiple implications: 1) every deployment will protect clients a bit more ("herd-immunity" principle) 2) weaknesses in ECH highlighted in the report do not necessarily affect all deployments (there will be variations) 3) ECH will grow over time and add privacy over time. It's not a binary turning privacy on or off situation.

ECH will be measured by its provisioning of privacy after all.

With these things being said, the report will skeptically look at ECH.

4 Overall conclusions and summary of the report

While ECH provides privacy improvements over cleartext SNI, the protocol also introduces complexity and new attack vectors primarily centered around DNS manipulation and complex deployment scenarios. The greatest risks arise from the protocol's dependency on DNS infrastructure and as well on the potential for downgrade attacks. Successful ECH deployments require close attention to

- operational security,
- comprehensive monitoring,
- and a secure DNS (signed zones, detectability of tampering with DNS answers, etc.)

From the clients' perspective, the main question is:

- which DoH/DoT recursive servers can one use? Are they even reachable from within the country of the client? Or are they blocked, re-directed or manipulated? Untrusted DoH/DoT recursive servers pose a security as well as a privacy problem. Depending on the trust the client may place into the recursive DNS server, it might get correct or incorrect answers and hence be able to reach ECH enabled servers or a mere look-alike middle-man server. It's debatable if this is an issue that ECH can address. Probably not. However, we believe it's worth noting.
- Can the client detect downgrade attacks and how does it deal with them? The RFC standard is a big vague here (SHOULD vs. MUST) ("…the client SHOULD abandon its attempt to reach the

service"[2]). We recommend looking at this issue when implementing ECH in TLS libraries.

In addition, the authors of this report see two major concerns with ECH deployments for the internet as a whole:

1. ECH depends on multiple protocols and specifications to work together seamlessly. The protocol's fundamental reliance on DNS creates a large attack surface that inherits many of DNS's security issues. In addition to DNS, quite a few other protocols are involved. This creates an even bigger attack surface. The interfaces between protocols are usually interesting targets for attackers. We recommend that the authors of the RFC drafts closely take a look at these edge-case vulnerabilities. The report highlights some of those such as the WKECH interface between DNS zones and CFS/backend servers.
2. ECH leads to more centralization of the internet: the CDNs will profit. More centralization actually leads to more data being in the hands of only one or just a handful of organisations. Therefore, the risk of de-anonymization by combining and correlating different data sets (ECH configs, DNS recursor query logs (such as passive DNS[3], DNS query traces), Certificate transparency logs (CTLs)[4]) actually *increases*: any organisation combining these data sets could most probably completely de-anonymize the traffic. The more of these datasets are in the hand of fewer organisations, the higher the risk these datasets can be combined.
3. The complexity of the protocol and the setup costs (in terms of manpower) favor the large players. Smaller players might make mistakes in their setups.
4. Small and medium size organisations often don't host that many services on one IP address anyway. See "The web is still small after more than a decade[5]", Nguyen Phong Hoang, et. al. Therefore the anonymity data set is usually small. Guessing and inferring which hostname a client connected to is trivial in the case that one IP address only hosts one service with one hostname.
5. Smaller and medium sized organisations won't necessarily profit from protecting the clients' privacy. Exceptions are listed in incentives. So why should they bother deploying ECH (and risking downtime)? We have a game-theoretic problem here. Which in turn will only lead to more centralization (see 3.) which will lead to 2.

Nevertheless, apart from these overall considerations, we recommend that organizations wishing to deploy ECH should prioritize

- security controls around DNS infrastructure,
- implement robust monitoring for ECH-related attacks,
- and prepare for the operational complexity introduced by the protocol's multi-layer dependencies.

[2] https://www.rfc-editor.org/rfc/rfc9460.html#name-handling-resolution-failure
[3] https://www.ietf.org/archive/id/draft-dulaunoy-dnsop-passive-dns-cof-12.html
[4] https://datatracker.ietf.org/doc/rfc6962/
[5] https://www.researchgate.net/publication/341627684_The_web_is_still_small_after_more_than_a_decade

5 Scope

This report mainly focuses on

- problems with deploying ECH and how to adress these,
- sources of misconfigurations
- implications of deploying ECH (assessing subsequent risks)
- attacks against ECH enabled servers
- possibilities for censorship or denial of service
- the interface layers between protocols which play together to make ECH possible

5.1 None-scope

The report does *NOT* try to

- analyse if the used cryptographic libraries, primitives or algorithms are safe (or quantum resistant)
- prove the security of individual implementations (or the lack of, by breaking or analysing concrete implementations)

6 Intended audience

This report is for:

- technical persons who want to deploy ECH and need to understand limitations and weaknesses of the protocol
- the protocol authors and implementers as inpiration for future revisions of the protocol

7 Deployment considerations

This section explores ECH deployment considerations. Relevant links to additional sections will be provided, detailing potential attacks against the protocols.

7.1 Process Overview

The following is a streamlined overview of the workflow involved when a browser accesses an ECH-protected website. The blue arrows signify the client-side process, while the green ones are the server-side process.

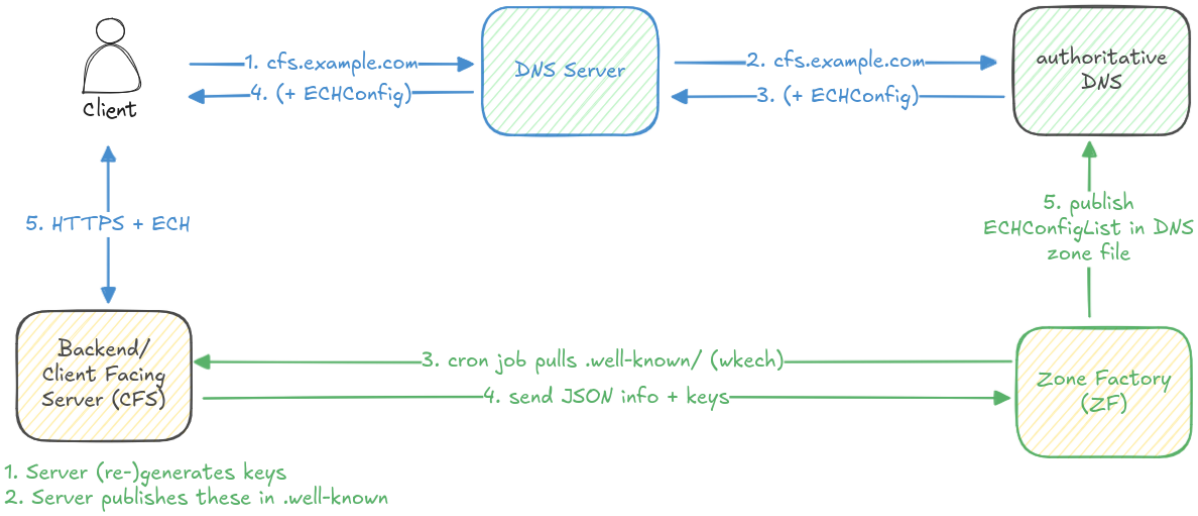

Figure 1: WKECH flow

7.1.1 Client-side Process

1. To initiate a website request, the browser first queries the A/AAAA records and the ECHConfig from the configured DNS server. This server may be a recursive DNS server provided by the network operator or a central DoH server.
2. The designated DNS server then queries the authoritative DNS server for the required information, which is managed by the website operator.
3. Once retrieved, the information is relayed from the authoritative DNS server to the DNS server, potentially being cached for future requests by this or other clients.
4. The DNS server subsequently transmits the information to the client.
5. Utilizing the A/AAAA records and the ECHConfig, the browser sends an HTTP request to the web server to fetch the website.

Typically, recursive DNS servers communicate with authoritative DNS servers using traditional unencrypted UDP-based DNS (Do53). Nonetheless, the adoption of DoT and DoH protocols is on the rise. Additionally, various protocol upgrades (either opportunistic or through SVCB records) are possible.

7.1.2 Server-side Process

1. The server regularly regenerates the ECH keys at defined intervals (for example, every hour) for each configured domain.
2. The server publishes the corresponding public ECH keys within the WKECH directories for every domain.

3. The Zone Factory (ZF) requests the ECH keys for each designated domain at pre-established intervals (preferably more frequent than once per hour).
4. The Client-Facing Server (CFS) responds with the requested ECH keys.
5. The ZF subsequently pushes the generated ECHConfig to the DNS server.

7.2 Webserver configuration

On the webserver side, several considerations must be addressed:

- Which component generates the ECH keys with the appropriate parameters?
- Which entity handles the rotation of these keys and reloads the web server configuration?
- What component creates (or services) the WKECH directory, ensuring only public keys are exposed and private keys remain secure?
- How is the ZF triggered after each key rotation, ideally operating separately on a different host? (see Separation).

There are similarities between the ACME protocols (made popular by the Letsencrypt initiative) and ECH, as both generate keys on the webserver and write information to the DNS zone. ECH adds the additional WKECH, see also WKECH for possible problems related to it.

To facilitate the ECH deployment, straightforward and easy tools, covering these processes, akin to ACME clients or Apache's mod_md[6] need to be developed.

Guidance on setting up webservers with ECH, can be found in the ECH Development utilities[7].

7.3 Complexity of Configuring the Zone Factory

The Zone Factory must be aware of the following:

1. Identifying well-known sites (`wkech`) to monitor.
2. Establishing a refresh schedule for the keys (either on a fixed interval or responsive to activity).
3. Knowing which zone files on which servers require updates.

The ZF requires write access to the zone files and must have the capability to reload the nameserver configuration. This setup is non-trivial for a systems administrator, as misconfigurations or oversights can introduce complications.

It is imperative to secure the WKECH directory: it must contain only public keys, be immutable (including to any aliases), and limit access solely to the web server itself. For more information, please refer to the section on WKECH.

[6] https://httpd.apache.org/docs/2.4/mod/mod_md.html
[7] https://github.com/defo-project/ech-dev-utils#user-content-server-details

7.4 DNSSEC implementation

DNSSEC (Domain Name System Security Extensions) implementation enables clients to validate ECH-enabled domains. This not only enhances the integrity of the DNS responses but also mitigates the risk of resolvers inadvertently blocking SVCB or ECH parameters.

7.5 Deployment Incentives

As mentioned in the overview section, we see a game-theoretic problem: most organizations that host web services might not have the proper incentives to protect the client's privacy without additional rewards: they have no incentive to do so. Instead, managing the complex ECH setup adds additional business continuity risks.

Therefore, this section looks at organizations that could be interested in deploying ECH.

The widespread **adoption** of ECH or ECH GREASE **is crucial** for the success of ECH. Without it, the use of ECH itself may raise suspicions among censors. Current versions of Firefox and Chrome implement ECH along with ECH GREASE mode, resulting in substantial portions of TLS traffic containing ECH.

7.6 Anti-Censor and Anti-Oppression

Various organizations that address humans in countries and regions with oppression and legal pressure, like human rights organisations have an interest in protecting their website visitors' identity and thus using ECH as well as protecting their website behind CDNs to reduce the risk of de-anonymization by traffic correlation analysis.

In all regions, the same applies to whistle-blower platforms which are possible under close observation by political, legal or corporate organizations.

7.7 Malware - C2 operators

Unencrypted SNI/Client Hello and TLS Metadata (cipher suite lists, advertised extensions) are being used to identify malware-generated traffic.

Therefore operators of malware networks have an interest in protecting their traffic and thus implementing ECH. Consequently, this will hinder - but not disable - this traffic-analysis. At the moment a similar effect could be gained by using different or random hosts in SNI, but that itself will form a pattern and and thus has limited effect.

For example, see: Detecting Encrypted Malware Traffic (Without Decryption)[8] for a study on traffic classification not using SNI.

Currently, the usage of ECH is very low and thus in itself suspicious. To hide their ECH traffic, malware operators may be inclined to increase the general usage of ECH.

7.8 Pornography industry

Starting with February 2019, South Korea started blocking TLS-encrypted traffic to sites forbidden by the policies, most prominently pornography.

The porn industry, being blocked, has therefore a commercial interest in using ECH, which allows them to reach customers in an entire 50-million-inhabitant country.

- South Korea to Extend Site Blocking by Snooping on SNI - technadu.com, August 1st, 2021[9]
- South Korea is Censoring the Internet by Snooping on SNI Traffic - Bleepingcomputer.com, February 13th, 2019[10]

7.9 CDNs

CDNs can offer their customers ECH (shared mode) as part of their offerings, selling them a privacy feature for low internal costs as they profit highly from scaling effects.

For website owners who want to protect their visitors and offer them enhanced privacy, it is much easier and cheaper to go to CDNs than to operate ECH on their own, given the high complexity and costs. An equivalent effect is in progress for about 15 years in the email sector. As operating email services safely got increasingly complex and challenging, more and more organizations outsourced their email infrastructure to large email service providers.

For ECH's split mode the business incentive is lower than for shared mode, as in split mode the CDNs can sell fewer services to their customers. Currently there are no known offers for split-mode ECH.

7.10 Small and Medium-Sized Hosting Providers and research and education institutions (NRENs)

As mapped out in the section Deployment considerations, the required implementation effort is quite high and only scales for lots of websites.

[8] https://blogs.cisco.com/security/detecting-encrypted-malware-traffic-without-decryption
[9] https://www.technadu.com/south-korea-extend-site-blocking-snooping-sni/58125/
[10] https://www.bleepingcomputer.com/news/security/south-korea-is-censoring-the-internet-by-snooping-on-sni-traffic/

Additionally, NRENs typically have a very decentralized infrastructure, so scaling effects won't take effect and Separation of responsibility cause additional barriers.

A group of security experts and IT service operators that we inquired about was very hesitant about the potential deployment of ECH, citing the significant effort and minor benefits.

7.11 Separation issues

7.11.1 Network separations

The synchronization between the CFS server and the ZF can be achieved through two different methods: pulling and pushing.

Pushing: The CFS server triggers the ZF after generating new ECH keys. For security reasons, the CFS server and the DNS server may be located on separate networks, with firewall rules preventing any outgoing connections from the CFS to internal networks. Additionally, the DNS server can block incoming connections from the CFS network range.

Pulling: In this method, the ZF retrieves the keys from the CFS server using WKECH. The ZF must be made aware of when to refresh the keys for this approach. This can be done through high-frequency polling at the ZF.

Both methods have their drawbacks, and it is up to the operator to evaluate the limitations and choose the most suitable approach.

7.11.2 Process separations

In large organisations we often observe that different teams are responsible for different tasks. ECH deployments require that at least the team for DNS (ZF) and the team responsible for hosting services talk to each other. We are not commenting on how often this might be a problem.

In addition, some organisations have to deal with IT security regulation standards (ISO certifications, laws such as NIS2), that might impact the way how ECH is being deployed (which team does what, documents what, etc.). Since deployments of ECH pose new questions, we can't yet fully assess the impact of regulations on deployments. However, it is worth keeping an eye on so that future revisions of the ECH drafts can be deployed easier for organisations facing tight regulations.

We have no information how often this may be hindering deployment.

7.11.3 Organizational separations

In the same way as a lot of IT services are outsources (web, e-mail, etc.) DNS Servers may be operated by third parties, e.g. a registrar, a DDoS protector etc. Thus, there is no direct access via SSH to the server administration, but rather via websites and APIs.

The let's encrypt project had the same challenge when starting with the ACME *dns-01* challenge. As a result, a lot of ACME clients (go-acme[11], posh[12], acme.sh[13] etc.) implemented support for a lot of provider's APIs. The same effort will become necessary for the Zone Factory software. Potentially existing code from ACME-projects could be re-used to reduce the expenses.

8 Overall concerns with complexity

The authors of this report want to emphasize that for solving the particular problem which ECH tries to solve, we don't currently see an easier solution. Hence, the ECH protocol achieves its goal. However, this comes at the cost of added complexity. This complexity might come to haunt us later on.

Higher complexity leads to:

- issues and mistakes in configuring it properly -> things go wrong easily
- weaknesses between protocol layers: with many protocols interacting with each other, the interface between protocols is often the weak spot
- potential design omissions: there might be more hidden oversights in the design of the complex interplay between the protocols

The authors of this report acknowledge that solving the issue is not an easy task and no easier methods are known. ECH as a protocol does its job at solving a particular issue. But at the cost of complexity.

9 Downgrade attacks

If a network operator does not want to allow the typical DoH servers, he/she essentially only has the option to block those. However, since clients will still try to use DNS over classical means then, the attacker has the possibility to promote his/her tampered recursive DNS server. Essentially this results in a downgrade attack on ECH as well, since DNS is a precondition to establishing a session with an ECH server.

[11] https://go-acme.github.io/lego/dns/index.html
[12] https://poshac.me/docs/v4/Plugins/
[13] https://github.com/acmesh-official/acme.sh/wiki/dnsapi

Of course, when the setup is working, the incremental security upgrade that ECH gives (according to what it was intended to do), is not to be dismissed.

10 Mandated use of DoH

On the other hand, if we require DoH/DoT, we essentially play into the hands of those who centralize recursive DNS servers. See the arguments on centralization in the introduction.

11 DNSSEC is de-facto mandatory

While DNS is complex, adding DNSSEC makes it even more complex. However, to protect against wrong DNS answers by byzantine recursors, DNSSEC will be de-facto mandatory for a secure ECH protocol.

- The web server **MUST** ensure that `.well-known/origin-svcb` is well protected.
- The web server **MUST** ensure that `.well-known/origin-svcb` does not contain any private keys.

11.1 Attacks against syncing the ZF with the CFS / backend

If we look at the (at the time of writing this report) most current version of the WKECH Draft[14], then we can identify a couple of weaknesses in the interplay between WKECH and ECH:

1. The synchronization between the ZF and the CFS is done strictly secured via HTTPS.
2. If an attacker manages to skew the timing info (for example by controlling the clock), this could result in invalid times for the `regeninterval`. However, it should be noted that if an attacker has these capabilities, targets other than ECH key regeneration intervals may have bigger effects.

11.1.1 Attacks against syncing the ZF and CFS

Assuming a state-sponsored attacker, we can assume this attacker has access to a CA and may issue arbitrary certificates. Since the connection between the ZF and CFS (according to the draft) MUST go over HTTPS, we could perform a Man-In-The-Middle attack[15] on this HTTPS conversation. Currently the standard does not recommend certificate pinning, CA pinning (CAA)[16] or similar techniques to counter this.

[14] https://datatracker.ietf.org/doc/html/draft-ietf-tls-wkech-07
[15] https://en.wikipedia.org/wiki/Man-in-the-middle_attack
[16] https://en.wikipedia.org/wiki/DNS_Certification_Authority_Authorization https://github.com/sftcd/wkesni/issues/44

Following the protocol specs in draft-ietf-tls-wkech-07, we can read *"An empty endpoints array means that all HTTPS records that the ZF has published for the origin should be deleted"*. This would invite a MiTM to delete all ECHConfig zone file entries for the given domains. Effectively forcing the clients to downgrade to pre-ECH connection mechanisms.

Recommended mitigation strategy: the draft authors might consider mandating certificate pinning or similar techniques.

12 Browsers and ECH

Modern web browsers are notably permissive toward emerging standards, often prioritizing user functionality over enforcing new security features if the potential impact is too disruptive. With their quick adoption of new technologies and fast release cycles, they effectively serve as experimental platforms for evaluating and implementing new protocols in real-world environments.

The browser Firefox adopted DoH as their default setting, reverting back to Do53 should a DoH connection fail to establish. The Browsers Chrome and Edge use DoH if the system's default resolver supports it. Opera, Brave and Vivaldi do not use DoH by default.

DoH connection failures can arise from active downgrade attacks, where malicious entities intercept and manipulate traffic. Consequently, the usage of ECH can be silently thwarted if an attacker holds sway the network path between the user and the intended DoH server or between recursive and authoritative DNS server. An attacker with control over the network connection can though also block TLS and other security measures, but not without alarm bells going off in the browser and other clients.

For the implementation of ECH, attention must not only be paid to pure HTTPS traffic but also to other communication channels such as WebRTC and network proxies, as neglecting ECH on these channels can introduce ways for de-anonymization.

12.1 Browsers' Policy Enforcement Power

In the past, Browsers and the CA/Browser forum have repeatedly shown that they can enforce new policies towards network and website operators, pushing them to fast adjustments to not risk their website's reputations, such as:

- Starting in 2015, Browsers gradually marked unencrypted HTTP connections as unsafe, pushing all website operators to use HTTPS.

- Previous to 2011, certificate lifetimes of up to 10 years were allowed by standards and accepted by browsers. Since 2020, a year is the maximum allowed lifetime. Discussions on further reductions are ongoing.
- Other examples include: Deprecation of SHA-1 (2014-2017) and Deprecation of RSA Keys (since 2010), Distrust of CAs (Symantec, 2017–2018), Mixed Content Blocking, Requirements to CAs[17] such as Certificate Transparency, OCSP, `SubjectAltName`, Domain Validation Methods, Deprecation of Certificates for Internal Server Names (2011-2016), Deprecation of TLS protocols and cipher suites

The reasons for this effectiveness lie in the huge market share of a small number of browsers and, on the operators' side, the high reputation risk: Website operators simply cannot afford security warnings or site breakage.

These policies are increasingly getting tighter, pushing the operators to more automation and more complex environments. This, in turn makes the operation less efficient pushing many of them to centralized service operators. This counteracts the original intention of a decentralized Internet.

Therefore, we emphasize that Browsers must not use its power to enforce ECH because they push users to DoH.

12.2 DoH server oligarchy

If communication between client and server only uses ECH to hide the destination server's name, but does not use encrypted DNS, intermediaries are still able to eavesdrop the destination by observing the Name Resolution traffic. Therefore ECH relies on the usage of DoH, DoT (DNS over TLS) or DoQ (DNS over QUIC) to hide the server's name entirely.

Of the three encrypted DNS protocols, DoH is the most used, therefore we focus here on this one. As DNS over QUIC isn't yet widely used on the client's side, we cannot yet foresee the consequences. And DoT is intended to be used as decentralized as Do53.

Firefox and Edge use Coudflare's DoH server, and Chrome uses Google's DoH server. Just two DoH servers provide DoH services to the majority of browser users. Although users could change the DoH server setting, only a fraction will do that or even understand what DoH is. This imbalance implies various problem areas:

- Privacy: Despite their emphasis that they won't save the query data, these policies can change at any time. Users are locked into a trust dependency without opt-in.

[17] https://cabforum.org/baseline-requirements/

- Jurisdiction and Geopolitics: The DoH servers are operated by companies in a single country. Their legal system can force them to share the query data at any time, impacting users **worldwide** without any possibility for them to notice.
- Market dominance: Two companies control the majority of all DoH traffic. This creates a huge market dominance. In the future, this could lead to them effectively taking over the role of Domain registrars.

Firefox used to require DoH for ECH in Firefox 119[18], but stopped doing so in Firefox 129[19]. As DoH is required for proper ECH, these problems are worsened by ECH. ECH aims to defend users' privacy, but its reliance on DoH may thwart this goal.

The Tor is reluctant on the usage of DoH for the same reason[20].

12.3 OCSP and CRL

Online Certificate Status Protocol (OCSP) and Certificate Revocation Lists (CRL) are core components in the validation of digital WebPKI certificates. The data is typically transmitted over HTTP but verified through signing by the Certificate Authority (CA). Access to these lists can be obstructed through packet inspection due to the unencrypted nature of the traffic. Browsers generally employ a soft-fail approach for CRL validations by default, meaning the information about a revoked certificate may not reach the client. If an adversary has the capability to interfere at this stage, they may effectively disrupt other, more important connections as well.

OCSP has recently been called into question and may be nearing obsolescence[21]. As a result, the CRLs' importance increases again.

This study found no relevant interplay of ECH on CRL and OCSP mechanisms.

13 Libraries and Devices in the Internet of Things

13.1 Internet of Things

The Internet of Things (IoT) is formed of a vast range of device types that typically have a significantly longer life-cycle compared to traditional internet-connected devices like clients and servers. Their applications can range from huge medical imaging tools to tiny gadgets. The extended lifespan and - in many cases - great certification efforts lead to variable software update cycles. First, the roll-out of

[18] Change Log of Firefox 119, Mozilla Blog Post on ECH introduction, Mozilla Wiki on ECH
[19] Change Log of Firefox 129
[20] Tor Issue Tracker: Think about using DNS over HTTPS for Tor Browser
[21] https://letsencrypt.org/2022/09/07/new-life-for-crls/

ECH on such devices poses a special challenge. And further, when a vulnerability or bug is identified within the software stack, the challenge of rolling out the necessary updates becomes substantially more complex.

Moreover, numerous IoT devices lack automatic update capabilities at all, and in some instances, these updates are not feasible as well. The underlying software quality often suffers as well, driven by constraints related to financial resources and the sheer scale of development. Paradoxically, due to the critical nature of maintaining device security amid the difficulties associated with updates, the quality of the software must actually be higher to mitigate these risks effectively beforehand.

13.2 Implications by Packaging

TLS libraries that are often packaged by other software provided by package management systems[22]. They might also face a similar challenge as IoT devices, but to a much smaller extent. Operating system providers for are keen on stability, new features are often released in batches as major/minor releases. For server systems these span can in practice be five years or more. The percentage of early adopters will likely be very small, and only viable for those with strong incentives.

13.3 Libraries and Programming Languages

To create a significant impact, ECH must be user-friendly and safe for developers, with integration into high-level APIs. Currently, this is not fully realized, and work is ongoing to address this issue. For updates on progress in this area, please refer to other reports from the Defo project.

For instance, to use ECH in CPython, the current proposal involves usage of low-level APIs[23]. However, for ECH to achieve widespread adoption, it needs to be accessible through high-level APIs, such as httpx[24], and be activated by default.

13.4 Command Line Tools

The proposal for curl requires DoH[25], similar to what Firefox implemented in the past. However, this should only be considered an intermediate step.

To encourage greater usage of Encrypted Client Hello (ECH), curl should also offer the ability to use ECH without the need for explicit DoH. The command line switch to enable ECH (`--ech true`) seems reasonable at the moment and should later by active by default.

[22] https://en.wikipedia.org/wiki/Package_manager
[23] https://github.com/defo-project/ech-dev-utils/blob/e375acd0a1ee4b8abe4f89d60cac5af624931c77/scripts/ech_url.py#L84-L104
[24] https://www.python-httpx.org/
[25] https://github.com/sftcd/curl/blob/25f2c38/docs/ECH.md#user-content-supporting-ech-without-doh

Further work is necessary in this area, particularly by adding support for more tools, especially wget[26].

Without ECH, clients who do not implement it risk creating side channels that can leak domain names, leading to potential de-anonymization.

14 Tor Network

Currently, Encrypted ClientHello (ECH) is not supported on the Tor network. Tor's architecture is designed well to enhance security and privacy, reducing the necessity for the additional layers that DoH and ECH provide. Also, Tor addresses the concerns that both DoH and ECH aim to resolve through its Tor onion services.

Previously, Firefox mandated using DNS over HTTPS (DoH) for ECH functionality (see Section Browsers), which was a blocker as Tor does not use or support DoH. Contrary to DoT and DoH, Tor employs an alternative approach for Name resolution inside the Tor network[27].

There are ongoing discussions on whether and how DoH can benefit for the users' privacy and how it needs to be configured and implemented. For detailed information, we refer to the discussion of the Tor community on DNS over HTTPS (DoH) in Tor[28] and on Encrypted ClientHello (ECH) in Tor[29]. For more information on how Tor protects their users' privacy, please refer to the Tor website[30].

15 Intended deactivation

Implementing Encrypted Client Hello (ECH) in organizations that succumb to specific IT security requirements (ISO standards, NIS2, etc) presents several challenges that may lead them to deactivate ECH usage entirely.

15.1 Incident Detection and Network Monitoring

Many security tools depend on the Server Name Indication (SNI) for threat detection, content filtering, and identifying malicious domains. Losing this information with ECH, organizations may struggle to maintain visibility into network traffic.

[26] https://github.com/defo-project/ech-dev-utils/blob/475fc17/howtos/wget.md
[27] Tor specifications: Remote hostname lookup: http://i3xi5qxvbrngh3g6o7czwjfxwjzigook7zxzjmgwg5b7xnjcn5hzciad.onion/tor-spec/remote-hostname-lookup.html, Opening a new stream: The begin/connected handshake: http://i3xi5qxvbrngh3g6o7czwjfxwjzigook7zxzjmgwg5b7xnjcn5hzciad.onion/tor-spec/opening-streams.html#opening
[28] https://gitlab.torproject.org/tpo/applications/tor-browser/-/issues/30753
[29] https://gitlab.torproject.org/tpo/applications/tor-browser/-/issues/42144
[30] https://support.torproject.org/censorship/

15.2 Incident Response

Obscured connection details can slow down or hinder incident investigations.

16 Censorship

This section provides a short overview of the regions currently use SNI for censorship purposes, those blocking ECH usage, and countries that may soon implement similar measures.

For a comprehensive analysis of internet censorship practices around the globe, see A Survey of Worldwide Censorship Techniques[31] and Open Observatory of Network Interference[32].

16.1 Russia

Russia is known to block traffic to Cloudflare when ECH is in use.[33] The Russian government utilizes Server Name Indicator (SNI) information to enforce censorship measures. Additionally, the administration promotes the use of domestic service providers instead of Cloudflare and other foreign Content Delivery Networks. This shift aims to facilitate greater domestic control over data transmission and access.

16.2 China

The Great Firewall (GFW) of China is one of the most extensive censorship implementation in the world and reports indicate that China is blocking ESNI and ECH.[34]

The GFW utilizes SNI alongside other technologies enforce content blocking.

16.3 South Korea

South Korea uses SNI to restrict access to specific online resources.[35]

While people in South Korea could previously used ESNI as a workaround to bypass these restrictions, browser updates have removed support for ESNI, complicating efforts to maintain online privacy.

[31] https://www.ietf.org/archive/id/draft-irtf-pearg-censorship-10.html
[32] https://ooni.org/
[33] https://github.com/net4people/bbs/issues/417 https://therecord.media/russia-blocks-thousands-of-websites-that-use-cloudflare-service
[34] https://gfw.report/blog/gfw_esni_blocking/en/ https://github.com/net4people/bbs/issues/43
[35] https://www.bleepingcomputer.com/news/security/south-korea-is-censoring-the-internet-by-snooping-on-sni-traffic/ https://www.technadu.com/south-korea-extend-site-blocking-snooping-sni/58125/

As laid out in Incentives, the affected industries have also commercial interest in using ECH.

It is yet unclear how South Korea's authorities will react to ECH.

16.4 Kazakhstan

Kazakhstan uses a national Certificate Authority (CA) to intercept and decrypt TLS traffic.[36] In this case, ECH offers no benefit at all, as the central decryption of all traffic serves all counter-measures ineffective.

16.5 Others

Several countries use unencrypted SNI for filtering and blocking websites, including:

Cuba, Egypt, India, Iran, Saudi Arabia, Syria, Turkey, Turkmenistan, United Arab Emirates, Vietnam

Once ECH plays a relevant role, we expect that these countries block ESNI/ECH to ensure the effectiveness of their measures.

17 De-anonymization by Metadata

17.1 Explicit: ECH usage

The ECH standard appears to be well designed minimizing metadata.

Only the support of ECH by clients in itself is left as a slightly suspicious marker. See Censorship for more information.

The GREASE mitigation effectively prevents that censors may use the existence of an ECH Extension in the Client Hello as identifier for active ECH use, as all clients supporting the standard - including major browsers - should always send a GREASE `encrypted_client_hello` extension also if ECH is not in use.

17.2 Implicit: DNS queries

As the Deployment Overview outlines, clients request the ECH configuration via DNS during the initial setup and for subsequent refreshes.

[36] https://censoredplanet.org/kazakhstan

Browsers utilize the same DNS-over-HTTPS (DoH) server across all installations, which presents several risks:

- Single point of failure
- Single point of truth
- Central point for de-anonymization attacks

These risks can be mitigated by using local - decentralized - resolvers over DoT, as with Do53, instead of central DoH servers.

It needs to mentioned that only the initial request causes a DNS lookup for the ECH Config. For the next key exchanges when the ECH key changes during the same user session, the server proactively sends the next ECH Config to the client with the `retry-config` ECH Extension. This method only works if the lifetime of the ECH Config covers the time between two calls of the client. The ECH key lifespan used by Cloudflare, which was then adopted by other players, is only one hour, which prevents effective caching. The ECH standard does not contain a guideline or recommendation on key refresh interval, only that "It is RECOMMENDED that servers rotate keys regularly."[37]

18 Correlations on traffic patterns

There is existing theoretic extensive research on this topic. Traffic correlations do not depend on unencrypted SNI, so ECH will not have any effect on this.

The idea is that you can still correlate a lot if you have independent datasets such as:

- CTLs (Certificate Transparency Logs)
- DNS traces (i.e. if you operate a DoH recursor such as the large CDNs / google / quad9, DNS4EU, etc.)
- active scanning data (which pages are hosted on which IPs?)
- A large corpus of HTTP Host headers (not everyone is going to switch immediately to ECH)
- Tor exit node traffic

etc.

Furthermore, correlation can be done on typical patterns such as fingerprints of traffic streams of static pages.

The effectiveness of traffic classification in real-world scenarios remains unclear. We refer to existing materials on the subject of "Encrypted (Network) Traffic Classification". Examples include:

- https://www.sciencedirect.com/science/article/pii/S2090447923002502

[37] draft-ietf-tls-esni-25 section 10.10.5. Maintain Forward Secrecy

- https://ieeexplore.ieee.org/document/8622812
- https://www.mdpi.com/2073-8994/13/6/1080

19 By Legal Means

As discussed in the section Censorship, some countries, such as Russia, already use legal measures to enforce censorship. Others are expected to adopt similar practices in the near future. ECH is unable to influence or bypass these restrictions.

20 Other references

- DNS over QUIC - sidn.nl
- DNS over QUIC - nordvpn.com
- Encrypted Client Hello (ECH) - Frequently asked questions - support.mozilla.come
- Tor protocol overview: http://i3xi5qxvbrngh3g6o7czwjfxwjzigook7zxzjmgwg5b7xnjcn5hzciad.onion/rend-spec/protocol-overview.html
- Detailed explanation on Tor's non-usage of DoH can be found here: https://lists.torproject.org/mailman3/hyperkitty/list/tor-dev@lists.torproject.org/thread/6GDO7CYEFIKID7QQCRVYVFNIVETWWWWY/#6ZBFGNSRPWRCEO7PVPSHHVLAOGF7KN3C