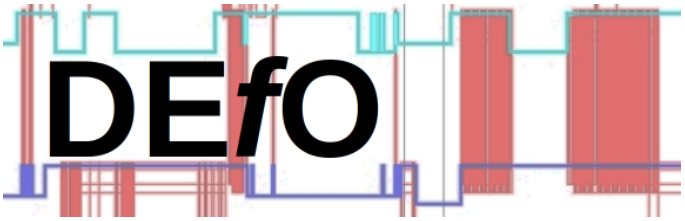

ECH Interoperability Report Version 2

Stephen Farrell
Trinity College Dublin/Tolerant Networks Ltd.
stephen.farrell@cs.tcd.ie

Monday 20th October, 2025

work-in-progress: build time 22:12 UTC

Abstract

Encrypted Client Hello (ECH) is a privacy-enhancement for the Transport Layer Security (TLS) protocol that underlies all web security and the security of many other Internet protocols. While the specification for ECH is relatively mature, (though not yet an Internet-RFC), and while implementations are already widespread, work remains to ensure that a random client and server can successfully use ECH. To that end, this report describes software libraries, client and server software packages, and Internet services for which ECH is relevant, and configurations of those where ECH currently works, or fails, as part of overall efforts to encourage ECH deployment and hence improve Internet security and privacy. This is the second iteration of this report, the first was done in January 2025. For this iteration packages were updated to current versions and tests were re-run. At the time of writing, the overall interoperability story for ECH could be summed up as: if you stick to a simple deployment that works with browsers, that'll be fine, but if you want to push out the boundaries, then something will break, so don't do that. We justify that claim via a test setup involving 67 ECH test URLs (some with broken or non-normal setups), six test clients, and coverage of almost all known ECH-capable server technologies. Currently, our tests indicate that things are about 90% "OK" but are not yet more than 95% "OK", and indeed while our test results for Firefox have improved somewhat, those for chromium have gotten significantly worse.

Contents

Keywords: Encrypted Client Hello (ECH), Interoperability

1 Introduction

Deployments of the Transport Layer Security (TLS [1]) protocol expose the name of the server (e.g. the web site DNS name) via the Server Name Indication (SNI) field in the first message sent (the ClientHello). The Encrypted Client Hello (ECH) [2] extension to TLS is a privacy-enhancing scheme that aims to address this leak. This report describes the current state of ECH interoperability. The primary audience for this document are those implementing and deploying ECH. Secondarily, there may be lessons to learn for those designing protocols like ECH.

In May-June 2025, together with the Wikimedia Foundation, the DEfO project carried out a small ECH trial using Wikimedia servers.[3] The high-level result of that was twofold: there were no service issues caused by ECH-enabling servers, and ECH worked for about one third of requests to the ECH-enabled resources. As more browsers are ECH-enabled that figure should increase, but there were also many cases of browsers GREASEing despite that user agent string indicfating that that browser version can really do ECH.

The Open Technology Fund (OTF, `https://www.opentech.fund/`) have funded the DEfO project (`https://defo.ie`) to develop ECH implementations for OpenSSL, and to otherwise encourage implementation and deployment of ECH. As we expect the implementation and deployment environment for ECH to change over time, this report will be updated as events warrant and is currently versioned based on the build-time of this PDF.

The first version of this report was published in January 2025. [4] For this version we updated all software to latest versions and re-ran tests.

The latest version can be found at `https://github.com/defo-project/ech-interop-report/blob/main/ech-interop-report.pdf`. Comments, additions or corrections are welcome. Those can be sent via email to the author, or as issues or PRs (preferably for the latex files in the "source2"

directory) in the github repository for this report which is `https://github.com/defo-project/ech-interop-report`.

2 Software

This section describes libraries, clients and servers that support ECH.

2.1 Libraries

The libraries listed in Table 1 have some support for ECH. Of the libraries listed, ECH support for all of the OpenSSL variants, Python, libcurl and Conscrypt, were developed by the DEfO project.

Table 1: Libraries with ECH

Library Name	Details
OpenSSL-feature-branch	Source: https://github.com/openssl/openssl/tree/feature/ech Version: 3.6.0-dev feature branch in upstream repo ECH support: ECH shared-mode client and server very basic "s_client", TLS only (no DTLS, no QUIC) the OpenSSL project's target for ECH PRs ECH code will end up here before being merged to master
OpenSSL-defo	Source: https://github.com/defo-project/openssl/ Version: 4.0.0-dev fork of master, same APIs as OpenSSL-feature-branch, but full ECH code ECH support: ECH shared and split mode client and server, TLS only (no DTLS, no QUIC) this is used for DEfO project CI builds and tests, and is rebased to upstream frequently
OpenSSL-sftcd	Source: https://github.com/sftcd/openssl/tree/ECH-draft-13c Version: 3.4.0-dev fork of master ECH support: client and server, TLS only (no DTLS, no QUIC) this was the DEfO project's original main development branch, but is now obsoleted
boringssl	Source: https://boringssl.googlesource.com/boringssl Version: boringssl doesn't do versions, last local build 2025-10-12 ECH support: ECH shared-mode client and server, ECH for TLS, QUIC and (possibly) DTLS in production use (Chromium, et al), a little more limited in HPKE suites than OpenSSL - only KEM is x25519
AWS-LC	Source: https://github.com/aws/aws-lc Version: 1.62.0 ECH support: ECH client and server, ECH for TLS (unsure of DTLS/QUIC) A versioned library based on boringssl Seems to work as well in terms of ECH interoperability
NSS	Source: https://github.com/nss-dev/nss.git Version: NSS 3.116 ECH support: ECH client and server, ECH for TLS (unsure of DTLS/QUIC) in production use (Firefox), a little more limited in HPKE suites than OpenSSL - only KEM is x25519
WolfSSL	Source: https://github.com/wolfSSL/wolfssl.git Version: 5.8.2 ECH support: ECH client and server, ECH for TLS (unsure of DTLS/QUIC) - ECH not built by default (needs "–enable-ech") - fails when HelloRetryRequest seen - https://github.com/wolfSSL/wolfssl/issues/6802
gnuTLS	Source: https://gitlab.com/gnutls/gnutls/blob/master/README.md Version: work-in-progress ECH support: interop untested (by DEfO-project) at this time an ECH merge request exists but has yet to be merged https://gitlab.com/gnutls/gnutls/-/merge_requests/1748 Seemingly no progress since Version 1 of this report
golang	Source: https://go.googlesource.com/go or https://github.com/golang/go/ Version: 1.23 or later required for ECH client, current version: 1.24.4 ECH support: ECH shared-mode client and server
rustls	Source: https://github.com/rustls/rustls/ Version: 0.23.32 ECH support: client only

Continued on next page

Table 1: Libraries with ECH (Continued)

Library Name	Details
CPython	Source: https://github.com/defo-project/cpython Version: CPython 3.13/3.14 ECH support: TLS client only - CPython tests were just (2024-12-05) added to our smokeping tests
libcurl	Source: part of https://github.com/curl/curl Version: 8.17.0-DEV ECH support: TLS client only
Conscrypt-GP	Source: https://github.com/guardianproject/conscrypt Version: 2.6-alpha ECH support: only ECH client is relevant This is based on boringssl and only runs on Android

2.2 Clients

Current versions of Firefox (e.g. 143.0, used in our smokeping tests) support ECH by default. Firefox does, however, automatically disable ECH when it detects the presence of "family safety" software [1]. We have received anecdotal evidence that this detection mechanism also triggers when detecting ISP-level and state-imposed censorship, preventing the use of ECH in those situations.

The Tor Browser (a Firefox derivative) doesn't support ECH by default, as it is currently built from an older "Extended Support Release" that only enables ECH when DNS-over-HTTPS (DoH) is enabled and Tor Browser prevents enabling DoH to prevent that being used as a user fingerprinting vector. To test if a browser supports ECH one can visit `https://test.defo.ie/` or, for much more detail `https://test.defo.ie/iframe_tests.html`.

Current versions of Chromium (version 140.0.7339.185 is used in our tests) also support ECH by default. Other Chromium-derived browsers also support ECH, while we don't include them in our ongoing tests described below, we have manually verified that current versions of Vivaldi, Brave and Opera also have ECH support.

Browsers in general are still somewhat quirky in terms of whether or not ECH will succeed, even when ECH is properly setup. This seems at least partly due to the delay that can inherently occur between the browser receiving DNS answers for A/AAAA records, and the time at which the browser receives a DNS answer for an HTTPS record (containing ECH information). It seems to be the case that browsers may be "impatient" in that they start the TLS session without attempting ECH if that delay is too long, and that that happens sufficiently often to be noticeable. (Note however, that this report does not in itself yet justify that claim, but work is ongoing.)

As this "impatience" has seemingly dis-improved since version 1 of this report we investigated further to see if we could determine for sure that our suspicions were correct. We did this with a variant of our smokeping script for chromium that use localhost port 53 for it's DNS recursive. In that setup we could see the IMCP error messages ('port unreachable') when the recursive was attempting to provide HTTPS RR answers. This seems to confirm that the browser has given up waiting for the HTTPS RR answer. We also confirmed this via chromium "netlog" traces where we see "ERR_DNS_TIMED_OUT". Informal discussion with people more familiar with chromium internals indicates that this situation may be a result of balancing contradictory requirements - some scenarios apparently call for failing quickly while others (such as ours) suffer from that impatience. As there does not appear to be a way to control those chromium-internal DNS timeouts in our tests, we properly report these unexpected test results as failure cases.

The only command line tool supporting ECH of which we're aware is curl. The DEfO project contributed ECH code as an experimental feature for curl that was merged to the master branch in April 2024. As an experimental feature, ECH is not built by default nor part of a release, so is only currently available to those who build from source. Curl supports using OpenSSL, boringssl or Wolfssl libraries for ECH. Build instructions are at `https://github.com/curl/curl/blob/master/docs/ECH.md`.

As a scripting tool, often used as part of a TLS client, Python also fits here. The DEfO project

[1] https://support.mozilla.org/en-US/kb/faq-encrypted-client-hello

CPython build adds ECH support to Python via the OpenSSL library. That build is available at `https://github.com/defo-project/cpython` with the new APIs required for ECH described at `https://irl.github.io/cpython/library/ssl.html#encrypted-client-hello`.

Conscrypt is Android's library for TLS. It is used in three forms: built into Android OS, a system-wide standalone TLS provider, and a library to provide TLS in an app. Our fork of that has ECH support via boringssl from Google which implements the Android side. Since it is quite unlikely that a multi-domain web service would ever run on Android, only the ECH client side is relevant.

2.3 Servers

The servers listed in Table 2 support ECH. In each case, the ECH integration, using OpenSSL, was developed as part of the DEfO project.

Table 2: Servers supporting ECH

Server Name	Details
lighttpd-upstream	Source: `https://github.com/lighttpd/lighttpd1.4.git` Version: 1.4.83 Comment: Upstream supports shared-mode ECH if built with ECH-enabled boringssl or OpenSSL
lighttpd-defo	Source: `https://github.com/defo-project/lighttpd` Version: 1.4.83 Comment: as for lighttpd-upstream but to enable these tests is built with "CFLAGS=-DLIGHTTPD_OPENSSL_ECH_DEBUG"
nginx	Source: `https://github.com/defo-project/nginx` Version: 1.29.2 Comment: both shared-mode and split-mode ECH
apache2-upstream	Source: `https://github.com/apache/httpd` Version: 2.4.65 Comment: Upstream supports shared-mode ECH if built with ECH-enabled boringssl or OpenSSL
apache2-defo	Source: `https://github.com/defo-project/apache-httpd` Version: 2.5.1.dev Comment: only shared-mode ECH
haproxy	Source: `https://github.com/defo-project/haproxy` Version: 3.3.dev8.197 Comment: both shared-mode and split-mode ECH
OpenSSL s_server	Source: `https://github.com/defo-project/openssl` Version: 4.0.0-dev Comment: shared-mode only but can easily "force" HelloRetryRequest (HRR)

3 Services

3.1 Test Services

We can separate services into test and operational services. Table 3 lists known test services supporting ECH. All .ie hosts, myechtest.site and my-own.net were setup by the DEfO project.

Table 3: Test Services with ECH

Service Name	Details
defo.ie	`https://defo.ie/ech-check.php` is often used to check ECH ECH keys are rotated hourly at :42, private usable for 3 hours, ECHConfigList in DNS contains only latest public
draft-13.esni.defo.ie	different server technology (as listed in Table 2) instances on different ports as listed at `https://defo.ie/`, e.g., `https://draft-13.esni.defo.ie:10413` is served by nginx ECH keys are rotated hourly at :42, private usable for 3 hours, ECHConfigList in DNS contains only latest public
test.defo.ie	hosts a number of ECH server setups with good and variously bad configurations - see `https://test.defo.ie/iframes_tests` which describes those and allows a browser to attempt connections to each via Iframes ECH keys/configs are static for these setups

Continued on next page

Table 3: Test Services with ECH (Continued)

Service Name	Details
foo.ie	https://foo.ie/ech-check.php was setup used to check the defo.ie setup was easily replicated ECH keys are rotated hourly at :42, private usable for 3 hours, ECHConfigList in DNS contains only latest public
my-own.net	this was to test the impact of having the same ECH keys on port 443 (https://my-own.net/ech-check.php) and another port (https://my-own.net:8443/ech-check.php) - at one point that made a difference to browsers ECH keys are rotated hourly at :42, private usable for 3 hours, ECHConfigList in DNS contains only latest public
myechtest.site	this server always treats ECH as if it were GREASE and, in addition always returns malformed values in retry-configs to enable some fuzzing of client handling of retry-configs the "malformed-ness" varies randomly, so if testing against this try record the values received as we don't (currently) have comprehensive logging of client accesses and the fuzzed values returned
tls-ech.dev	https://tls-ech.dev/ was setup by the boringssl developers as a test server that uses boringssl ECH keys/configs seem to be static for this setup
Cloudflare	https://cloudflare-ech.com/cdn-cgi/trace is a test page setup by cloudflare that reports on ECH success/failure apparently, the server implementation and infrastructure are part of Cloudflare's normal setup a similar test service used to be available at https://crypto.cloudflare.com/cdn-cgi/trace but that was turned off around the time that ECH was re-enabled for Cloudflare customers based on one test on 2024-12-09, new ECH keys are published roughly hourly with a 300 second TTL; old ECH public values seem usable for approximately 4 or up to 5 hours; and the ECHConfigList published in DNS contains only one public value

3.2 Operational Services

The only operational ECH service we know of is Cloudflare's deployment, though our domain probe data in the next sections hints that that may be starting to change. However, the Cloudflare deployment is non-negligible. Cloudflare earlier enabled ECH but disabled it soon after in October 2023 [2] as it caused some back-end issues. They then re-enabled ECH in October 2024. Our understanding of Cloudflare's deployment is that ECH is enabled by default for their "free" tier customers, but that paying customers have to take action to enable ECH.

In contrast, non-paying customers are not provided with a control to disable only ECH - they reportedly have to downgrade to TLSv1.2 in order to disable ECH. This is something that has been seen in recent weeks after the Russian government started blocking use of ECH to Cloudflare. [3]

3.3 Domain Probe Data

As part of our DEfO test setup, we have a web page where one can enter a host name and port and a script on our server will check if ECH is enabled for a web server at the name and port. (That's at `https://test.defo.ie/domainechprobe.php`.) Data is only stored if there is an HTTPS resource record for the name and port, if there is not we store nothing. In cases where this is an HTTPS resource record, we only store the name and port, whether the HTTPS value includes an "ech=" field, if there is, whether or not ECH worked using our ECH-enabled curl as the client. We also store the HTTPS resource record value, unless there are CNAMES or other kinds of re-direction involved.

We can use this data to get some further insight into services for which ECH is enabled. For now, we are not publishing the raw data - while data for the most recent 50 queries is shown on the web page, if we wanted to publish the raw data, we'd have to enable some form of consent for users, and it's not clear that'd be a) easy or b) worthwhile. (We currently take no steps to try record who made which requests to this page.)

Figure 1 shows the cumulative number of queries made between August 2024 and October 2025, with Figure 2 showing the number of queries per day. We can see a noticeable uptick in accesses just-before and after Russia blocked Cloudflare's ECH service in November 2024. There was also a service outage between 2025-07-18 and 2025-08-19, and a burst of queries on 2025-09-25 that looks like an automated script, but seems to have no interesting pattern in terms of the names

[2] https://community.cloudflare.com/t/early-hints-and-encrypted-client-hello-ech-are-currently-disabled-globally/567730

[3] https://betanews.com/2024/11/20/encrypted-client-hello-didnt-solve-censorship-but-still-may-have-a-role-to-play/

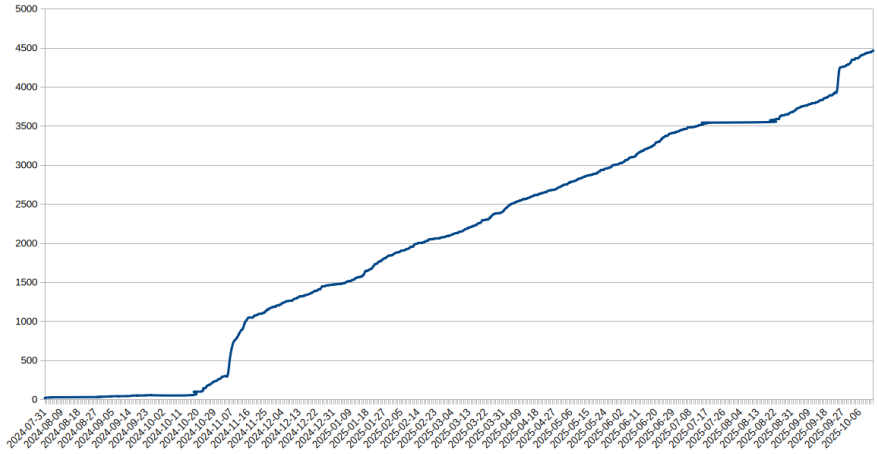

Figure 1: Cumulative number of queries seen at `https://test.defo.ie/domainechprobe.php` versus time. The total number of queries was 4465 as of 2025-10-13.

queried. Otherwise we see a steady rate of a few queries per day as one would expect for a service like this.

Table 4 lists some of the numbers involved with this test. So at least for the names entered to our domain probe page, we can seemingly conclude that Cloudflare have the by-far dominant service at present with 95% of the most recent successes using ECH likely deployed on the Cloudflare network. However, we do see at least a number of other ECH client facing servers now whereas in December 2024 there was only Cloudflare plus a few test services, including those operated by the DEfO project. Table 5 breaks down the non-Cloudflare ECH success cases some more - we basically see that we've gone from 2 "hobbyist" deployments in December 2024 to about 25 seemingly ECH independent deployments in October 2025 (not counting DEfO project deployments). That's still a tiny number but may indicate a good level of desire to deploy ECH, given that at the time of writing ECH is not available as a standard feature in released web server packages.

In version 1 of this report we saw some usage (at least 15 cases) that appeared to indicate some web site owners may have been using this service to check whether or not they have successfully disabled ECH. (The pattern was two quick accesses to the same name/port, the first of which indicates ECH worked, and the second that the HTTPS RR no longer contains an "ech=" value.) Figure 3 shows the first-seen to last-seen times for 119 domains that may have "switched off" ECH. Table 6 shows the counts of the TLDs for the domains concerned, with ".ru" domains being the clear outlier. The median time for which we saw ECH enabled for these domains was only 6198 seconds, so in many cases we see that pattern of a check where ECH succeeds followed shortly thereafter by a check where ECH no longer succeeds. Note though that for example "crypto.cloudflare.com" is a domain that was used for ECH interoperability testing that was subsequently re-purposed for another trial, so not all the "turn-offs" here are the same. (That domain corresponds to the longest bar in Figure 3.)

4 Interoperability

Our primary interoperability testing is based on our "smokeping"-like tests. While smokeping measures latency for names/addresses, our tests aim to establish whether and which ECH configurations interoperate or fail to do so.

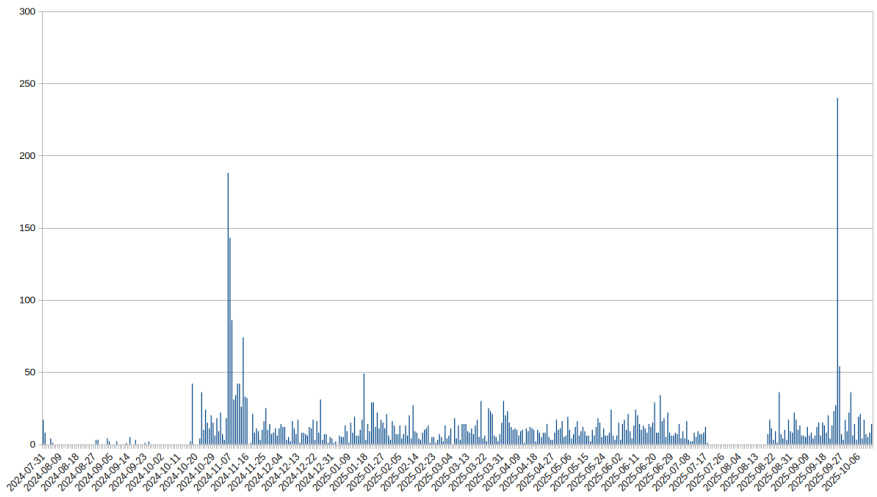

Figure 2: Queries per day seen at `https://test.defo.ie/domainechprobe.php` versus time. Overall average: 13, average over the month to 2025-10-13: 22.

Table 4: Counts of Domains Scanned from 2024-07-31 to 2025-10-13. Version 1 of this document reported on the queries to this service seen up to December 2024; where number weren't reported then, we show a "-". "pn" refers to the ECH "public_name" field seen. The "Last access" rows sample the queries counting only the last successful use of ECH for the domain name concerned.

Count Type	2024-12	2025-10
Total Records	1244	4465
Unique ECHConfigList	276	1067
Unique Names with ech=	298	970
ech= and Cloudflare NS	271	794
ech= and DEfO Test Site	15	14
ech= but now NXDOMAIN/SERVFAIL	9	66
ech= and non-DEfO and non-Cloudflare	1	5
ech= and other	3	110
Records with ech=	660	1916
ech= and pn==cloudflare-ech.com	-	1734
ech= and pn!=cloudflare-ech.com	-	180
ech= and bad ECHConfigList	-	2
Last accesses with ech=	-	953
Last accesses with ech= and pn==cloudflare-ech.com	-	903
Last accesses with ech= and pn!=cloudflare-ech.com	-	50

Table 5: Counts of Domains per related ECH "public_name" operator for the 50 non-Cloudflare "Last access" ECH success cases shown in the last row of Table 4. The last row in this table reflects the DEfO project operated test services but all the others are independent ECH deployments.

Count of domains "protected"	Count of ECH "operators"
1	19
2	3
3	1
5	2
12	1

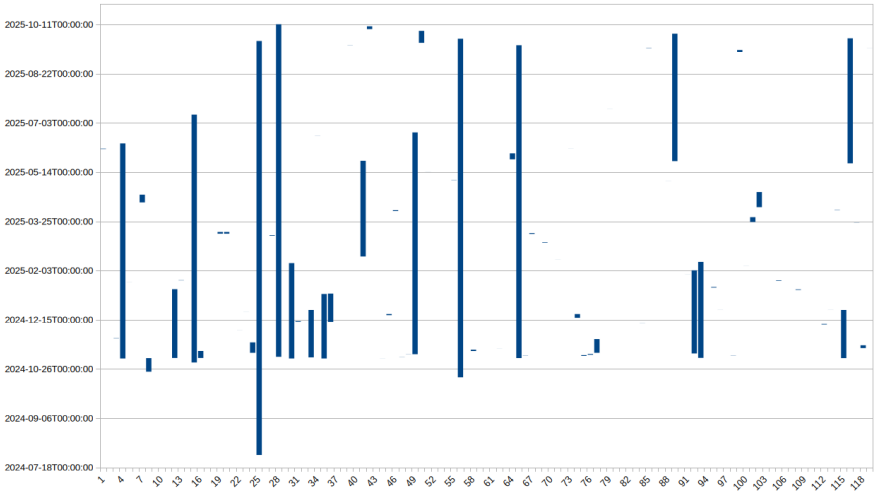

Figure 3: Span of time for which we saw ECH enabled, for domains where the last thing we saw was ECH disabled.

Table 6: Counts of TLDs for domains that "turned off" ECH.

TLD	Count	TLD	Count
by	1	nu	1
casino	1	online	5
cc	2	org	2
click	1	qpon	1
com	16	ru	36
dev	2	services	1
hk	1	shop	3
info	2	space	3
io	3	su	1
me	2	tech	1
movie	1	top	24
net	4	tv	3
nl	1	xyz	1

4.1 Test Clients

We mainly do this via a set of hourly cronjobs running on the test.defo.ie VM that attempt to access 67 URLs (some deliberately with broken configurations) from six different clients (described in Table 7. Between version 1 and 2 of this report client packages have been updated as per the version numbers shown in Table 7. The test scripts that call those client and the cron setup remain the same. We report results for the most recent 6 runs at `https://test.defo.ie/smokeping-summary.php`.

Table 7: Smokeping clients

Test Client Name	Details
Chromium	headless browser tests via Selenium 2025-01 Version: 131.0.6778.85; 2025-10 Version: 140.0.7339.185 Python script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/selenium_test.py ECH implementation based on boringssl Uses DoH to https://chrome.cloudflare-dns.com/dns-query
curl	bash script using curl 2025-01 Version: 8.11.1-DEV; 2025-10 Version: 8.17.0-DEV Bash script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_curl.sh curl currently only pays attention the the first HTTPS resource record seen in DNS answers ECH implementation based on OpenSSL Uses DoH to https://one.one.one.one/dns-query
Firefox	headless browser tests via Selenium 2025-01 Version: 133.0; 2025-10 Version: 143.0 Python script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/selenium_test.py ECH implementation based on NSS Uses DoH to https://one.one.one.one/dns-query
golang	custom golang programme and bash script 2025-01 Version: golang 1.23; 2025-10 Version: golang 1.24.4 Golang programme: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/ech_url.go Bash script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_go.sh ECH implementation developed for golang Uses DoH to https://cloudflare-dns.com/dns-query using GET
rustls	custom rustls programme and bash script 2025-01 Version: rustls 0.23.19; 2025-10 Version: rustls 0.23.32 Rustls programme: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/ech_url.rs Bash script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_rs.sh ECH implementation developed for rustls Uses DoH to Google DNS (8.8.8.8/8.8.4.4) via Hickory DNS
Python	custom CPython build and python script 2025-01 Version: CPython 3.13; 2025-10 Version: CPython 3.13.1 Python script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/ech_url.py Bash script: https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_py.sh ECH implementation based on OpenSSL Uses Do53 to system resolver

There are some notes on these tests worth calling out in case someone wants to reproduce them:

- Some of the headless browser tests (e.g. with badly encoded ECHConfigList values) will cause Selenium to throw exceptions, so test scripts need to catch those in order to properly determine that we got an expected failure.
- Firefox will not attempt ECH to a name when the hostname of the test machine/VM is beneath the "second level domain"[4] for that name, thus causing unexpected failures if that hostname matches the origin of a test URL. In our case, we run both servers and test clients on the same VM ("test.defo.ie") so we have to set the hostname on the VM to something "peculiar" for tests to run as expected. (That of course causes "sudo" to complain which could disturb logs.)

[4]Or 2LD, as defined e.g. at `https://en.wikipedia.org/wiki/Second-level_domain`.

- In the course of testing ECH in recent years, we have repeatedly made one misconfiguration error: using "." as the target name in HTTPS RRs for ports other then 443, when the correct thing do is to include the origin as the target name in such cases. That is, when making an HTTPS RR for `https://example.com:12345/` one needs to publish an HTTPS RR at "_12345._https.example.com" and the (presentation) value of that record needs to start with "1 example.com ..." where 1 is the priority and example.com is that targetName. So simply copying the value of the HTTPS RR for port 443 and re-publishing that for another port is not sufficient for interoperability. (But also does work for some clients in some cases, making it easier to accidentally do this and not notice.)

- Tests are run hourly, with curl tests at the top of the hour, CPython tests at :05, Firefox at :10, Chromium at :20, golang at :30 and rustls at :40. The test summary page is generated at :45. The TTLs for DNS records below test.defo.ie are all set to 10 seconds. For our other DEfO test services (e.g. draft-13.esni.defo.ie cases) TTLs are set to 1800 seconds.

We also have a standalone test suite on Android to run our ECH-enabled Conscrypt build against a number of domains (`https://github.com/defo-project/EchInteropTest`). This manually-run test passes on all of the draft-13.esni.defo.ie test servers, defo.ie itself, Google's tls-ech.dev, and two Cloudflare domains that support ECH.

4.2 Results

The set of 67 test URLs covers all of the server technologies listed in Table 2 though as all of those use OpenSSL for ECH, we direct most of the URLs towards one nginx instance. On our test VM, we have an haproxy instance (not doing ECH) listening on port 443, that routes TLS sessions to the ECH-enabled web servers, namely, nginx (port 15443), apache (15444), lighttpd (15445) and two instances of "openssl s_server", one with a standard configuration on port 15447, and one, that only supports the p-384 curve for TLS key exchange on port 15448. That last server's configuration will trigger TLS HelloRetryRequest for many TLS clients. The back-end ports for all servers (e.g. 15443 for nginx) are also open. Our test URLs and associated HTTPS RRs can therefore be exercised both for port 443 and the relevant back-end port, but in most cases we have only set those up for the nginx back-end instance as testing the same setup on different web server instances wouldn't be likely to reveal much new information. The test set also include URLs for the test services listed in Table 3. Some test URLs have "broken" configurations, e.g. with badly encoded HTTPS resource records, or have specification-conformant configurations that we expect may not work.

We have 6 clients testing hourly against 67 URLs, giving us 402 measurements per hour. The test setup underlying these URLs are further described in Appendix A and the list of URLs and expected outcomes is shown in Appendix B. The full set of URLs and the last 6 hours of test outcomes are also at `https://test.defo.ie/smokeping-summary.php` or web pages with (most of) the test URLs loaded via an Iframe can be accessed at `https://test.defo.ie/iframe_tests_sort.html`. That web page also includes some explanatory text about each test URL.

Table 8 below shows, for each test URL and each client, the ratio of expected and total results as per version 1 of this report. [5] Table 9 updates that for a recent interval. Table 10 shows the summary lines from both.

Table 8: ECH interop tests from 2025-01-23 00:00:00 to 2025-01-29 14:24:11.632424.When less than 95 percent of tests are as expected, the cell is in bold text.

num	url	chromium	curl	firefox	golang	rustls	python
1	https://2thenp-ng.test.defo.ie/echstat.php?format=json	**0.93/159**	1.00/159	1.00/159	1.00/158	1.00/158	1.00/159
2	https://2thenp-ng.test.defo.ie:15443/echstat.php?format=json	**0.94/159**	1.00/159	1.00/159	1.00/158	0.99/158	1.00/159

Continued on next page

[5] Table 8 is produced using `https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smokeping-summary-report.py`.

Table 8: ECH interop tests from 2025-01-23 00:00:00 to 2025-01-29 14:24:11.632424.
When less than 95 percent of tests are as expected, the cell is in bold text. (Continued)

num	url	chromium	curl	firefox	golang	rustls	python
3	https://ap.test.defo.ie/echstat.php?format=json	0.96/159	1.00/159	1.00/159	1.00/158	1.00/158	0.99/159
4	https://ap.test.defo.ie:15444/echstat.php?format=json	**0.95/159**	0.99/159	1.00/159	0.99/158	0.99/158	1.00/159
5	https://badalpn-ng.test.defo.ie/echstat.php?format=json	0.96/159	1.00/159	**0.00/159**	**0.89/158**	1.00/158	0.99/159
6	https://badalpn-ng.test.defo.ie:15443/echstat.php?format=json	0.97/159	1.00/159	**0.00/159**	1.00/158	1.00/158	1.00/159
7	https://bk1-ng.test.defo.ie/echstat.php?format=json	1.00/159	1.00/159	1.00/159	1.00/158	1.00/158	1.00/159
8	https://bk1-ng.test.defo.ie:15443/echstat.php?format=json	1.00/159	1.00/159	1.00/159	1.00/158	1.00/158	0.98/159
9	https://bk2-ng.test.defo.ie/echstat.php?format=json	1.00/159	1.00/159	1.00/159	1.00/158	1.00/158	0.99/159
10	https://bk2-ng.test.defo.ie:15443/echstat.php?format=json	1.00/159	1.00/159	1.00/159	1.00/158	1.00/158	1.00/159
11	https://bv-ng.test.defo.ie/echstat.php?format=json	1.00/159	0.99/159	1.00/159	0.98/158	1.00/158	0.99/159
12	https://bv-ng.test.defo.ie:15443/echstat.php?format=json	1.00/159	1.00/159	1.00/159	1.00/158	1.00/158	0.99/159
13	https://cloudflare-ech.com/cdn-cgi/trace	0.96/159	0.99/159	1.00/159	0.99/158	0.99/158	**0.00/159**
14	https://curves1-ng.test.defo.ie/echstat.php?format=json	0.96/159	1.00/159	1.00/159	1.00/158	**0.48/158**	1.00/159
15	https://curves1-ng.test.defo.ie:15443/echstat.php?format=json	**0.94/159**	1.00/159	1.00/159	1.00/158	**0.54/158**	1.00/159
16	https://curves2-ng.test.defo.ie/echstat.php?format=json	0.96/159	1.00/159	1.00/159	0.99/158	**0.56/158**	1.00/159
17	https://curves2-ng.test.defo.ie:15443/echstat.php?format=json	**0.94/159**	1.00/159	1.00/159	1.00/158	**0.50/158**	1.00/159
18	https://curves3-ng.test.defo.ie/echstat.php?format=json	**0.00/159**	1.00/159	**0.00/159**	**0.00/158**	**0.50/158**	1.00/159
19	https://curves3-ng.test.defo.ie:15443/echstat.php?format=json	**0.00/159**	1.00/159	**0.00/159**	**0.00/158**	**0.55/158**	1.00/159
20	https://defo.ie/ech-check.php	0.96/159	1.00/159	1.00/159	0.99/158	0.97/158	1.00/159
21	https://draft-13.esni.defo.ie:10413/	**0.92/159**	0.99/159	1.00/159	1.00/158	0.99/158	1.00/159
22	https://draft-13.esni.defo.ie:11413/	**0.94/159**	1.00/159	1.00/159	1.00/158	**0.95/158**	0.99/159
23	https://draft-13.esni.defo.ie:12413/	**0.00/159**	1.00/159	**0.00/159**	1.00/158	1.00/158	1.00/159
24	https://draft-13.esni.defo.ie:12414/	**0.95/159**	1.00/159	1.00/159	0.99/158	1.00/158	1.00/159
25	https://draft-13.esni.defo.ie:8413/stats	0.96/159	1.00/159	1.00/159	0.98/158	0.99/158	0.99/159
26	https://draft-13.esni.defo.ie:8414/stats	**0.92/159**	0.99/159	0.99/159	0.99/158	0.97/158	0.99/159
27	https://draft-13.esni.defo.ie:9413/	**0.92/159**	1.00/159	1.00/159	0.99/158	1.00/158	0.99/159
28	https://h1alpn-ng.test.defo.ie/echstat.php?format=json	**0.95/159**	1.00/159	1.00/159	0.99/158	1.00/158	1.00/159
29	https://h1alpn-ng.test.defo.ie:15443/echstat.php?format=json	0.96/159	0.99/159	1.00/159	1.00/158	0.99/158	0.99/159
30	https://h2alpn-ng.test.defo.ie/echstat.php?format=json	0.98/159	0.99/159	**0.00/159**	1.00/158	1.00/158	0.99/159
31	https://h2alpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.94/159**	1.00/159	**0.00/159**	1.00/158	0.99/158	0.99/159
32	https://hidden.hoba.ie/	**0.92/159**	1.00/159	1.00/159	1.00/158	1.00/158	1.00/159
33	https://longalpn-ng.test.defo.ie/echstat.php?format=json	**0.95/159**	1.00/159	1.00/159	1.00/158	1.00/158	0.99/159
34	https://longalpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.35/159**	1.00/159	1.00/159	0.99/158	1.00/158	1.00/159
35	https://ly.test.defo.ie/echstat.php?format=json	0.98/159	1.00/159	1.00/159	1.00/158	1.00/158	1.00/159
36	https://many-ng.test.defo.ie/echstat.php?format=json	**0.95/159**	1.00/159	0.99/159	0.99/158	1.00/158	0.99/159
37	https://many-ng.test.defo.ie:15443/echstat.php?format=json	**0.28/159**	1.00/159	**0.00/159**	1.00/158	1.00/158	1.00/159
38	https://min-ng.test.defo.ie/echstat.php?format=json	0.99/159	1.00/159	1.00/159	1.00/158	0.99/158	1.00/159
39	https://min-ng.test.defo.ie:15443/echstat.php?format=json	**0.95/159**	1.00/159	0.98/159	1.00/158	1.00/158	1.00/159
40	https://mixedalpn-ng.test.defo.ie/echstat.php?format=json	0.97/159	1.00/159	1.00/159	0.99/158	0.99/158	1.00/159
41	https://mixedalpn-ng.test.defo.ie:15443/echstat.php?format=json	0.96/159	1.00/159	1.00/159	1.00/158	0.99/158	1.00/159
42	https://mixedmode-ng.test.defo.ie/echstat.php?format=json	1.00/159	1.00/159	**0.00/159**	1.00/158	**0.00/158**	**0.00/159**
43	https://mixedmode-ng.test.defo.ie:15443/echstat.php?format=json	1.00/159	1.00/159	**0.00/159**	1.00/158	**0.00/158**	**0.00/159**
44	https://my-own.net/ech-check.php	**0.94/159**	0.99/159	1.00/159	0.99/158	0.99/158	0.99/159
45	https://my-own.net:8443/ech-check.php	**0.69/159**	0.99/159	1.00/159	0.99/158	0.98/158	0.99/159
46	https://myechtest.site/	**0.01/159**	0.99/159	**0.83/159**	1.00/158	0.99/158	1.00/159
47	https://ng.test.defo.ie/echstat.php?format=json	0.96/159	1.00/159	1.00/159	1.00/158	0.99/158	1.00/159
48	https://ng.test.defo.ie:15443/echstat.php?format=json	**0.93/159**	1.00/159	1.00/159	1.00/158	1.00/158	1.00/159
49	https://noaddr-ng.test.defo.ie/echstat.php?format=json	1.00/159	0.99/159	1.00/159	1.00/158	1.00/158	0.99/159
50	https://noaddr-ng.test.defo.ie:15443/echstat.php?format=json	1.00/159	1.00/159	1.00/159	0.99/158	1.00/158	1.00/159
51	https://p256-ng.test.defo.ie/echstat.php?format=json	**0.00/159**	1.00/159	**0.00/159**	**0.00/158**	**0.00/158**	1.00/159
52	https://p256-ng.test.defo.ie:15443/echstat.php?format=json	**0.00/159**	1.00/159	**0.00/159**	**0.00/158**	**0.00/158**	1.00/159
53	https://pthen2-ng.test.defo.ie/echstat.php?format=json	**0.94/159**	1.00/159	1.00/159	0.99/158	0.99/158	1.00/159
54	https://pthen2-ng.test.defo.ie:15443/echstat.php?format=json	**0.94/159**	1.00/159	1.00/159	0.99/158	1.00/158	0.99/159
55	https://ss.test.defo.ie/stats	**0.94/159**	1.00/159	1.00/159	0.98/158	0.99/158	0.99/159
56	https://sshrr.test.defo.ie/stats	0.96/159	1.00/159	1.00/159	1.00/158	1.00/158	1.00/159
57	https://tls-ech.dev/	**0.84/159**	1.00/159	1.00/159	1.00/158	0.99/158	**0.00/159**
58	https://v1-ng.test.defo.ie/echstat.php?format=json	0.99/159	1.00/159	0.98/159	1.00/158	0.99/158	1.00/159
59	https://v1-ng.test.defo.ie:15443/echstat.php?format=json	0.99/159	1.00/159	0.99/159	1.00/158	0.99/158	1.00/159
60	https://v2-ng.test.defo.ie/echstat.php?format=json	**0.91/159**	1.00/159	0.99/159	0.99/158	1.00/158	1.00/159

Continued on next page

Table 8: ECH interop tests from 2025-01-23 00:00:00 to 2025-01-29 14:24:11.632424.
When less than 95 percent of tests are as expected, the cell is in bold text. (Continued)

num	url	chromium	curl	firefox	golang	rustls	python
61	https://v2-ng.test.defo.ie:15443/echstat.php?format=json	**0.93/159**	1.00/159	1.00/159	1.00/158	0.99/158	1.00/159
62	https://v3-ng.test.defo.ie/echstat.php?format=json	0.97/159	1.00/159	0.99/159	0.99/158	1.00/158	1.00/159
63	https://v3-ng.test.defo.ie:15443/echstat.php?format=json	0.96/159	0.99/159	1.00/159	0.99/158	1.00/158	0.99/159
64	https://v4-ng.test.defo.ie/echstat.php?format=json	1.00/159	1.00/159	1.00/159	1.00/158	**0.66/158**	**0.00/159**
65	https://v4-ng.test.defo.ie:15443/echstat.php?format=json	1.00/159	1.00/159	1.00/159	0.99/158	**0.65/158**	**0.00/159**
66	https://withext-ng.test.defo.ie/echstat.php?format=json	0.97/159	1.00/159	1.00/159	1.00/158	1.00/158	0.99/159
67	https://withext-ng.test.defo.ie:15443/echstat.php?format=json	**0.92/159**	1.00/159	1.00/159	1.00/158	1.00/158	1.00/159
	Totals	**0.85/10653**	1.00/10653	**0.82/10653**	**0.94/10586**	**0.88/10586**	**0.91/10653**
	Count below 95% threshold	34	0	13	5	13	6
	Overall: **0.90/63784**						

Table 9: ECH interop tests from 2025-09-15 00:00:00 to 2025-10-17 19:33:40.501330.When less than 95 percent of tests are as expected, the cell is in bold text.

num	url	chromium	curl	firefox	golang	rustls	python
1	https://2thenp-ng.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
2	https://2thenp-ng.test.defo.ie:15443/echstat.php?format=json	**0.46/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
3	https://ap.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
4	https://ap.test.defo.ie:15444/echstat.php?format=json	**0.46/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
5	https://badalpn-ng.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	**0.73/788**	**0.94/787**	1.00/788
6	https://badalpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.46/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
7	https://bk1-ng.test.defo.ie/echstat.php?format=json	1.00/788	1.00/788	1.00/788	1.00/788	0.97/787	0.99/788
8	https://bk1-ng.test.defo.ie:15443/echstat.php?format=json	1.00/788	0.99/788	1.00/788	1.00/788	0.97/787	0.99/788
9	https://bk2-ng.test.defo.ie/echstat.php?format=json	1.00/788	1.00/788	1.00/788	1.00/788	0.97/787	1.00/788
10	https://bk2-ng.test.defo.ie:15443/echstat.php?format=json	1.00/788	1.00/788	1.00/788	0.99/788	0.97/787	0.99/788
11	https://bv-ng.test.defo.ie/echstat.php?format=json	1.00/788	1.00/788	1.00/788	1.00/788	0.97/787	1.00/788
12	https://bv-ng.test.defo.ie:15443/echstat.php?format=json	1.00/788	1.00/788	1.00/788	0.99/788	0.97/787	0.99/788
13	https://cloudflare-ech.com/cdn-cgi/trace	1.00/788	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
14	https://curves1-ng.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	1.00/788	**0.82/787**	0.99/788
15	https://curves1-ng.test.defo.ie:15443/echstat.php?format=json	**0.46/788**	1.00/788	1.00/788	1.00/788	**0.81/787**	1.00/788
16	https://curves2-ng.test.defo.ie/echstat.php?format=json	**0.93/788**	1.00/788	1.00/788	0.99/788	**0.83/787**	1.00/788
17	https://curves2-ng.test.defo.ie:15443/echstat.php?format=json	**0.45/788**	1.00/788	1.00/788	1.00/788	**0.81/787**	1.00/788
18	https://curves3-ng.test.defo.ie/echstat.php?format=json	**0.00/788**	1.00/788	**0.00/788**	**0.00/788**	**0.83/787**	0.99/788
19	https://curves3-ng.test.defo.ie:15443/echstat.php?format=json	**0.00/788**	1.00/788	**0.00/788**	**0.00/788**	**0.80/787**	1.00/788
20	https://defo.ie/ech-check.php	**0.93/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
21	https://draft-13.esni.defo.ie:10413/	**0.46/788**	1.00/788	1.00/788	1.00/788	**0.93/787**	0.99/788
22	https://draft-13.esni.defo.ie:11413/	**0.45/788**	0.99/788	1.00/788	1.00/788	**0.82/787**	1.00/788
23	https://draft-13.esni.defo.ie:12413/	**0.00/788**	1.00/788	**0.00/788**	1.00/788	**0.93/787**	1.00/788
24	https://draft-13.esni.defo.ie:12414/	**0.45/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
25	https://draft-13.esni.defo.ie:8413/stats	**0.85/788**	1.00/788	1.00/788	0.99/788	**0.93/787**	0.99/788
26	https://draft-13.esni.defo.ie:8414/stats	**0.46/788**	0.99/788	1.00/788	0.99/788	**0.94/787**	1.00/788
27	https://draft-13.esni.defo.ie:9413/	**0.47/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
28	https://h1alpn-ng.test.defo.ie/echstat.php?format=json	0.96/788	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
29	https://h1alpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.45/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
30	https://h2alpn-ng.test.defo.ie/echstat.php?format=json	**0.95/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
31	https://h2alpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.46/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
32	https://hidden.hoba.ie/	0.96/788	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
33	https://longalpn-ng.test.defo.ie/echstat.php?format=json	**0.47/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
34	https://longalpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.25/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	0.99/788
35	https://ly.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
36	https://many-ng.test.defo.ie/echstat.php?format=json	**0.46/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
37	https://many-ng.test.defo.ie:15443/echstat.php?format=json	**0.25/788**	1.00/788	**0.00/788**	1.00/788	**0.94/787**	1.00/788
38	https://min-ng.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
39	https://min-ng.test.defo.ie:15443/echstat.php?format=json	**0.45/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
40	https://mixedalpn-ng.test.defo.ie/echstat.php?format=json	**0.93/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
41	https://mixedalpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.45/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
42	https://mixedmode-ng.test.defo.ie/echstat.php?format=json	1.00/788	1.00/788	**0.00/788**	1.00/788	**0.03/787**	**0.00/788**

Continued on next page

Table 9: ECH interop tests from 2025-09-15 00:00:00 to 2025-10-17 19:33:40.501330.
When less than 95 percent of tests are as expected, the cell is in bold text. (Continued)

num	url	chromium	curl	firefox	golang	rustls	python
43	https://mixedmode-ng.test.defo.ie:15443/echstat.php?format=json	1.00/788	1.00/788	**0.00/788**	1.00/788	**0.03/787**	**0.00/788**
44	https://my-own.net/ech-check.php	0.99/788	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
45	https://my-own.net:8443/ech-check.php	**0.85/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
46	https://myechtest.site/	**0.00/788**	1.00/788	**0.78/788**	0.99/788	0.97/787	1.00/788
47	https://ng.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
48	https://ng.test.defo.ie:15443/echstat.php?format=json	**0.46/788**	0.99/788	1.00/788	1.00/788	**0.94/787**	1.00/788
49	https://noaddr-ng.test.defo.ie/echstat.php?format=json	1.00/788	1.00/788	1.00/788	0.99/788	0.97/787	1.00/788
50	https://noaddr-ng.test.defo.ie:15443/echstat.php?format=json	1.00/788	1.00/788	1.00/788	1.00/788	0.97/787	1.00/788
51	https://p256-ng.test.defo.ie/echstat.php?format=json	**0.00/788**	1.00/788	**0.00/788**	**0.00/788**	**0.00/787**	1.00/788
52	https://p256-ng.test.defo.ie:15443/echstat.php?format=json	**0.00/788**	1.00/788	**0.00/788**	**0.00/788**	**0.00/787**	1.00/788
53	https://pthen2-ng.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
54	https://pthen2-ng.test.defo.ie:15443/echstat.php?format=json	**0.45/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
55	https://ss.test.defo.ie/stats	0.95/788	1.00/788	1.00/788	1.00/788	**0.94/787**	0.99/788
56	https://sshrr.test.defo.ie/stats	**0.94/788**	0.99/788	1.00/788	1.00/788	**0.94/787**	0.99/788
57	https://tls-ech.dev/	**0.94/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
58	https://v1-ng.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	0.99/788	**0.94/787**	1.00/788
59	https://v1-ng.test.defo.ie:15443/echstat.php?format=json	**0.44/788**	0.99/788	1.00/788	1.00/788	**0.94/787**	1.00/788
60	https://v2-ng.test.defo.ie/echstat.php?format=json	**0.94/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
61	https://v2-ng.test.defo.ie:15443/echstat.php?format=json	**0.46/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
62	https://v3-ng.test.defo.ie/echstat.php?format=json	**0.95/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
63	https://v3-ng.test.defo.ie:15443/echstat.php?format=json	**0.46/788**	1.00/788	1.00/788	1.00/788	**0.94/787**	1.00/788
64	https://v4-ng.test.defo.ie/echstat.php?format=json	1.00/788	1.00/788	1.00/788	1.00/788	**0.91/787**	**0.00/788**
65	https://v4-ng.test.defo.ie:15443/echstat.php?format=json	1.00/788	1.00/788	1.00/788	1.00/788	**0.87/787**	**0.00/788**
66	https://withext-ng.test.defo.ie/echstat.php?format=json	**0.93/788**	1.00/788	1.00/788	0.99/788	**0.93/787**	1.00/788
67	https://withext-ng.test.defo.ie:15443/echstat.php?format=json	**0.46/788**	0.99/788	1.00/788	1.00/788	**0.94/787**	1.00/788
	Totals	**0.69/52796**	1.00/52796	**0.88/52796**	**0.93/52796**	**0.87/52729**	**0.94/52796**
	Count below 95% threshold	50	0	9	5	58	4
	Overall: **0.88/316709**						

Table 10: ECH interop tests compared.

Time	chromium	curl	firefox	golang	rustls	python	Overall
2025-01	**0.85/10653**	1.00/10653	**0.82/10653**	**0.94/10586**	**0.88/10586**	**0.91/10653**	**0.90/63784**
2025-10	**0.69/52796**	1.00/52796	**0.88/52796**	**0.93/52796**	**0.87/52729**	**0.94/52796**	**0.88/316709**

Note that the numbers in Tables 8 and 9 are not a direct measure of the "quality" of the client - they are a snapshot of counts of how often we see expected outcomes for the specific test setting, and the failure to see that outcome is sometimes due to imperfection in the test setup, or even network conditions, and not the client. Nonetheless, the numbers are useful in terms of pointing us at where we can look for improvement - to help with that, cells in the table where we see fewer than 95% of the results being as expected are in **bold**. Based on that, the set of "interesting" things arising from Tables 8 and 9 include:

- The numbers for chromium seem to have gotten notably worse, both in terms of the overall percentage of expected results and the range of test URLs where we see failures. This seems to be due to the browser's "impatience" at not waiting for HTTPS RR answers as described earlier.
- Firefox seems to have improved to a notable degree, though still getting less than 90% of expected results overall.
- Performance in terms of producing our expected results doesn't seem to have changed for the other test clients.
- A number of the notes from version 1 of this report on problems with specific tests are no

longer needed - for example the python client with the test in row 13 used to never work and now works 100% of the time. There are a number of similar examples.

- Line 23 - browsers seem to dislike `https://draft-13.esni.defo.ie:12413/`. That's due to our test result handlers and is not a failure in the clients or servers. That test case is an ECH shared-mode haproxy instance with lighttpd as the backend. As lighttpd is the backend it can't produce an HTTP response that indicates ECH worked as ECH was terminated with TLS at the haproxy instance. It's unclear how we can handle this test case when browsers don't themselves indicate ECH success or failure.
- Line 37 - it's not clear why publishing many (20) identical HTTPS RRs would work for port 443 for browsers but not off port 443.
- Line 46 - as `https://myechtest.site` returns variously malformed retry-configs, sometimes the malformed-ness is not detected by clients. Our test code does correctly interpret some situations but not all.
- The test setup for lines 51/52 calls for use of P256-HKDF-SHA256 for the KEM, a KDF of HKDF-SHA384 and the CHACHA20-POLY-1305 AEAD. It seems that some implementations (including rustls) don't support that combination as their ECH code is compiled/configured on a per-HPKE suite basis so they don't support this combination even though they support all the component algorithms. In addition, neither the ECH nor HPKE specifications nominate one or a small number of HPKE suites as mandatory-to-implement (MTI). Sadly, rather than a small set of MTI suites, HPKE defines 96 possibilities in RFC9180 [5] arguably giving rise to this interoperability issue. Whether that makes this a good or a bad test case is perhaps a matter of opinion. (The oddball selection is the author's fault.) This is discussed a little in `https://github.com/defo-project/ech-interop-report/issues/2`.)
- Lines 51/52 - curl and CPython do work with our oddball P256 combination, but Chromium, Firefox, golang and rustls (as explained above) do not support that choice.

Some additional issues encountered in validating test results are described in Appendix C.

5 Conclusions

Based on all the above, we can reach some conclusions:

- At the ECH protocol level, interoperability, ignoring broad ECH ciphersuite support, is very good. The (real) issues in getting interoperable wide deployment of ECH revolve around complexity in HTTPS RR processing.
- If deploying ECH: Keep it simple, because ECH ciphersuite handling is not at all flexible in many clients. Only publish a "singleton" ECHConfigList in one HTTPS RR and use x25519/HKDF-SHA256/AES128 for that one ECHConfig. Do not use aliasMode. If rotating ECH keys, do keep "old" private values usable at servers for a duration that is a multiple (e.g. 3x or 5x) of the duration for which the "current" ECH public value is published in the DNS. It's probably wise to also have a shorter TTL on HTTPS RR values, perhaps half of the key rotation duration or less.
- It is very non-trivial for non-browser clients to fully handle HTTPS RRs. For typical non-browser clients HTTPS RR processing requires a change from the very simple situation of using the system resolver for A/AAAA RRs to a new situation where far more complex HTTPS processing may be needed. We expect a possibly long transition period where such clients have very simplistic processing of HTTPS RRs for ECH, likely just processing a single HTTPS RR value. Again: deployers will need to keep it simple, and possibly for an extended period.

- Browsers are complex beasts. It is very likely that real browser behaviour will be better than indicated in our test results above, which all involve accessing one single URL with fully "cold" caches. However, our ECH trial [3] indicates that one can't be overly confident that ECH will always succeed even if the browser user-agent string indicates use of a browser-version that is capable of ECH success. So, one cannot be 100% confident that ECH will always be used, and always succeed, for browser users, even with the most conservative deployment. However, one can likely be confident that in such scenarios, ECH will often be used and will often succeed.

As a last word, the author is happy to use the benefit of hindsight and speculate that had a far simpler form of ECH been developed as a first version, with no options at all, and had that design been deployed and iterated upon, we would be much closer to seeing ECH widely deployed today with excellent interoperability. The lesson there however, is for protocol designers, and not for those implementing and deploying the current ECH specification, who are the main target for this document.

Acknowledgments

Thanks to the Open Technology Fund for ongoing funding of the DEfO project. (And for their patience while the IETF process for ECH takes... ages;-) Thanks in particular to the people working on DEfO who contributed to this work including: Matthew Bogner, A. Custura, Kerry Hartnett, John Hess, Iain Learmonth, Niall O'Reilly, Jochen Sprickerhof and Hans-Christoph Steiner. Thanks also to the developers of other ECH implementations (including NSS, boringssl, wolfssl, golang and rustls) for co-operating as we worked on interoperability, and to the maintainers of OpenSSL, curl, lighttpd, apache2, nginx and haproxy for ECH related discussions and PR-processing along the way.

All errors and omissions of course remain the fault of the author.

References

[1] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[2] E. Rescorla, K. Oku, N. Sullivan, and C. A. Wood, "TLS Encrypted Client Hello," Internet Engineering Task Force, Internet-Draft draft-ietf-tls-esni-25, Jun. 2025, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/25/

[3] S. Farrell and S. Singh, "ECH trial report, version 1," DEfO Project Report, Tech. Rep., 2025. [Online]. Available: https://defo.ie/ech-trial-report.pdf

[4] S. Farrell, "ECH interoperability report, version 1," DEfO Project Report, Tech. Rep., Jan. 2025. [Online]. Available: https://github.com/defo-project/ech-interop-report/blob/main/source/ech-interop-report1.pdf

[5] R. Barnes, K. Bhargavan, B. Lipp, and C. A. Wood, "Hybrid Public Key Encryption," RFC 9180, Feb. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9180

Appendices

Appendix A ECH Test Descriptions

This appendix describes the semantics of the various "*.test.defo.ie" test URLs. The other test services used are described in section 3.1. The left-most label for "*.test.defo.ie" URLs describes

the test-case and server technology, for example the "mixedmode" test-case when run against an nginx server will result in the a URL with an origin of "mixedmode-ng.test.defo.ie". For each of these test cases we also publish a TXT RR at the relevant name that has a very short description of the test.

We have a mostly-automated way of producing the set of artefacts needed for adding new tests to this list. That's a Python script that reads in a description (similar to the text below, but in JSON) and produces the set of ECH keys needed (if new ones are needed), haproxy and web-server configuration stanzas. The JSON inputs can include e.g. badly formed ECHConfigList values. The Python script also produces a bash script that can be run on an authoritative name server to clear/populate the relevant DNS records (using bind's "nsupdate"). That code and further details can be found at `https://github.com/defo-project/ech-dev-utils/tree/main/test-cases`.

Some of the text below overflows the page, but all values are published in the DNS, and the same information is visible on our IFrame test page at `https://test.defo.ie/iframe_tests_sort.html`. so hopefully this presentation is sufficient.

```
Test: "nginx server/minimal HTTPS RR"
Expected result: success
HTTPS RR:
1 . ech=AEb+DQBCqQAgACBlm7cfDx/gKuUAwRTe+Y9MExbIyuLpLcgTORIdi69uewAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
https://min-ng.test.defo.ie/echstat.php?format=json
https://min-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/nominal, HTTPS RR"
Expected result: success
HTTPS RR:
1 . ipv4hint=185.88.140.5 ech=AEb+DQBCqQAgACBlm7cfDx/gKuUAwRTe+Y9MExbIyuLpLcgTORIdi69uewAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:c6c0:0:134:2::1
https://v1-ng.test.defo.ie/echstat.php?format=json
https://v1-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/nominal, HTTPS RR"
Expected result: success
HTTPS RR:
1 . alpn="http/1.1,h2" ipv4hint=185.88.140.5 ech=AEb+DQBCqQAgACBlm7cfDx/gKuUAwRTe+Y9MExbIyuLpLcgTORIdi69uewAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:c6c0:0:134:2::1
https://v2-ng.test.defo.ie/echstat.php?format=json
https://v2-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/two RRvals for nominal, minimal, HTTPS RR"
Expected result: success
HTTPS RRs:
1 . ech=AEb+DQBCqQAgACBlm7cfDx/gKuUAwRTe+Y9MExbIyuLpLcgTORIdi69uewAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
2 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
https://v3-ng.test.defo.ie/echstat.php?format=json
https://v3-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/three RRvals, 1st bad, 2nd good, 3rd bad, HTTPS RR"
Expected result: error, but maybe arguable
HTTPS RRs:
1 . ech=AEb+DQBCmczMACBrf4D75W04lOLJ4RVtJYz71FamxDjiETWJA4KLCXeFUAAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
2 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
3 . ech=AEb+DQBCmczMACBrf4D75W04lOLJ4RVtJYz71FamxDjiETWJA4KLCXeFUAAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
https://v4-ng.test.defo.ie/echstat.php?format=json
https://v4-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/ECHConfigList with bad alg type (0xcccc) for ech kem"
Expected result: error
HTTPS RR:
1 . ech=AEb+DQBCmczMACBrf4D75W04lOLJ4RVtJYz71FamxDjiETWJA4KLCXeFUAAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
https://bk1-ng.test.defo.ie/echstat.php?format=json
https://bk1-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/zero-length ECHConfig within ECHConfigList"
Expected result: error
HTTPS RR:
1 . ech=AAT+DQAA
https://bk2-ng.test.defo.ie/echstat.php?format=json
https://bk2-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/ECHConfigList with bad ECH version (0xcccc)"
Expected result: error
HTTPS RR:
1 . ech=AEbMzABCmQAgACBrf4D75W04lOLJ4RVtJYz71FamxDjiETWJA4KLCXeFUAAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
https://bv-ng.test.defo.ie/echstat.php?format=json
https://bv-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/nominal, HTTPS RR, bad alpn"
Expected result: client-dependent
HTTPS RR:
1 . alpn="+" ech=AEb+DQBCqQAgACBlm7cfDx/gKuUAwRTe+Y9MExbIyuLpLcgTORIdi69uewAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
https://badalpn-ng.test.defo.ie/echstat.php?format=json
https://badalpn-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/HTTPS RR, with hints but no A/AAAA"
Expected result: error
HTTPS RR:
1 . ipv4hint=185.88.140.5 ech=AEb+DQBCqQAgACBlm7cfDx/gKuUAwRTe+Y9MExbIyuLpLcgTORIdi69uewAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:c6c0:0:134:2::1
https://noaddr-ng.test.defo.ie/echstat.php?format=json
https://noaddr-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/20 values in HTTPS RR"
Expected result: success
HTTPS RRs:
1 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
2 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
```

```
3 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
4 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
5 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
6 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
7 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
8 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
9 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
10 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
11 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
12 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
13 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
14 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
15 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
16 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
17 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
18 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
19 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
20 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
https://many-ng.test.defo.ie/echstat.php?format=json
https://many-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/AliasMode (0) and ServiceMode (!=0) are not allowed together"
Expected result: error, but likely ignored
HTTPS RRs:
1 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
0 test.defo.ie
https://mixedmode-ng.test.defo.ie/echstat.php?format=json
https://mixedmode-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/uses p256, hkdf-385 and chacha"
Expected result: success, but client-dependent
HTTPS RR:
1 . ipv4hint=185.88.140.5 ech=AGf+DQBjngAQAEEEFgkLkdHS+tz18PXmdYpYBZdKGxjBm+WNG9AsKNaOCj4Bradbury4KyvHEZGj6+Wls1VBe1OXierpjGiYOEriOWnAAEAAIAAwATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:
https://p256-ng.test.defo.ie/echstat.php?format=json
https://p256-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/two RRVALs one using x25519 and one with p256, same priority"
Expected result: success, but client-dependent
HTTPS RRs:
1 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
1 . ipv4hint=185.88.140.5 ech=AGf+DQBjngAQAEEEFgkLkdHS+tz18PXmdYpYBZdKGxjBm+WNG9AsKNaOCj4bmDbuF4KyvHEZGj6+Wls1VBe1OXierpjGiYOEriOWnAAEAAIAAwATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:c6
https://curves1-ng.test.defo.ie/echstat.php?format=json
https://curves1-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/two RRVALs one using x25519 (priority=1) and one with p256 (priority=2)"
Expected result: success, but client-dependent
HTTPS RRs:
1 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
2 . ipv4hint=185.88.140.5 ech=AGf+DQBjngAQAEEEFgkLkdHS+tz18PXmdYpYBZdKGxjBm+WNG9AsKNaOCj4bmDbuF4KyvHEZGj6+Wls1VBe1OXierpjGiYOEriOWnAAEAAIAAwATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:c6
https://curves2-ng.test.defo.ie/echstat.php?format=json
https://curves2-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/two RRVALs one using x25519 (priority=2) and one with p256 (priority=1)"
Expected result: success, but client-dependent
HTTPS RRs:
1 . ipv4hint=185.88.140.5 ech=AGf+DQBjngAQAEEEFgkLkdHS+tz18PXmdYpYBZdKGxjBm+WNG9AsKNaOCj4bmDbuF4KyvHEZGj6+Wls1VBe1OXierpjGiYOEriOWnAAEAAIAAwATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:c6
2 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
https://curves3-ng.test.defo.ie/echstat.php?format=json
https://curves3-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/alpn is only h2"
Expected result: success
HTTPS RR:
1 . alpn="h2" ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
https://h2alpn-ng.test.defo.ie/echstat.php?format=json
https://h2alpn-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/alpn is only http/1.1"
Expected result: success
HTTPS RR:
1 . alpn="http/1.1" ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c0:0:134:2::1
https://h1alpn-ng.test.defo.ie/echstat.php?format=json
https://h1alpn-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/alpn is http/1.1,foo,bar,bar,bom,h2"
Expected result: success
HTTPS RR:
1 . alpn="http/1.1,foo,bar,baz,bom,h2" ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsM06v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJwdWJsaWMudGVzdC5kZWZvLmllAAA= ipv6hint=2a00:c6c
https://mixedalpn-ng.test.defo.ie/echstat.php?format=json
https://mixedalpn-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/alpn is very long ending with http/1.1,h2"
Expected result: success
HTTPS RR:
1 . alpn="bogus0,bogus1,bogus2,bogus3,bogus4,bogus5,bogus6,bogus7,bogus8,bogus9,bogus10,bogus11,bogus12,bogus13,bogus14,bogus15,bogus16,bogus17,bogus18,bogus19,bogus20,bogus21,bogus22,bog
https://longalpn-ng.test.defo.ie/echstat.php?format=json
https://longalpn-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/ECHConfiglist with 2 entries a 25519 one then a p256 one (both good keys)"
Expected result: success
HTTPS RR:
1 . ech=AK3+DQBCqQAgACBlm7cfDx/gKuUAwRTe+Y9MExbIyuLpLcgTORIdi69uewAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA/gOAY54AEABBBBYJC5HROvrc9fD15nWKWAWXShsYwZvljRvQLCjWjgo+G5g27heCsrxxGRo+vlpbNVQXtTl
https://2thenp-ng.test.defo.ie/echstat.php?format=json
https://2thenp-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/ECHConfiglist with 2 entries a p256 one then a 25519 one (both good keys)"
Expected result: success
HTTPS RR:
1 . ech=AK3+DQBjngAQAEEEFgkLkdHS+tz18PXmdYpYBZdKGxjBm+WNG9AsKNaOCj4bmDbuF4KyvHEZGj6+Wls1VBe1OXierpjGiYOEriOWnAAEAAIAAwATcHVibGljLnRlc3QuZGVmby5pZQAA/gOAQqkAIAAgZZu3Hw8f4CrlAMEU3vmPTBMWyMr
https://pthen2-ng.test.defo.ie/echstat.php?format=json
https://pthen2-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server/minimal HTTPS RR but with 2 ECHConfig extensions"
Expected result: success
HTTPS RR:
1 . ech=ANf+DQDTWwAgACC76HoR3yQax/iwgeViB+vMCfbQ5wG5mT9NOUV26kc+VgAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQCRD8oARjExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTE
https://withext-ng.test.defo.ie/echstat.php?format=json
```

```
https://withext-ng.test.defo.ie:15443/echstat.php?format=json

Test: "nginx server"
Expected result: success
HTTPS RR:
1 . ech=AEb+DQBCagAgACBzLL6V07+zX4LFRuUpv7JZk3HleJFSSLc/Ut7uJK3XSAAEAAEAAQATbmctcHViLnRlc3QuZGVmby5pZQAA
https://ng.test.defo.ie/echstat.php?format=json
https://ng.test.defo.ie:15443/echstat.php?format=json

Test: "apache server"
Expected result: success
HTTPS RR:
1 . ech=AEb+DQBCjQAgACCPS3scGY9usAlr42IN/kwgOt94mhpTlJxaYn/3sXh9cgAEAAEAAQATYXAtcHViLnRlc3QuZGVmby5pZQAA
https://ap.test.defo.ie/echstat.php?format=json

Test: "lighttpd server"
Expected result: success
HTTPS RR:
1 . ech=AEb+DQBC1wAgACDhUNWChDXXfBhwQFySLx7ibTWW12pqOLLZ6+34p32gHwAEAAEAAQATbHktcHViLnRlc3QuZGVmby5pZQAA

https://ly.test.defo.ie/echstat.php?format=json
Test: "OpenSSL s_server"
Expected result: success
HTTPS RR:
1 . ech=AEb+DQBCbwAgACCo6aUdA7y16xiWdAOxAy8QYTompl9EZCOXJTgBC/EzEAAEAAEAAQATc3MtcHViLnRlc3QuZGVmby5pZQAA

https://ss.test.defo.ie/stats
Test: "OpenSSL s_server forcing HRR"
Expected result: success
HTTPS RR:
1 . ech=AEn+DQBFKgAgACA6jN+s2Mc+r+cTrJGIO4/ewXbNQ9vv4TRGM89dqL2UegAEAAEAAQAWc3NocnItcHViLnRlc3QuZGVmby5pZQAA
https://sshrr.test.defo.ie/stats
```

Appendix B Test URLs and Expected Outcomes

The list below shows our current set of test URLs and the expected outcomes from our automated tests. An expected outcome of 0 means that we expect ECH to have succeeded, a non-zero value means we expect ECH to not work, and with the expected client-specific error code.

```
url,curl_expected,firefox_expected,chrome_expected,golang_expected,rust_expected,py_expected
https://ss.test.defo.ie/stats,0,0,0,0,0,0
https://sshrr.test.defo.ie/stats,0,0,0,0,0,0
https://my-own.net/ech-check.php,0,0,0,0,0,0
https://my-own.net:8443/ech-check.php,0,0,0,0,0,0
https://defo.ie/ech-check.php,0,0,0,0,0,0
https://draft-13.esni.defo.ie:8413/stats,0,0,0,0,0,0
https://draft-13.esni.defo.ie:8414/stats,0,0,0,0,0,0
https://draft-13.esni.defo.ie:9413/,0,0,0,0,0,0
https://draft-13.esni.defo.ie:10413/,0,0,0,0,0,0
https://draft-13.esni.defo.ie:11413/,0,0,0,0,0,0
https://draft-13.esni.defo.ie:12413/,0,0,0,0,0,0
https://draft-13.esni.defo.ie:12414/,0,0,0,0,0,0
https://cloudflare-ech.com/cdn-cgi/trace,0,0,0,0,0,0
https://tls-ech.dev/,0,0,0,0,0,0
https://myechtest.site/,101,1,1,1,101,1
https://hidden.hoba.ie/,0,0,0,0,0,0
https://min-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://min-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://v1-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://v1-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://v2-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://v2-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://v3-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://v3-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://v4-ng.test.defo.ie/echstat.php?format=json,35,1,1,1,101,2
https://v4-ng.test.defo.ie:15443/echstat.php?format=json,35,1,1,1,101,2
https://bk1-ng.test.defo.ie/echstat.php?format=json,35,1,1,1,101,2
https://bk1-ng.test.defo.ie:15443/echstat.php?format=json,35,1,1,1,101,2
https://bk2-ng.test.defo.ie/echstat.php?format=json,35,1,1,1,101,2
https://bk2-ng.test.defo.ie:15443/echstat.php?format=json,35,1,1,1,101,2
https://bv-ng.test.defo.ie/echstat.php?format=json,35,1,1,1,101,2
https://bv-ng.test.defo.ie:15443/echstat.php?format=json,35,1,1,1,101,2
https://badalpn-ng.test.defo.ie/echstat.php?format=json,0,1,0,0,0,0
https://badalpn-ng.test.defo.ie:15443/echstat.php?format=json,0,1,0,0,0,0
https://noaddr-ng.test.defo.ie/echstat.php?format=json,6,1,1,1,101,1
https://noaddr-ng.test.defo.ie:15443/echstat.php?format=json,6,1,1,1,101,1
https://many-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://many-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://mixedmode-ng.test.defo.ie/echstat.php?format=json,35,1,1,1,101,1
https://mixedmode-ng.test.defo.ie:15443/echstat.php?format=json,35,1,1,1,101,1
https://p256-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://p256-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://curves1-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://curves1-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://curves2-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://curves2-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://curves3-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://curves3-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://h2alpn-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://h2alpn-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://h1alpn-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://h1alpn-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://mixedalpn-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://mixedalpn-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://longalpn-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://longalpn-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://2thenp-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://2thenp-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
```

```
https://pthen2-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://pthen2-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://withext-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://withext-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0
https://ap.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
https://ap.test.defo.ie:15444/echstat.php?format=json,0,0,0,0,0,0
https://ly.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0
```

Appendix C Recently Fixed Issues with Test Setup

C.1 Version 2

This sub-section describe issues/lessons from writing version 2 of the Interop report, some 8 months after the first version.

- Although the interop tests were running continuously over the period, they were not working continuously for the entire time. We had to do various updates of certificates but also including of the underlying operating system over the period and variously broke tests. So the first step was to get all tests working again.
- As part of that, the selenium tests were broken by the new operating system, perhaps (speculating) due to a change of path name in the virtual environment. Re-installing the python virtual environment fixed that.
- There was also a change in what firefox presents to selenium when we try to access the JSON HTTP response from our "echstats.php" - previously we got the clean JSON via an XPath expression for both chromium and firefox but now firefox produces some additional wrapping HTML that messes things up for us. We need to turn that off by setting the "devtools.jsonview.enabled" setting to "False" in the python script.
- Updating the rust tests was an adventure - we base our test on the "examples/src/bin/ech-client.rs" script from the rustls repo, but that seemed to depend on things that weren't in "standard" crates yet, so we ended up with a mixture of code from our December 2024 test script and what's in the current rustls repository.
- We spent a significant amount of time trying to establish for sure that chromium's impatience is the cause of the reduced numbers for those tests.
- We did some better analysis of the domain probe data, now that we have more of it.

C.2 Version 1

The test setup used here was developed some months before this report was written. Over that time, clients changed, and new test clients were added, so we had to fix a number of issues, and validate test results before producing this report. It may be illustrative to describe some self-caused test failures seen as we were preparing this report, and how those were resolved:

- Line 2 - `https://2thenp-ng.test.defo.ie:15443/echstat.php?format=json` works for all clients except Chromium where it never works, yet the same configuration on port 443 works 100% of the time on all clients. That configuration has an ECHConfigList with two entries - the first an x25519 key and the second a p256 key - and the same ECHConfigList value is published for both ports. The error here however is in the configuration - the targetName for the port 15443 RR is "." rather than the hostname, so Chromium here is correct and our test setup is wrong. We fixed a bug in `https://github.com/defo-project/ech-dev-utils/blob/main/test-cases/test_cases_gen.py` addressing this on 20241208.

- Line 7 and elsewhere - fixed a bug in `https://github.com/defo-project/ech-dev-utils/blob/main/scripts/selenium_test.py` where expected failures from tests using the "echstat_check()" handler were being logged as test failures when they should be logged as expected.
- Line 7 and elsewhere - fixed same bug in both `https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_py.sh` and `https://github.com/defo-project/ech-dev-utils/blob/main/scripts/smoke_ech_rs.sh` where we were assuming all non-zero returns were errors, which didn't match with the configured expected values. (Also tweaked a few of those expected values for rustls and CPython too.)
- Line 13 - our "cf_check" result handler hadn't been changed to be applied with the change in Cloudflare's ECH test URL from "crypto.cloudflare.com" to "cloudflare-ech.com" which caused erroneous failure reports.
- Line 20 and similar - our CPython test client needed a tweak to not fail for tests not using the "echstat.php" server side script - fix was pushed on 20241209
- Line 32 - somehow certbot was no longer installed on test.defo.ie so we got a cert expiry for hidden.hoba.ie - re-installed certbot to fix, which leads to some redundancy as we end up with more than one cert for some names (more or less all other name on that VM are below *.test.defo.ie)
- Line 35 - our CPython test client hits a certificate validation error for this lighttpd server - a change to the server configuration seems to fix this, the change being to send the full cert chain to clients (browsers don't need that but our other test client do).
- Lines 51 and 52: these needed an edit for the expected values, which were wrong. (That happened more than once, with the most recent fix being on 2024-12-13.)

Line numbers above refer to Table 8 as before.

On 2025-01-05 we also fixed some issues with temporary files that were being left behind. That had resulted in the "/tmp" partition filling up on 2024-12-31 leading to some tests failing and hence skewing results between those dates.

On 2025-01-13 we updated the packages on test.defo.ie to use OpenSSL-defo rather than OpenSSL-sftcd. We also updated all other packages on that VM, e.g. moving from FF 133 to FF 134. (So we need to updated the s/w section above to match.)

Appendix D Document Versions

As stated, we plan to update this report from time to time and versioning is based on the build time. This appendix notes the major changes between "published" versions.

- 2024-12-16: This was the 1st version of the report.
- 2025-01-05: A version that just extends the run time for Table 8 up to 2024-12-30 - on 2024-12-31 our chromium and firefox tests started to fail. This was due to "/tmp" being full as some temporary files generated during tests weren't always being deleted.
- 2025-09-04: Work started on an updated version of the report - we'll call the result of that work (when done) "Version 2".