ECH Interoperability Report

Stephen Farrell

Trinity College Dublin/Tolerant Networks Ltd.

stephen.farrell@cs.tcd.ie

Saturday 14th December, 2024

work-in-progress: build time 03:02 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. 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. At the time of writing, our tests indicate that things are more than 90% "OK" but are not yet more than 95% "OK".

Contents

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.

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 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 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-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 is the DEfO project's main development branch
OpenSSL-defo	Source: https://github.com/defo-project/openssl/
	Version: 3.5.0-dev fork of master, same ECH code as OpenSSL-sftcd
	ECH support: client and server, TLS only (no DTLS, no QUIC)
	this is used for DEfO project CI builds and tests
OpenSSL-feature-branch	Source: https://github.com/openssl/openssl/tree/feature/ech
	Version: 3.5.0-dev feature branch in upstream repo
	ECH support: stub APIs so far, next PR will have ECH client, then server after
	this is the OpenSSL project's target for ECH PRs, and is where ECH code will end up prior to eventually being merged to master
boringssl	Source: https://boringssl.googlesource.com/boringssl
	Version: boringssl doesn't do versions, last local build 2024-11-29
	ECH support: ECH 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 KEMs are x25519 and p256
NSS	Source: https://github.com/nss-dev/nss.git
	Version: NSS 3.108
	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: 5.7.4
	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
golang	Source: https://go.googlesource.com/go or https://github.com/golang/go/
	Version: 1.23 or later required
	ECH support: client only, server coming in 1.24 (server code is merged)
	- golang tests were just (2024-12-02) added to our smokeping tests
rustls	Source: https://github.com/rustls/rustls/
	Version: 0.23.19
	ECH support: client only
	Comment:
	- rustls tests were just (2024-12-03) added to our smokeping tests
CPython	Source: https://github.com/defo-project/cpython
	Version: CPython 3.13/3.14
	ECH support: TLS client only
	Comment:
	- CPython tests were just (2024-12-05) added to our smokeping tests
libcurl	Source: part of https://github.com/curl/curl
	Version: 8.10.0-DEV
	ECH support: TLS client only
	Comment: not well tested other than via command line curl
Conscrypt-GP	Source: https://github.com/guardianproject/conscrypt
	Version: 2.6-alpha
	ECH support: only ECH client is relevant
	Comment: This is based on boringssl and only runs on Android

2.2 Clients

Current versions of Firefox (e.g. 133.0, 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

---

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

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 131.0.6778.85 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.)

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 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 from Google in our fork 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	Source: `https://github.com/defo-project/lighttpd`
	Version: 1.4.76.81
	Comment: only shared-mode ECH
nginx	Source: `https://github.com/defo-project/nginx`
	Version: 1.27.3.4
	Comment: both shared-mode and split-mode ECH
apache2	Source: `https://github.com/defo-project/apache-httpd`
	Version: 2.5.0.ech.351
	Comment: only shared-mode ECH
haproxy	Source: `https://github.com/defo-project/haproxy`
	Version: 3.2.dev0.62
	Comment: both shared-mode and split-mode ECH
OpenSSL s_server	Source: `https://github.com/defo-project/openssl`
	Version: 3.5.0-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 of those below .ie, 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, 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, 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
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, 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, 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 infrastacrure 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
rfc5746.mywaifu.best	this ECH-enabled web page (`https://rfc5746.mywaifu.best/`) seems to have been setup as an ECH test site by someone using DEfO artefacts (nginx and documentation) but without any contact having happened between the person who set that up and any of the DEfO-project participants
	the person who set this up documented some of that at `https://ckcr4lyf.github.io/tech-notes/services/nginx/nginx-ech.html`

3.2 Operational Services

The only operational ECH service we know of is Cloudflare's deployment. However, that 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 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]

---

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

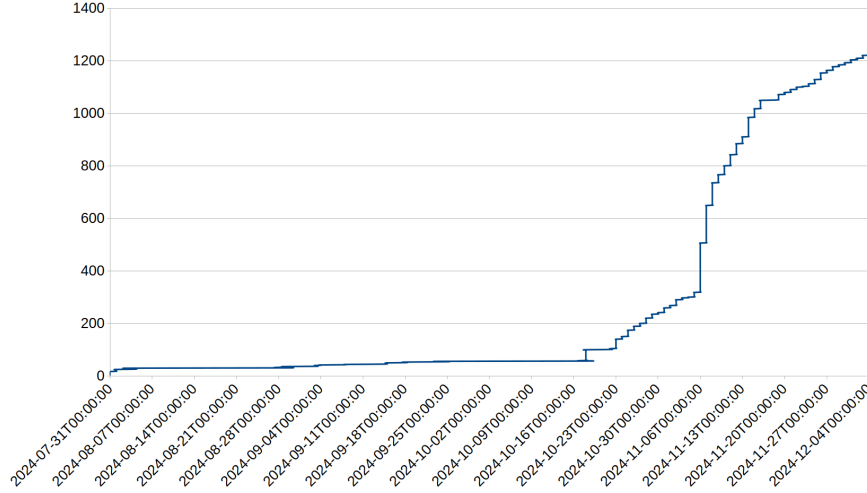[3] `https://betanews.com/2024/11/20/encrypted-client-hello-didnt-solve-censorship-but-still-may-have-a-role-to-play/`

Figure 1: Cumulative number of queries seen at `https://test.defo.ie/domainechprobe.php` versus time. The total number of queries was 1244 as of 2024-12-05.

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 (small) numbers of queries made since August 2024. We can see a noticeable uptick in accesses just-before and after Russia blocked Cloudflare's ECH service. We also see some usage (at least 15 cases) that appear to indicate some web site owners may be using this service to check whether or not they have successfully disabled ECH. (The pattern is 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.)

The 1244 queries involved 517 unique names (117 of which are under the ".ru" ccTLD). The most commonly queried name is "youtube.com" (131 times), for which ECH is not currently enabled. Some 354 names were only queried once, and 86 were queried twice.

Of the 1244 queries, 660 referred to a name that had an "ech=" in the relevant HTTPS record, referring to 298 unique names. Within the 660 HTTPS values that contain an "ech=" value, we see 276 unique ECHConfigList values (4 of which are corrupt values published for our DEfO test services). Of the 298 unique names, 271 seem to be served by Cloudflare (based on the relevant name server names), 15 relate to test services described in Table 3. At the time of writing, 9 of the names result in an NXDOMAIN or SERVFAIL response, 1 seems to correspond to a possible hobbyist site and 2 seem to (now) be parked domains or part of some advertising campaign. None of those last 3 currently publish an HTTPS record.

So at least for the names entered to our domain probe page, we can seemingly conclude that Cloudflare have the only operational service at present.

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.

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 4. We report results for the most recent 6 runs at `https://test.defo.ie/smokeping-summary.php`.

Table 4: Smokeping clients

Test Client Name	Details
Chromium	headless browser tests via Selenium
	Version: 131.0.6778.85
	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
	Version: 8.11.1-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
	Version: 133.0
	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
	Version: golang 1.23
	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
	Version: rustls 0.23.19
	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
	Version: CPython 3.13
	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

---

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

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.)

- 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 iteself, Google's tls-ech.dev, and two Cloudflare domains that support ECH. (Running that with the full set of test URLs from Table 5 is a work-in-progres.)

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 an 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 5 below shows, for each test URL and each client, the ratio of expected and total results for a recent interval. [5]

> **TODO: Update Table 5 having left things running a week or so.**

---

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

Table 5: ECH interop tests from 2024-12-10 00:00:00 to 2024-12-14 03:01:59.002608.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	**0.93/99**	1.00/100	1.00/99	0.99/99	0.99/99	1.00/99
2	https://2thenp-ng.test.defo.ie:15443/echstat.php?format=json	**0.91/99**	1.00/100	1.00/99	0.99/99	1.00/99	1.00/99
3	https://ap.test.defo.ie/echstat.php?format=json	**0.93/99**	1.00/100	1.00/99	1.00/99	1.00/99	0.99/99
4	https://ap.test.defo.ie:15444/echstat.php?format=json	**0.89/99**	1.00/100	1.00/99	1.00/99	1.00/99	1.00/99
5	https://badalpn-ng.test.defo.ie/echstat.php?format=json	**0.95/99**	1.00/100	**0.00/99**	1.00/99	1.00/99	1.00/99
6	https://badalpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.94/99**	0.99/100	**0.00/99**	1.00/99	1.00/99	1.00/99
7	https://bk1-ng.test.defo.ie/echstat.php?format=json	1.00/99	1.00/100	0.99/99	0.99/99	**0.89/99**	**0.89/99**
8	https://bk1-ng.test.defo.ie:15443/echstat.php?format=json	1.00/99	1.00/100	1.00/99	0.98/99	**0.89/99**	**0.89/99**
9	https://bk2-ng.test.defo.ie/echstat.php?format=json	1.00/99	1.00/100	1.00/99	0.99/99	**0.89/99**	**0.88/99**
10	https://bk2-ng.test.defo.ie:15443/echstat.php?format=json	1.00/99	1.00/100	1.00/99	0.99/99	**0.89/99**	**0.88/99**
11	https://bv-ng.test.defo.ie/echstat.php?format=json	1.00/99	1.00/100	1.00/99	0.98/99	**0.89/99**	**0.89/99**
12	https://bv-ng.test.defo.ie:15443/echstat.php?format=json	1.00/99	1.00/100	1.00/99	1.00/99	**0.89/99**	**0.89/99**
13	https://cloudflare-ech.com/cdn-cgi/trace	1.00/99	1.00/100	1.00/99	0.98/99	0.99/99	**0.00/99**
14	https://curves1-ng.test.defo.ie/echstat.php?format=json	0.96/99	1.00/100	1.00/99	1.00/99	**0.56/99**	0.99/99
15	https://curves1-ng.test.defo.ie:15443/echstat.php?format=json	0.96/99	1.00/100	1.00/99	1.00/99	**0.47/99**	1.00/99
16	https://curves2-ng.test.defo.ie/echstat.php?format=json	0.96/99	0.98/100	1.00/99	0.97/99	**0.49/99**	1.00/99
17	https://curves2-ng.test.defo.ie:15443/echstat.php?format=json	**0.90/99**	1.00/100	1.00/99	1.00/99	**0.46/99**	0.99/99
18	https://curves3-ng.test.defo.ie/echstat.php?format=json	**0.00/99**	1.00/100	**0.00/99**	**0.00/99**	**0.49/99**	1.00/99
19	https://curves3-ng.test.defo.ie:15443/echstat.php?format=json	**0.00/99**	1.00/100	**0.00/99**	**0.00/99**	**0.51/99**	1.00/99
20	https://defo.ie/ech-check.php	**0.93/99**	1.00/100	1.00/99	1.00/99	0.97/99	1.00/99
21	https://draft-13.esni.defo.ie:10413/	**0.92/99**	1.00/100	1.00/99	1.00/99	1.00/99	1.00/99
22	https://draft-13.esni.defo.ie:11413/	**0.92/99**	1.00/100	1.00/99	0.99/99	0.97/99	1.00/99
23	https://draft-13.esni.defo.ie:12413/	**0.00/99**	1.00/100	**0.00/99**	1.00/99	1.00/99	1.00/99
24	https://draft-13.esni.defo.ie:12414/	**0.92/99**	1.00/100	1.00/99	0.99/99	1.00/99	0.99/99
25	https://draft-13.esni.defo.ie:8413/stats	**0.90/99**	0.99/100	0.99/99	0.99/99	0.99/99	0.98/99
26	https://draft-13.esni.defo.ie:8414/stats	**0.90/99**	0.98/100	0.99/99	0.99/99	0.98/99	0.99/99
27	https://draft-13.esni.defo.ie:9413/	**0.85/99**	1.00/100	1.00/99	1.00/99	0.98/99	1.00/99
28	https://h1alpn-ng.test.defo.ie/echstat.php?format=json	0.99/99	1.00/100	0.99/99	1.00/99	1.00/99	1.00/99
29	https://h1alpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.90/99**	1.00/100	1.00/99	1.00/99	1.00/99	1.00/99
30	https://h2alpn-ng.test.defo.ie/echstat.php?format=json	**0.95/99**	1.00/100	1.00/99	0.99/99	1.00/99	0.99/99
31	https://h2alpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.92/99**	1.00/100	1.00/99	0.99/99	1.00/99	1.00/99
32	https://hidden.hoba.ie/	**0.85/99**	1.00/100	1.00/99	0.99/99	1.00/99	1.00/99
33	https://longalpn-ng.test.defo.ie/echstat.php?format=json	**0.93/99**	1.00/100	1.00/99	1.00/99	0.98/99	1.00/99
34	https://longalpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.21/99**	1.00/100	1.00/99	0.98/99	1.00/99	0.99/99
35	https://ly.test.defo.ie/echstat.php?format=json	0.96/99	1.00/100	1.00/99	0.98/99	0.99/99	1.00/99
36	https://many-ng.test.defo.ie/echstat.php?format=json	**0.95/99**	1.00/100	1.00/99	0.99/99	1.00/99	1.00/99
37	https://many-ng.test.defo.ie:15443/echstat.php?format=json	**0.21/99**	1.00/100	**0.00/99**	1.00/99	1.00/99	1.00/99
38	https://min-ng.test.defo.ie/echstat.php?format=json	**0.92/99**	1.00/100	1.00/99	1.00/99	0.99/99	1.00/99
39	https://min-ng.test.defo.ie:15443/echstat.php?format=json	**0.86/99**	1.00/100	0.99/99	1.00/99	0.99/99	0.99/99
40	https://mixedalpn-ng.test.defo.ie/echstat.php?format=json	**0.95/99**	1.00/100	1.00/99	0.99/99	1.00/99	1.00/99
41	https://mixedalpn-ng.test.defo.ie:15443/echstat.php?format=json	**0.92/99**	1.00/100	1.00/99	0.99/99	0.99/99	1.00/99
42	https://mixedmode-ng.test.defo.ie/echstat.php?format=json	1.00/99	1.00/100	**0.00/99**	1.00/99	**0.81/99**	**0.83/99**
43	https://mixedmode-ng.test.defo.ie:15443/echstat.php?format=json	1.00/99	1.00/100	**0.00/99**	1.00/99	**0.81/99**	**0.83/99**
44	https://my-own.net/ech-check.php	**0.93/99**	0.99/100	1.00/99	0.99/99	0.98/99	0.99/99
45	https://my-own.net:8443/ech-check.php	**0.74/99**	0.99/100	1.00/99	0.98/99	0.98/99	0.99/99
46	https://myechtest.site/	**0.00/99**	1.00/100	**0.60/99**	1.00/99	**0.89/99**	0.98/99
47	https://ng.test.defo.ie/echstat.php?format=json	**0.93/99**	1.00/100	1.00/99	1.00/99	0.99/99	1.00/99
48	https://ng.test.defo.ie:15443/echstat.php?format=json	**0.89/99**	1.00/100	1.00/99	1.00/99	0.98/99	1.00/99
49	https://noaddr-ng.test.defo.ie/echstat.php?format=json	1.00/99	0.98/100	1.00/99	0.99/99	**0.89/99**	0.97/99
50	https://noaddr-ng.test.defo.ie:15443/echstat.php?format=json	1.00/99	1.00/100	1.00/99	1.00/99	**0.89/99**	0.98/99
51	https://p256-ng.test.defo.ie/echstat.php?format=json	**0.93/99**	**0.07/100**	**0.94/99**	**0.00/99**	**0.00/99**	**0.89/99**
52	https://p256-ng.test.defo.ie:15443/echstat.php?format=json	**0.93/99**	**0.07/100**	**0.94/99**	**0.00/99**	**0.00/99**	**0.89/99**
53	https://pthen2-ng.test.defo.ie/echstat.php?format=json	**0.95/99**	1.00/100	1.00/99	0.99/99	0.99/99	1.00/99
54	https://pthen2-ng.test.defo.ie:15443/echstat.php?format=json	**0.87/99**	1.00/100	1.00/99	1.00/99	1.00/99	0.98/99
55	https://ss.test.defo.ie/stats	0.97/99	1.00/100	1.00/99	0.99/99	0.99/99	1.00/99
56	https://sshrr.test.defo.ie/stats	0.97/99	1.00/100	1.00/99	1.00/99	1.00/99	1.00/99
57	https://tls-ech.dev/	**0.94/99**	1.00/100	1.00/99	1.00/99	1.00/99	**0.00/99**
58	https://v1-ng.test.defo.ie/echstat.php?format=json	**0.94/99**	1.00/100	1.00/99	0.99/99	0.99/99	1.00/99

Table 5: ECH interop tests from 2024-12-10 00:00:00 to 2024-12-14 03:01:59.002608.
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
59	https://v1-ng.test.defo.ie:15443/echstat.php?format=json	**0.91/99**	1.00/100	1.00/99	1.00/99	0.99/99	1.00/99
60	https://v2-ng.test.defo.ie/echstat.php?format=json	0.96/99	1.00/100	1.00/99	1.00/99	1.00/99	1.00/99
61	https://v2-ng.test.defo.ie:15443/echstat.php?format=json	**0.92/99**	1.00/100	1.00/99	1.00/99	1.00/99	1.00/99
62	https://v3-ng.test.defo.ie/echstat.php?format=json	**0.93/99**	1.00/100	1.00/99	1.00/99	1.00/99	1.00/99
63	https://v3-ng.test.defo.ie:15443/echstat.php?format=json	**0.92/99**	1.00/100	1.00/99	0.97/99	0.99/99	0.99/99
64	https://v4-ng.test.defo.ie/echstat.php?format=json	1.00/99	0.99/100	1.00/99	1.00/99	**0.64/99**	**0.81/99**
65	https://v4-ng.test.defo.ie:15443/echstat.php?format=json	1.00/99	1.00/100	1.00/99	0.98/99	**0.61/99**	**0.82/99**
66	https://withext-ng.test.defo.ie/echstat.php?format=json	**0.95/99**	1.00/100	1.00/99	1.00/99	0.99/99	0.99/99
67	https://withext-ng.test.defo.ie:15443/echstat.php?format=json	**0.91/99**	1.00/100	1.00/99	0.99/99	1.00/99	1.00/99
	Totals	**0.86/6633**	0.97/6700	**0.87/6633**	0.93/6633	**0.89/6633**	0.94/6633
	Count below 95% threshold	46	2	11	4	21	14
	Overall: **0.91/39865**						

Note that the numbers in Table 5 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 Table 5 include:

- Column 3 - Chromium seems to currently suffer relatively more badly from the "impatiance" problem with HTTPS RR DNS answer arrival times than other clients. For other clients, unexpected test outcomes seem to be more commonly due to inherent, but easily fixed, client limitations, such as not handling some ECH ciphersuites. (Again though: this requires further investigation.)

- Lines 5 and 6 - `https://badalpn-ng.test.defo.ie/echstat.php?format=json` and it's equivalent off port 443, work fine with all clients except Firefox and never work for Firefox. The relevant DNS RR has an illegal alpn value (a '+' character). Hard to see this as other than Firefox being over-fussy, though there could of course be some good reason to reject such alpn values.

- Lines 13 and 57 - our CPython test client throws an exception for these (the Cloudflare operational, and boringssl test, servers)

```
[SSL: SSLV3_ALERT_ILLEGAL_PARAMETER] ssl/tls alert
        illegal parameter (_ssl.c:1024)
```

  This is being investigated.

- Lines 14-19 and 64/65 are roughly 50:50 likely as our rustls test client is affected by the order of HTTPS RR answers, and we get a panic when some of the badly formed ECHConfigList values are being ingested. This may be due to test client code or could be due to the rustls ECH implementation. We opened an issue `https://github.com/rustls/rustls/issues/2276` with the rust developers to explore the reasons/fix for this, and a fix was developed (`https://github.com/rustls/rustls/pull/2278`) We've yet to deploy that fix in our test environment.

- 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 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. (Between haproxy and lighttpd we should be able to engineer some HTTP response header that indicates ECH success or failure - that'd not be secure but would be fine for testing. We can have haproxy add a request header to indicate that ECH decryption worked or failed and then have lighttpd reflect that in the HTTP response. That's not been done yet.)

- 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. To be investigated.

- Line 46 - as `https://myechtest.site` returns variously malformed retry-configs, sometimes the malformedness is not detected by clients. Our test code does correctly interpret some situations but not all. To be further investigated.

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. Nonetheless, it seems at this time, that one could not 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 typically be used and will typically 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.

Acknowledgements

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-22, Sep. 2024, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/22/

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+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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+DQBCmczMACBrf4D75WO4lOLJ4RVtJYz7lFamxDjiETWJA4KLCXeFUAAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
2 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
3 . ech=AEb+DQBCmczMACBrf4D75WO4lOLJ4RVtJYz7lFamxDjiETWJA4KLCXeFUAAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
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+DQBCmczMACBrf4D75WO4lOLJ4RVtJYz7lFamxDjiETWJA4KLCXeFUAAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
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=AEbMzABCmQAgACBrf4D75WO4lOLJ4RVtJYz7lFamxDjiETWJA4KLCXeFUAAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQAA
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+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
2 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
3 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
4 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
5 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
6 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
7 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
8 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
9 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
10 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
11 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
12 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
13 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
14 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
15 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
16 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
17 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
18 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
19 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= ipv6hint=2a00:c6c0:0:134:2::1
20 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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+WNG9AsKNaOCj4bmDbuF4KyvHEZGj6+Wls1VBe1OXierpjGiYOEriOWnAAEAAIAAwATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:c6
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+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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+Wls1VBe1OXierpjGiYOEriOWnAAEAAIIAAwATcHVibGljLnRlc3QuZGVmby5pZQAA ipv6hint=2a00:c6
2 . ipv4hint=185.88.140.5 ech=AEv+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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+DQBHdAAgACCCU49qdxKOUXJPs3wlsMO6v/t42sMH5xQOL37MAd3HaAAEAAEAAQAYb3RoZXJJwdWJsaWMudGVzdC5kZWZvLmllIAAA= 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/gOAY54AEABBBBYJC5HR0vrc9fD15nWKWAWXShsYwZvljRvQLCjWjgo+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+Wls1VBe1OXierpjGiYOEriOWnAAEAAIIAAwATcHVibGljLnRlc3QuZGVmby5pZQAA/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+VgAEAAEAAQATcHVibGljLnRlc3QuZGVmby5pZQCRD8oARjExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTE
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+DQBCjQAgACCPS3scOY9usAlr42IN/kwgOt94mhpTlJxaYn/3sXh9cgAEAAEAAQATYXAtcHViLnRlc3QuZGVmby5pZQAA
https://ap.test.defo.ie/echstat.php?format=json

Test: "lighttpd server"
Expected result: success
HTTPS RR:
1 . ech=AEb+DQBC1wAgACDhUNWChDXXfBhwQFySLx7ibTWW12pq0LLZ6+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,0,0,0,0,0,0,0
https://my-own.net:8443/ech-check.php,0,0,0,0,0,0,0
https://defo.ie/ech-check.php,0,0,0,0,0,0,0
https://draft-13.esni.defo.ie:8413/stats,0,0,0,0,0,0,0
https://draft-13.esni.defo.ie:8414/stats,0,0,0,0,0,0,0
https://draft-13.esni.defo.ie:9413/,0,0,0,0,0,0,0
https://draft-13.esni.defo.ie:10413/,0,0,0,0,0,0,0
https://draft-13.esni.defo.ie:11413/,0,0,0,0,0,0,0
https://draft-13.esni.defo.ie:12413/,0,0,0,0,0,0,0
https://draft-13.esni.defo.ie:12414/,0,0,0,0,0,0,0
https://cloudflare-ech.com/cdn-cgi/trace,0,0,0,0,0,0,0
https://tls-ech.dev/,0,0,0,0,0,0,0
https://myechtest.site/,101,1,1,1,101,1
https://hidden.hoba.ie/,0,0,0,0,0,0,0
https://min-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0,0
https://min-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0,0
https://v1-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0,0
https://v1-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0,0
https://v2-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0,0
https://v2-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0,0
https://v3-ng.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0,0
https://v3-ng.test.defo.ie:15443/echstat.php?format=json,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
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,0,0,0,0,0,0,0
https://ly.test.defo.ie/echstat.php?format=json,0,0,0,0,0,0,0
```

Appendix C   Recently Fixed Issues with Test Setup

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 "ech-

stat_stat()" 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-instaled 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.)

Line numbers above refer to Table 5 as before.