ECH Interoperability Report

Stephen Farrell stephen.farrell@cs.tcd.ie

Stephen Farrell

Trinity College Dublin/Tolerant Networks Ltd.

stephen.farrell@cs.tcd.ie

work-in-progress 2024/11/30:05:06

Abstract

The abstract.

Index Terms

Encrypted Client Hello (ECH), Interoperability

I. 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.

II. SOFTWARE

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

A. Libraries

The libraries listed in Table I have some support for ECH.

TABLE I: Libraries with ECH

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) Comment: 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) Comment: 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 Comment: this is the DEfO-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 Comment: 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)

Continued on next page

TABLE I: Libraries with ECH (Continued)

Name	Details
	Comment: 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) Comment: - 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 Comment: 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 ECH support: client only, server coming in 1.24 (server code is merged) Comment: - golang tests are a work-in-progress, basic interop seems ok, we’re adding to our smokeping tests
rustls	Source: https://github.com/rustls/rustls/ Version: 1.83.0 ECH support: client basically works Comment: - rustls tests are a work-in-progress, basic interop seems ok, we’re adding to our smokeping tests
python	Source: https://github.com/defo-project/cpython Version: python 3.13/3.14 source

Continued on next page

TABLE I: Libraries with ECH (Continued)

Name	Details
	ECH support: TLS client only Comment:
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

B. Clients

1) *Browsers:* firefox, chromium, brave, vivaldi

2) *Command Line Tools:* curl

C. Servers

ours: nginx, apache, lighttpd, haproxy

III. Services

Describe services

IV. Interoperability

Describe interop

V. Conclusions

Conclude

Acknowledgements

Thanks to DEfO folks, OTF...

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/