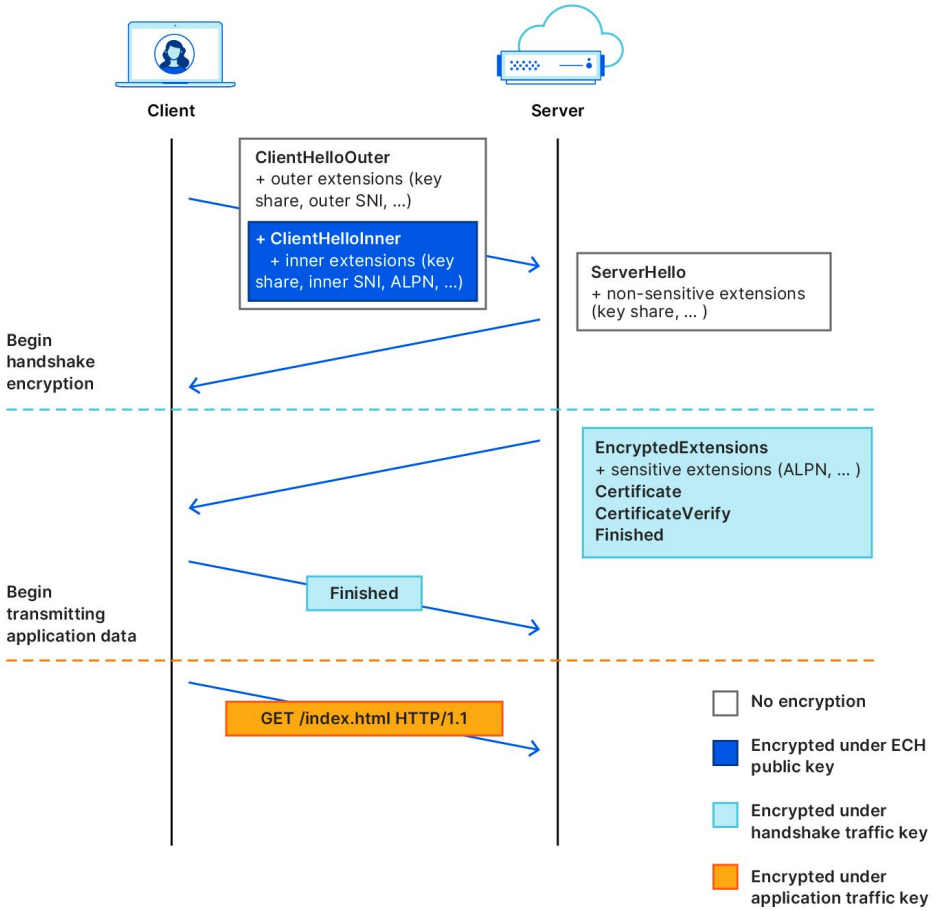
TLS Encrypted ClientHello

draft-ietf-tls-esni

IETF 118 - Prague

Reminder


Client
Server
ClientHelloOuter
+ outer extensions (key share, outer SNI, ...)
+ ClientHelloInner
+ inner extensions (key share, inner SNI, ALPN, ...)
ServerHello
+ non-sensitive extensions (key share, ...)
Begin handshake encryption
EncryptedExtensions
+ sensitive extensions (ALPN, ...)
Certificate
CertificateVerify
Finished
Finished
Begin transmitting application data
GET /index.html HTTP/1.1
No encryption
Encrypted under ECH public key
Encrypted under handshake traffic key
Encrypted under application traffic key

Deployment Status

Several implementations exist

Shipping support in Firefox, Chrome, and Cloudflare*

Today's Objective: Ship It

Several open issues exist

- #572: Public name and compatibility
- #264: Handshake-level vs record-level padding
- #441: Non-HRR acceptance signal
- #450: Grease HRR acceptance
- #427 and #567: Extensions
- #401: Complexity

Public name and compatibility

Problem: What should clients put in the public name, and how should servers enforce it

Behavior here impacts interoperability and compatibility

Language should be flexible enough to permit different server-side policies (e.g., to prevent domain fronting)

Proposed resolution: Merge #575

Padding

Problem: ECH requires *consistent* server-side visible behavior, including what's sent on the wire

Padding is necessary to avoid leaking server name

Proposed resolution: Leave as-is (close issue)

Punt padding details to separate draft, be it on that specifies padding with a handshake message, padding with an extension, or padding at the record layer

Non-HRR acceptance signal

Problem: Non-HRR acceptance signal goes in SH.random

HRR acceptance signal is in an extension, so aligning is somewhat aesthetic, but also pragmatic (avoids ossification on lack of extension and HRR possibly being broken)

Proposed resolution: Leave as-is (close issue)

Grease HRR acceptance signal

Problem: We don’t grease the HRR acceptance signal

Since we send this signal in an extension, greasing is probably useful, but so far we haven’t done it and it’s not clear the threat model requires it

Proposed resolution: Leave as-is (close issue)

Extensions

Problem: ECH extensions add complexity to the protocol

Extensions can also be used to augment protocol behavior (padding details? Public name variants? etc...)

Proposed resolution: Leave as-is (close issue)

Complexity

Problem: ECH is complex

Proposed resolution: Leave as-is (close issue)

Next steps

WGLC!