Encrypted SNI Client Hello

draft-ietf-tls-esni-05

Eric Rescorla, Kazuho Oku, Nick Sullivan, Christopher A. Wood

IETF 106 - TLS - Singapore
- OY had some problems

Ticket oracle

HRR key mismatch

Probing attacks (inconsistent cryptographic configuration)

…

https://github.com/chris-wood/encrypted-sni-model
Client \( C \) sends `ClientHello1: (g^x, Enc(K_{\text{pub}}^{\text{sni}}, (sni, nonce), g^x))` to Att. \( A \), which then sends `ClientHello1: (g^x, Enc(K_{\text{pub}}^{\text{sni}}, (sni, nonce), g^x), Ticket[\text{SNI=X}])` to Server \( S \). Server \( S \) compares SNIs and accepts the ticket if the `sni` matches \( X \). If the resumption is valid, `sni = X` if PSK accepted.
Client C

ClientHello1: \( (g^x, \text{Enc}(K_{\text{sni}}^{\text{pub}}, (\text{sni}, \text{nonce}), g^x)) \)

Att. A

HRR: (...)

ClientHello2: \( (g^x', \text{Enc}(K_{\text{sni}}^{\text{pub}}, (\text{sni}, \text{nonce}), g^{x'}) \)

Server S

ClientHello2: \( (g^a, \text{Enc}(K_{\text{sni}}^{\text{pub}}, (\text{dummysni, dummynonce}), g^a)) \)

Derive Keys:
\( ss = \text{DH}(g^a, g^y) \)
\( hk = \text{KDF}(ss + \text{secret}) \)

Second Flight: SH, Enc(ss, (EE = dummynonce, CERT = Cert, ...))

Derive Keys:
\( ss = \text{DH}(g^a, g^y) \)
\( hk = \text{KDF}(ss + ??) \)
Root Cause(s)

Lack of proper bindings:

- Between ESNI and CH contents, including resumption PSKs
- Between CH1 and CH2 in the event of HRR
- Between ESNI and remaining handshake secrets

Proposed solution:

- Encrypt (tunnel) the entire ClientHello
- Tie CH2 to CH1 for HRR
- Make handshake secrets depend on ESNI block
Probably Wrong Strawman Tunnelling Proposal

```c
struct {
    CipherSuite suite; // for ESNI
    KeyShareEntry key_share; // for ESNI
    opaque record_digest<0..2^16-1>;
    opaque ch1_binder<0..2^8-1>; // TBD
    opaque encrypted_ch<0..2^16-1>; // ClientHelloInner
} ClientEncryptedCH;
```
What is the transcript?

ESNI Accepted → ClientHelloInner

ESNI Rejected (fallback) → ClientHelloOuter

- This includes the encrypted ClientHelloInner

How does the client know what happened?

- Trial decryption
How does this help?

Entire ClientHelloInner is protected

- Prevents changing any piece

CH2 contains a hash of CH1

- Prevents mix-and-match between CH1 and CH2

Handshake secrets depend on ESNI block

- Option 1: ESNI Nonce is part of transcript, and so affects handshake keys
- Option 2: Explicitly inject ESNI-based keys into key schedule
Isn’t this really huge?

ClientHelloOuter is roughly 2X the normal size

- Includes an ordinary ClientHello
- Real problem with post-quantum key exchange

Solution: “hoist” extensions from ClientHelloInner into ClientHelloOuter

- Client removes duplicate values from ClientHelloInner when sending
- Client-facing server restores them after decrypting ESNI block
- Important they be authenticated as part of ESNI block
  - E.g., Include a hash of the value of the extensions
Open Issue: Handshake Keys

Handshake keys must be depend on ESNI block (prevent HRR oracles)

Option 1: Nonce as part of transcript

- Maybe allows unmodified back-end server
- Requires more assumptions about transcript secrecy and the nature of HKDF

Option 2: Inject a key (no nonce) derived from ESNI key into the key schedule

- Requires modifying back-end server
- Seems to rely on simpler assumptions

Proposed resolution: publish draft-06 with Option 1 while we model both. Follow up on list.
**Bundle Multiple ESNICOnfigs (PR #200)**

**Problem:**
- Currently one ESNICOnfig per HTTPSVC.
- What if the HTTPSVC record you pick has an ESNICOnfig version you don’t support?

**Solution:** Bundle all your ESNICOnfig objects into ESNICOnfigs, put that in HTTPSVC
Flatten ESNICConfig (PR #201)

Problem:

- ESNICConfig contains a list of parameters plus multiple KeyShares
- David Benjamin suggests flattening so you have one KeyShare per ESNICConfig
  - More keys → more ESNIC_configs

Upside: Implementation simplicity (?)
Downside: Duplication

Proposal: Discuss.
**Next steps**

Publish -06

Adopt HPKE for ClientHello encryption (?)

Resolve DNS extensibility PRs #200 and #201

Rename document? Encrypted ClientHello → ECHO

Start WGLC in early 2020