Authenticated ECH Config Distribution and Rotation

draft-sullivan-tls-signed-ech-updates-00

Nick Sullivan
Dennis Jackson
Alessandro Ghedini

TLS WG
IETF 124
Montreal

What's the issue?

- ECH requires clients to have a current ECHConfig before attempting an ECH handshake
- Configs are typically fetched out-of-band (DNS HTTPS/SVCB or a well-known HTTPS endpoint)
- If ECH fails due to a stale or unknown config, servers can send a new config, but clients cannot authenticate it unless the fallback "public_name" has a valid certificate
- Requiring a certificate for every potential outer SNI name limits operational flexibility and privacy strategies

Need: **a single, authenticated update mechanism**

Our proposal

Introduce an ECHConfig extension:

- In DNS: advertises public key hashes or certificate info
- In TLS Encrypted Extensions: conveys a signature over the retry config and a not_after timestamp

Two authentication modes

Raw Public Key (RPK)

Initial config lists allowed signing keys by SPKI hash (SHA-256).

Server signs updates with one of these keys.

Client checks hash match, freshness, and signature

PKIX (certificate-based)

Server signs updates with a key whose certificate chains to a trusted root, covers the public_name in SAN, and has a critical ECH-config-signing extension.

Client validates the chain and the signature.

What it looks like

```
ECHConfig (new fields in bold)
  version (u16)
  kem_id (u16)
  public_key (opaque)
  cipher_suites_len (u16)
  cipher_suites[cipher_suites_len]
  maximum_name_length (u8)
  public_name_len (u8)
  public_name[public_name_len]
  extensions_len (u16)
  extensions[extensions_len]
```

```
...
ech_auth (NEW, MUST be last)
  auth_method: "rpk" | "pkix"
  auth_info:
    if rpk:
      trusted_key_hashes[]  (SHA-256(SPKI))
      not_after (unix time)
    if pkix:
      certificate_chain  (leaf has ECH-Config-Signing critical ext)
      not_after = 0  (cert validity governs)
  signature = Sign(... ECHConfig bytes incl. ech_auth ...)
```

Benefits

- **Deployment agility**: rapid, authenticated ECH key rotation without downtime or guesswork
- **Privacy and cover names**: enables many public_name cover domains without needing certificates for each when using RPK mode
- **Unified trust**: same signed object format works in-band and with DNS SVCB/HTTPS, reducing spoofing risk by requiring valid signatures
- **Simplicity**: minimal new client state, no polling schedule, updates are delivered on demand
- **Compatibility**: integrates with TLS 1.3 and existing ECH; no changes to HPKE algorithms or core handshake flows

What we’re asking for

- Interop experiments across at independent stacks, covering both RPK and PKIX modes
- Discuss operational guidance: safe expiry windows, retry policies, and DNS integration
- Encourage early implementation and feedback to refine the mechanism and speed deployment

Authenticated ECH Config Distribution and Rotation

draft-sullivan-tls-signed-ech-updates-00

Nick Sullivan
Dennis Jackson
Alessandro Ghedini

TLS WG
IETF 124
Montreal