TLS 1.3 Extension for Certificate-based Authentication with an External Pre-Shared Key

draft-housley-tls-tls13-cert-with-extern-psk

Russ Housley

TLS WG at IETF 101

March 2018

Authentication and Key Schedule

Initial Handshake:

Authentication:	**Key Schedule Secret Inputs:**
Signature and Certificate	(EC)DHE

Subsequent Handshake:

Authentication:	**Key Schedule Secret Inputs:**
Resumption PSK	Resumption PSK + (EC)DHE
Resumption PSK	(EC)DHE

This Extension Adds Another Choice

Initial Handshake:

Authentication:	**Key Schedule Secret Inputs:**
Signature and Certificate	(EC)DHE
Signature and Certificate	External PSK + (EC)DHE

Subsequent Handshake:

Authentication:	**Key Schedule Secret Inputs:**
Resumption PSK	Resumption PSK + (EC)DHE
Resumption PSK	(EC)DHE

External PSK for Quantum Protection

- Open question whether a large-scale quantum computer is feasible, and if so, when it might happen
- If it happens, (EC)DHE becomes vulnerable
- Today: Adversary saves TLS 1.3 handshake and the associated ciphertext
- Someday: Decrypt communications when a large-scale quantum computer becomes available
- Near-term solution: Strong external PSK as an input to the TLS 1.3 key schedule
- Long-term solution: Quantum-resistant public-key cryptographic algorithms (winners of NIST competition)

Extension Overview

```
Client                                              Server

ClientHello
+ tls_cert_with_extern_psk
+ supported_groups*
+ key_share
+ signature_algorithms*
+ psk_key_exchange_modes(psk_dhe_ke)
+ pre_shared_key
                           -------->
                                                 ServerHello
                                  + tls_cert_with_extern_psk
                                                 + key_share
                                            + pre_shared_key
                                     + {EncryptedExtensions}
                                       {CertificateRequest*}
                                               {Certificate}
                                         {CertificateVerify}
                           <--------              {Finished}
{Certificate*}
{CertificateVerify*}
{Finished}                 -------->
[Application Data]         <-------->     [Application Data]
```

Extension Syntax

- The successful negotiation of the "tls_cert_with_extern_psk" extension requires the TLS 1.3 key schedule processing to include both the selected external PSK and the (EC)DHE shared secret value, and it requires the server to send the Certificate and CertificateVerify messages in the handshake
- The "tls_cert_with_extern_psk" extension will always be used along with the "key_share", "psk_key_exchange_modes", and "pre_shared_key" extensions
- The "psk_key_exchange_modes" extension will always offer psk_dhe_ke
- The "pre_shared_key" extension has obfuscated_ticket_age set to zero

- Inclusion of the extension is willingness to authenticate the server with a certificate and include an external PSK in the key schedule processing:

```
struct {
    select (Handshake.msg_type) {
        case client_hello: Empty;
        case server_hello: Empty;
    };
} CertWithExternPSK;
```

Allow Certificates with External PSK

- TLS 1.3 does not permit the server to send a CertificateRequest message when a PSK is being used. This restriction is removed when the "tls_cert_with_extern_psk" extension is negotiated, allowing the client and the server to be authenticated with a certificates
- TLS 1.3 does not permit an external PSK to be used in the same fashion as a resumption PSK, and this extension does not alter those restrictions
- A certificate MUST NOT be used with a resumption PSK