Network Working Group M. Thomson
Internet-Draft Mozilla
Updates: I-D.ietf-tls-tls13 (if May 24, 2016
approved)
Intended status: Standards Track
Expires: November 25, 2016

Cipher Suites for Negotiating Zero Round Trip (0-RTT) Transport Layer Security (TLS) with Renewed Certificate Authentication
draft-thomson-tls-0rtt-and-certs-01

Abstract

New cipher suites are defined that allow a client to use zero round trip (0-RTT) with Transport Layer Security (TLS), while also enabling the peers to renewed certificate-based authentication.

Table of Contents

1. Introduction

Transport Layer Security version 1.3 (TLS 1.3) [I-D.ietf-tls-tls13] defines a zero round trip (0-RTT) handshake mode for connections where client and server have previously communicated. In the two defined 0-RTT modes, keying material from a previous connection is used as a pre-shared key.

A 0-RTT handshake can rely entirely on the pre-shared key. These handshakes use cipher suites denoted "TLS_PSK_WITH_*". Alternative modes use the pre-shared key to authenticate the connection and secure any 0-RTT data, but then a fresh ephemeral Diffie-Hellman (or elliptic curve Diffie-Hellman) key exchange is performed. These handshakes use cipher suites denoted "TLS_DHE_PSK_WITH_*" or "TLS_ECDHE_PSK_WITH_*".

Neither of the two 0-RTT handshake modes permits either client or server to send the Certificate and CertificateVerify authentication messages. Endpoints are expected to store any authentication state with any resumption state. This means that endpoints are unable to update their understanding that a peer has continuing access to authentication keys without choosing a one round trip handshake mode and sacrificing any potential performance gained by 0-RTT.

This document defines a third mode for 0-RTT, where the pre-shared key is used to authenticate and protect 0-RTT data only. The remainder of the handshake is identical to a regular one round trip handshake with the only difference being that the resumption secret is mixed into the key schedule. This allows peers to provide fresh

proof that they control authentication keys without losing the latency advantages provided by the 0-RTT mode.

1.1. Notational Conventions

The words "MUST", "MUST NOT", "SHOULD", and "MAY" are used in this document. It's not shouting; when they are capitalized, they have the special meaning defined in [RFC2119].

2. New Cipher Suites

The following cipher suites are defined:

```
"TLS_ECDHE_PSK_ECDSA_WITH_AES_128_GCM_SHA256 = 0xXXXX
TLS_ECDHE_PSK_RSA_WITH_AES_128_GCM_SHA256 = 0xXXXX
TLS_DHE_PSK_RSA_WITH_AES_128_GCM_SHA256 = 0xXXXX
TLS_ECDHE_PSK_ECDSA_WITH_AES_256_GCM_SHA384 = 0xXXXX
TLS_ECDHE_PSK_RSA_WITH_AES_256_GCM_SHA384 = 0xXXXX
TLS_DHE_PSK_RSA_WITH_AES_256_GCM_SHA384 = 0xXXXX
TLS_ECDHE_PSK_ECDSA_WITH_CHACHA20_POLY1305_SHA256 = 0xXXXX
TLS_ECDHE_PSK_RSA_WITH_CHACHA20_POLY1305_SHA256 = 0xXXXX
TLS_DHE_PSK_RSA_WITH_CHACHA20_POLY1305_SHA256 = 0xXXXX "
```

All these cipher suites include the use of pre-shared keys and therefore permit the use of 0-RTT. These cipher suites can only be used with TLS 1.3. All include server authentication. A server MAY request client authentication by sending a CertificateRequest if it negotiates one of these cipher suites.

All the necessary cryptographic operations and the key schedule are as described in [I-D.ietf-tls-tls13].

These cipher suites use a pre-shared key for 0-RTT data, with subsequent data protected by both the PSK and an ephemeral key exchange using finite field or elliptic curve Diffie-Hellman. The pre-shared key forms the static secret (SS) and the ephemeral key exchange produces the ephemeral secret (ES). DHE_PSK_RSA suites use finite field Diffie-Hellman key exchange [DH]; ECDHE_PSK_ECDSA and ECDHE_PSK_RSA suites use elliptic curve Diffie-Hellman key exchange [X962].

These cipher suites are all authenticated using both the pre-shared key and a signature, either from an RSA certificate [RFC3447] (for DHE_PSK_RSA and ECDHE_PSK_RSA), or an ECDSA certificate (for ECDHE_PSK_ECDSA) [X962].

AES_128_GCM and AES_256_GCM use the AEAD_AES_128_GCM and AEAD_AES_256_GCM authenticated encryption defined in [RFC5116].

These are similar to the other AES-GCM modes that are described in [RFC5288]. CHACHA20_POLY1305 cipher suites use the authenticated encryption defined in [RFC7539]. Other ChaCha20-Poly1305 modes are described in [I-D.ietf-tls-chacha20-poly1305]. All authenticated encryption modes use the nonce formulation from [I-D.ietf-tls-tls13].

Suites ending with SHA256 use SHA-256 for the pseudorandom function; suites ending with SHA384 use SHA-384 [FIPS180-4].

3. Combining Certificate and PSK Authentication

TLS 1.3 forbids a server from selecting different values for many of the connection parameters when resuming a connection. Though a client might need to offer a choice in order to support a fallback to a 1-RTT handshake, a server cannot change parameters such as the selected application layer protocol [RFC7301]. Though it is theoretically possible to offer a different certificate with these cipher suites, servers MUST NOT change certificates when resuming. When resuming, clients MUST treat a change in certificate as a fatal error.

Outside of their use with 0-RTT, these cipher suites also permit the use of a combination of pre-shared key and certificate authentication. No real use case for this has been unearthed other than with the use of resumption.

The cached-info extension [I-D.ietf-tls-cached-info] can be used to reduce the size of a handshake, allowing more space for application data. Since the server certificate is not permitted to change when using 0-RTT with one of these cipher suites, this extension trivially saves a considerable amount of space.

4. Signaling Support

A TLS server that supports these cipher suites needs to indicate that it does so in the NewSessionTicket message. A new "allow_dhe_cert_resumption" value is added to TicketFlags that, when set, indicates that the server will accept resumption with cipher suites that do both (EC)DHE and certificate authentication.

```
enum {
  allow_early_data(1),
  allow_dhe_resumption(2),
  allow_psk_resumption(4),
  allow_dhe_cert_resumption(8) // new
} TicketFlags;
```

There is no IANA registry for these values, so [I-D.ietf-tls-tls13] is updated to include this value.

5. Security Considerations

Data sent after the Finished messages in the complete handshake are protected based on both the ephemeral key exchange and the pre-shared key. Learning either an (EC)DHE private key or the pre-shared key is insufficient to compromise the record protection.

The combination of pre-shared key and certificate authentication relies on peers maintaining the confidentiality of the pre-shared key for the confidentiality and integrity of 0-RTT data.

6. IANA Considerations

IANA is requested to add the following entries in the TLS Cipher Suite Registry:

```
"TLS_ECDHE_PSK_ECDSA_WITH_AES_128_GCM_SHA256 = 0xXXXX
TLS_ECDHE_PSK_RSA_WITH_AES_128_GCM_SHA256 = 0xXXXX
TLS_DHE_PSK_RSA_WITH_AES_128_GCM_SHA256 = 0xXXXX
TLS_ECDHE_PSK_ECDSA_WITH_AES_256_GCM_SHA384 = 0xXXXX
TLS_ECDHE_PSK_RSA_WITH_AES_256_GCM_SHA384 = 0xXXXX
TLS_DHE_PSK_RSA_WITH_AES_256_GCM_SHA384 = 0xXXXX
TLS_ECDHE_PSK_ECDSA_WITH_CHACHA20_POLY1305_SHA256 = 0xXXXX
TLS_ECDHE_PSK_RSA_WITH_CHACHA20_POLY1305_SHA256 = 0xXXXX
TLS_DHE_PSK_RSA_WITH_CHACHA20_POLY1305_SHA256 = 0xXXXX "
```

7. References

7.1. Normative References

[DH] Diffie, W. and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, V.IT-22 n.6 , June 1977.

[FIPS180-4] Department of Commerce, National., "NIST FIPS 180-4, Secure Hash Standard", March 2012, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

[I-D.ietf-tls-cached-info] Santesson, S. and H. Tschofenig, "Transport Layer Security (TLS) Cached Information Extension", draft-ietf-tls-cached-info-23 (work in progress), May 2016.

[I-D.ietf-tls-tls13]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-12 (work in progress), March 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, DOI 10.17487/RFC3447, February 2003, <http://www.rfc-editor.org/info/rfc3447>.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, DOI 10.17487/RFC5116, January 2008, <http://www.rfc-editor.org/info/rfc5116>.

[RFC7539] Nir, Y. and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols", RFC 7539, DOI 10.17487/RFC7539, May 2015, <http://www.rfc-editor.org/info/rfc7539>.

[X962] ANSI, "Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", ANSI X9.62, 1998.

7.2. Informative References

[I-D.ietf-tls-chacha20-poly1305]
Langley, A., Chang, W., Mavrogiannopoulos, N., Strombergson, J., and S. Josefsson, "ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)", draft-ietf-tls-chacha20-poly1305-04 (work in progress), December 2015.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[RFC5288] Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", RFC 5288, DOI 10.17487/RFC5288, August 2008, <http://www.rfc-editor.org/info/rfc5288>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <http://www.rfc-editor.org/info/rfc7301>.

Appendix A. Acknowledgments

TBD.

Author's Address

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com