TLS/DTLS 1.3 Profiles for the Internet of Things

Abstract

   This document is a companion to RFC 7925 and defines TLS/DTLS 1.3
   profiles for Internet of Things devices. It also updates RFC 7925
   with regards to the X.509 certificate profile.

Discussion Venues

   This note is to be removed before publishing as an RFC.

   Source for this draft and an issue tracker can be found at https://
   github.com/thomas-fossati/draft-tls13-iot.

Table of Contents

1. Introduction

   This document defines a profile of DTLS 1.3 [I-D.ietf-tls-dtls13]
   and TLS 1.3 [RFC8446] that offers communication security services
   for IoT applications and is reasonably implementable on many
   constrained devices. Profile thereby means that available
   configuration options and protocol extensions are utilized to best
   support the IoT environment.

   For IoT profiles using TLS/DTLS 1.2 please consult [RFC7925]. This
   document re-uses the communication pattern defined in [RFC7925] and
   makes IoT-domain specific recommendations for version 1.3 (where
   necessary).

   TLS 1.3 has been re-designed and several previously defined
   extensions are not applicable to the new version of TLS/DTLS
   anymore. This clean-up also simplifies this document. Furthermore,
   many outdated ciphersuites have been omitted from the TLS/DTLS 1.3
   specification.

1.1. Conventions and Terminology

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in
   BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all
   capitals, as shown here.

2. Credential Types

   In accordance with the recommendations in [RFC7925], a compliant
   implementation MUST implement TLS_AES_128_CCM_8_SHA256. It SHOULD
   implement TLS_CHACHA20_POLY1305_SHA256.

   Pre-shared key based authentication is integrated into the main TLS/
   DTLS 1.3 specification and has been harmonized with session
   resumption.

   A compliant implementation supporting authentication based on
   certificates and raw public keys MUST support digital signatures
   with ecdsa_secp256r1_sha256. A compliant implementation MUST support
   the key exchange with secp256r1 (NIST P-256) and SHOULD support key
   exchange with X25519.

A plain PSK-based TLS/DTLS client or server MUST implement the
following extensions:

   *supported_versions

   *cookie

   *server_name

   *pre_shared_key

   *psk_key_exchange_modes

For TLS/DTLS clients and servers implementing raw public keys and/or
certificates the guidance for mandatory-to-implement extensions
described in Section 9.2 of [RFC8446] MUST be followed.

3. Error Handling

   TLS 1.3 simplified the Alert protocol but the underlying challenge
   in an embedded context remains unchanged, namely what should an IoT
   device do when it encounters an error situation. The classical
   approach used in a desktop environment where the user is prompted is
   often not applicable with unattended devices. Hence, it is more
   important for a developer to find out from which error cases a
   device can recover from.

4. Session Resumption

   TLS 1.3 has built-in support for session resumption by utilizing
   PSK-based credentials established in an earlier exchange.

5. Compression

   TLS 1.3 does not have support for compression.

6. Perfect Forward Secrecy

   TLS 1.3 allows the use of PFS with all ciphersuites since the
   support for it is negotiated independently.

7. Keep-Alive

   The discussion in Section 10 of [RFC7925] is applicable.

8. Timeouts

   The recommendation in Section 11 of [RFC7925] is applicable. In
   particular this document RECOMMENDED to use an initial timer value

of 9 seconds with exponential back off up to no less then 60 seconds.

Question: DTLS 1.3 now offers per-record retransmission and therefore introduces much less congestion risk associated with spurious retransmissions. Hence, should we relax the 9s initial timeout?

9. Random Number Generation

The discussion in Section 12 of [RFC7925] is applicable with one exception: the ClientHello and the ServerHello messages in TLS 1.3 do not contain gmt_unix_time component anymore.

10. Server Name Indication (SNI)

This specification mandates the implementation of the SNI extension. Where privacy requirements require it, the encrypted SNI extension [I-D.ietf-tls-esni] prevents an on-path attacker to determine the domain name the client is trying to connect to. Note, however, that the extension is still at an experimental state.

11. Maximum Fragment Length Negotiation

The Maximum Fragment Length Negotiation (MFL) extension has been superseded by the Record Size Limit (RSL) extension [RFC8449]. Implementations in compliance with this specification MUST implement the RSL extension and SHOULD use it to indicate their RAM limitations.

12. Crypto Agility

The recommendations in Section 19 of [RFC7925] are applicable.

13. Key Length Recommendations

The recommendations in Section 20 of [RFC7925] are applicable.

14. 0-RTT Data

When clients and servers share a PSK, TLS/DTLS 1.3 allows clients to send data on the first flight ("early data"). This features reduces communication setup latency but requires application layer protocols to define its use with the 0-RTT data functionality.

For HTTP this functionality is described in [RFC8470]. This document specifies the application profile for CoAP, which follows the design of [RFC8470].

For a given request, the level of tolerance to replay risk is
specific to the resource it operates upon (and therefore only known
to the origin server). In general, if processing a request does not
have state-changing side effects, the consequences of replay are not
significant. The server can choose whether it will process early
data before the TLS handshake completes.

It is RECOMMENDED that origin servers allow resources to explicitly
configure whether early data is appropriate in requests.

This specification specifies the Early-Data option, which indicates
that the request has been conveyed in early data and that a client
understands the 4.25 (Too Early) status code. The semantic follows
[RFC8470].

```
+-----+---+---+---+---+------------+--------+--------+---------+---+
| No. | C | U | N | R | Name       | Format | Length | Default | E |
+-----+---+---+---+---+------------+--------+--------+---------+---+
| TBD | x |   |   |   | Early-Data | empty  | 0      | (none)  | x |
+-----+---+---+---+---+------------+--------+--------+---------+---+
```

        C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable,
        E=Encrypt and Integrity Protect (when using OSCORE)

                    Figure 1: Early-Data Option

15. Certificate Profile

This section is intended for discussing updates to the certificate
profile defined in [RFC7925]. Initial set of things to consider:

  *pathLenConstraint

Question: should also we move the ASN.1 schema from Appendix B of
[I-D.raza-ace-cbor-certificates] here and let it have it by
reference?

15.1. Compression

The compression methods defined in [I-D.ietf-tls-certificate-
compression] do not seem to deal effectively with [RFC7925] profiled
certificates: zlib compresses the example cert by 9%, but other
certificates and compression algorithms do in many cases increase
the overall size. On the other hand, [I-D.raza-ace-cbor-
certificates] provides a more efficient scheme, yielding to
compression rates higher than 50% (see Section 3 of [I-D.mattsson-
cose-cbor-cert-compress]).

Question: should we RECOMMEND CBOR compression? How is that
negotiated?

16. Security Considerations

   This entire document is about security.

17. IANA Considerations

   IANA is asked to add the Option defined in Figure 2 to the CoAP
   Option Numbers registry.

```
+--------+------------+-----------+
| Number | Name       | Reference |
+--------+------------+-----------+
| TBD    | Early-Data | RFCThis   |
+--------+------------+-----------+
```

                     Figure 2: Early-Data Option

   IANA is asked to add the Response Code defined in Figure 3 to the
   CoAP Response Code registry.

```
+--------+-------------+-----------+
| Code   | Description | Reference |
+--------+-------------+-----------+
| 4.25   | Too Early   | RFCThis   |
+--------+-------------+-----------+
```

                   Figure 3: Too Early Response Code

18. References

18.1. Normative References

   [I-D.ietf-tls-dtls13] Rescorla, E., Tschofenig, H., and N. Modadugu,
              "The Datagram Transport Layer Security (DTLS) Protocol
              Version 1.3", Work in Progress, Internet-Draft, draft-
              ietf-tls-dtls13-38, 29 May 2020, <http://www.ietf.org/
              internet-drafts/draft-ietf-tls-dtls13-38.txt>.

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

   [RFC7925] Tschofenig, H., Ed. and T. Fossati, "Transport Layer
              Security (TLS) / Datagram Transport Layer Security (DTLS)

                Profiles for the Internet of Things", RFC 7925, DOI
                10.17487/RFC7925, July 2016, <https://www.rfc-editor.org/
                info/rfc7925>.

   [RFC8174]   Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
                2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
                May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8446]   Rescorla, E., "The Transport Layer Security (TLS)
                Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446,
                August 2018, <https://www.rfc-editor.org/info/rfc8446>.

   [RFC8449]   Thomson, M., "Record Size Limit Extension for TLS", RFC
                8449, DOI 10.17487/RFC8449, August 2018, <https://
                www.rfc-editor.org/info/rfc8449>.

   [RFC8470]   Thomson, M., Nottingham, M., and W. Tarreau, "Using Early
                Data in HTTP", RFC 8470, DOI 10.17487/RFC8470, September
                2018, <https://www.rfc-editor.org/info/rfc8470>.

18.2. Informative References

   [I-D.ietf-tls-certificate-compression]
                Ghedini, A. and V. Vasiliev, "TLS Certificate
                Compression", Work in Progress, Internet-Draft, draft-
                ietf-tls-certificate-compression-10, 6 January 2020,
                <http://www.ietf.org/internet-drafts/draft-ietf-tls-
                certificate-compression-10.txt>.

   [I-D.ietf-tls-esni] Rescorla, E., Oku, K., Sullivan, N., and C.
                Wood, "TLS Encrypted Client Hello", Work in Progress,
                Internet-Draft, draft-ietf-tls-esni-07, 1 June 2020,
                <http://www.ietf.org/internet-drafts/draft-ietf-tls-
                esni-07.txt>.

   [I-D.mattsson-cose-cbor-cert-compress]
                Mattsson, J., Selander, G., Raza, S., Hoglund, J., and M.
                Furuhed, "CBOR Object Signing and Encryption (COSE):
                Headers for Carrying CBOR Compressed Certificates", Work
                in Progress, Internet-Draft, draft-mattsson-cose-cbor-
                cert-compress-00, 9 March 2020, <http://www.ietf.org/
                internet-drafts/draft-mattsson-cose-cbor-cert-
                compress-00.txt>.

   [I-D.raza-ace-cbor-certificates]
                Raza, S., Hoglund, J., Selander, G., Mattsson, J., and M.
                Furuhed, "CBOR Profile of X.509 Certificates", Work in
                Progress, Internet-Draft, draft-raza-ace-cbor-
                certificates-04, 9 March 2020, <http://www.ietf.org/
                internet-drafts/draft-raza-ace-cbor-certificates-04.txt>.

Authors' Addresses

Hannes Tschofenig
Arm Limited

Email: Hannes.Tschofenig@gmx.net

Thomas Fossati
Arm Limited

Email: Thomas.Fossati@arm.com