Workgroup: UTA
Internet-Draft:
draft-ietf-uta-tls13-iot-profile-01
Updates: 7925 (if approved)
Published: 22 February 2021
Intended Status: Standards Track
Expires: 26 August 2021
Authors: H. Tschofenig T. Fossati
Arm Limited Arm Limited

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.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 26 August 2021.

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 Furthermore, many outdated ciphersuites have been omitted from the TLS/DTLS 1.3 specification.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 contains updates and clarifications to the certificate profile defined in [RFC7925]. The content of Table 1 of [RFC7925] has been split by certificate "type" in order to clarify exactly what requirements and recommendations apply to which entity in the PKI hierarchy.

15.1. All Certificates

15.1.1. Version

Certificates MUST be of type X.509 v3.

15.1.2. Serial Number

CAs SHALL generate non-sequential Certificate serial numbers greater than zero (0) containing at least 64 bits of output from a CSPRNG (cryptographically secure pseudo-random number generator).

15.1.3. Signature

The signature MUST be ecdsa-with-SHA256 or stronger [RFC5758].

15.1.4. Issuer

Contains the DN of the issuing CA.

15.1.5. Validity

No maximum validity period is mandated. Validity values are expressed as UTCTime in notBefore and notAfter fields, as mandated in [RFC5280].

In many cases it is necessary to indicate that a certificate does not expire. This is likely to be the case for manufacturer-provisioned certificates. RFC 5280 provides a simple solution to convey the fact that a certificate has no well-defined expiration date by setting the notAfter to the GeneralizedTime value of 99991231235959Z.

Some devices might not have a reliable source of time and for those devices it is also advisable to use certificates with no expiration date and to let a device management solution manage the lifetime of all the certificates used by the device. While this approach does not utilize certificates to its widest extent, it is a solution that extends the capabilities offered by a raw public key approach.

15.1.6. subjectPublicKeyInfo

The SubjectPublicKeyInfo structure indicates the algorithm and any associated parameters for the ECC public key. This profile uses the id-ecPublicKey algorithm identifier for ECDSA signature keys, as defined and specified in [RFC5480].

15.2. Root CA Certificate

- *basicConstraints MUST be present and MUST be marked critical. The cA field MUST be set true. The pathLenConstraint field SHOULD NOT be present.
- *keyUsage MUST be present and MUST be marked critical. Bit position for keyCertSign MUST be set.
- *extendedKeyUsage MUST NOT be present.

15.3. Intermediate CA Certificate

- *basicConstraints MUST be present and MUST be marked critical. The cA field MUST be set true. The pathLenConstraint field MAY be present.
- *keyUsage MUST be present and MUST be marked critical. Bit position for keyCertSign MUST be set.
- *extendedKeyUsage MUST NOT be present.

15.4. End Entity Certificate

- *extendedKeyUsage MUST be present and contain at least one of id-kp-serverAuth or id-kp-clientAuth.
- *keyUsage MAY be present and contain one of digitalSignature or keyAgreement.
- *Domain names MUST NOT be encoded in the subject commonName, instead they MUST be encoded in a subjectAltName of type DNS-ID. Domain names MUST NOT contain wildcard (*) characters. subjectAltName MUST NOT contain multiple names.

15.4.1. Client Certificate Subject

The requirement in Section 4.4.2 of [RFC7925] to only use EUI-64 for client certificates is lifted.

If the EUI-64 format is used to identify the subject of a client certificate, it MUST be encoded in a subjectAltName of type DNS-ID as a string of the form HH-HH-HH-HH-HH-HH-HH-HH where 'H' is one of the symbols '0'-'9' or 'A'-'F'.

16. Certificate Revocation Checks

The considerations in Section 4.4.3 of [RFC7925] hold.

Since the publication of RFC 7925 the need for firmware update mechanisms has been reinforced and the work on standardizing a secure and interoperable firmware update mechanism has made substantial progress, see [I-D.ietf-suit-architecture]. RFC 7925 recommends to use a software / firmware update mechanism to provision devices with new trust anchors.

The use of device management protocols for IoT devices, which often include an onboarding or bootstrapping mechanism, has also seen considerable uptake in deployed devices and these protocols, some of which are standardized, allow provision of certificates on a regular basis. This enables a deployment model where IoT device utilize end-entity certificates with shorter lifetime making certificate revocation protocols, like OCSP and CRLs, less relevant.

Hence, instead of performing certificate revocation checks on the IoT device itself this specification recommends to delegate this task to the IoT device operator and to take the necessary action to allow IoT devices to remain operational.

16.1. Open Issues

A list of open issues can be found at https://github.com/thomas-fossati/draft-tls13-iot/issues

17. Security Considerations

This entire document is about security.

18. Acknowledgements

We would like to thank Ben Kaduk and John Mattsson.

19. 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

20. References

20.1. 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-dtls13-40, 20 January 2021, <http://www.ietf.org/internet-drafts/draft-ietf-tls-dtls13-40.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>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key

Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <https://www.rfc-editor.org/info/rfc5480>.

[RFC5758] Dang, Q., Santesson, S., Moriarty, K., Brown, D., and T. Polk, "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA", RFC 5758, DOI 10.17487/RFC5758, January 2010, <https://www.rfc-editor.org/info/rfc5758>.

[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>.

20.2. Informative References

[I-D.ietf-suit-architecture]
Moran, B., Tschofenig, H., Brown, D., and M. Meriac, "A Firmware Update Architecture for Internet of Things", Work in Progress, Internet-Draft, draft-ietf-suit-architecture-15, 17 January 2021, <http://www.ietf.org/internet-drafts/draft-ietf-suit-architecture-15.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-09, 16 December 2020, <http://www.ietf.org/internet-drafts/draft-ietf-tls-esni-09.txt>.

Authors' Addresses

Hannes Tschofenig
Arm Limited

Email: Hannes.Tschofenig@gmx.net

Thomas Fossati
Arm Limited

Email: Thomas.Fossati@arm.com