UTA
Internet-Draft
Intended status: Informational
Expires: May 7, 2020

H. Tschofenig
T. Fossati
Arm Limited

November 4, 2019

TLS/DTLS 1.3 Profiles for the Internet of Things
draft-tschofenig-uta-tls13-profile-02

Abstract

This document is a companion to RFC 7925 and defines TLS/DTLS 1.3 profiles for Internet of Things devices.

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 May 7, 2020.

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 RFC 7925 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 Furthermore, many outdated ciphersuites have been omitted from the TLS/DTLS 1.3 specification.

2. 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 RFC 2119 [RFC2119].

3. 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.

4. 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.

5. Session Resumption

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

6. Compression

TLS 1.3 does not have support for compression.

7. Perfect Forward Secrecy

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

8. Keep-Alive

The discussion in Section 10 of RFC 7925 is applicable.

9. Timeouts

The recommendation in Section 11 of RFC 7925 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.

10. Random Number Generation

The discussion in Section 12 of RFC 7925 is applicable with one exception: the ClientHello and the ServerHello messages in TLS 1.3 do not contain gmt_unix_time component anymore.

11. Server Name Indication (SNI)

This specification mandates the implementation of the SNI extension.

12. 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.

13. Crypto Agility

The recommendations in Section 19 of RFC 7925 are applicable.

14. Key Length Recommendations

The recommendations in Section 20 of RFC 7925 are applicable.

15. 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 [I-D.ietf-httpbis-replay]. This document specifies the application profile for CoAP.

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 defines a new CoAP option "timestamp", which allows the server to attach a timestamp to each CoAP message for the purpose of replay detection.

16. Security Considerations

This entire document is about security.

17. References

17.1. Normative References

[I-D.ietf-tls-dtls13]
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", draft-ietf-tls-dtls13-33 (work in progress), October 2019.

[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>.

[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>.

17.2. Informative References

[I-D.ietf-httpbis-replay]
Thomson, M., Nottingham, M., and W. Tarreau, "Using Early Data in HTTP", draft-ietf-httpbis-replay-02 (work in progress), November 2017.

[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>.

Appendix A. The Timestamp Option

The Timestamp option encodes time in standard UNIX 32-bit format (seconds since the midnight starting Jan 1, 1970, UTC, ignoring leap seconds) according to the sender's internal clock.

No.	C	U	N	R	Name	Format	Length	Default	E
TBD					Timestamp	opaque	4	(none)	x

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

Figure 1: Timestamp Option.

Authors' Addresses

Hannes Tschofenig
Arm Limited

EMail: Hannes.Tschofenig@gmx.net

Thomas Fossati
Arm Limited

EMail: Thomas.Fossati@arm.com