

UTA
Internet-Draft
Updates: [7925](#) (if approved)
Intended status: Standards Track
Expires: 28 April 2022

H. Tschofenig
T. Fossati
Arm Limited
25 October 2021

TLS/DTLS 1.3 Profiles for the Internet of Things
draft-ietf-uta-tls13-iot-profile-03

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 28 April 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Internet-Draft

TLS/DTLS 1.3 IoT Profiles

October 2021

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Conventions and Terminology	3
2.	Credential Types	3
3.	Error Handling	4
4.	Session Resumption	4
5.	Compression	4
6.	Perfect Forward Secrecy	5
7.	Keep-Alive	5
8.	Timeouts	5
9.	Random Number Generation	5
10.	Server Name Indication	5
11.	Maximum Fragment Length Negotiation	5
12.	Crypto Agility	6
13.	Key Length Recommendations	6
14.	0-RTT Data	6
15.	Certificate Profile	7
15.1.	All Certificates	7
15.1.1.	Version	7
15.1.2.	Serial Number	7
15.1.3.	Signature	7
15.1.4.	Issuer	7
15.1.5.	Validity	7
15.1.6.	subjectPublicKeyInfo	8
15.2.	Root CA Certificate	8
15.3.	Intermediate CA Certificate	8
15.4.	End Entity Certificate	8
15.4.1.	Client Certificate Subject	9
16.	Certificate Revocation Checks	9
17.	Certificate Overhead	9
18.	Ciphersuites	10
19.	Open Issues	10
20.	Security Considerations	10
21.	Acknowledgements	10
22.	IANA Considerations	11
23.	References	11

23.1.	Normative References	11
23.2.	Informative References	12
	Authors' Addresses	14

[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 Indication (SNI),
- * Pre-Shared Key,
- * PSK Key Exchange Modes, and
- * Application-Layer Protocol Negotiation (ALPN).

The SNI extension is discussed in this document and the justification for implementing and using the ALPN extension can be found in [\[I-D.ietf-uta-rfc7525bis\]](#).

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 of application data traffic, as offered by previous versions of TLS. Applications are therefore responsible for transmitting payloads that are either compressed or use a more efficient encoding otherwise.

With regards to the handshake itself, various strategies have been applied to reduce the size of the exchanged payloads. TLS and DTLS 1.3 use less overhead, depending on the type of key confirmations, when compared to previous versions of the protocol. Additionally, the work on Compact TLS (cTLS) [[I-D.ietf-tls-ctls](#)] has taken compression of the handshake a step further by utilizing out-of-band knowledge between the communication parties to reduce the amount of data to be transmitted at each individual handshake, among applying other techniques.

[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 than 60 seconds.

[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

This specification mandates the implementation of the Server Name Indication (SNI) extension. Where privacy requirements require it, the Encrypted Client Hello extension [[I-D.ietf-tls-esni](#)] prevents an on-path attacker to determine the domain name the client is trying to connect to.

Note: To avoid leaking DNS lookups from network inspection altogether further protocols are needed, including DoH [[RFC8484](#)] and DPRIVE [[RFC7858](#)] [[RFC8094](#)]. Since the Encrypted Client Hello extension requires use of Hybrid Public Key Encryption (HPKE) [[I-D.irtf-cfrg-hpke](#)] and additional protocols require further protocol exchanges and cryptographic operations, there is a certain amount of overhead associated with this privacy property.

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 feature 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 in notBefore and notAfter fields, as described in [Section 4.1.2.5 of \[RFC5280\]](#). In particular, values MUST be expressed in Greenwich Mean Time (Zulu) and MUST include seconds even where the number of seconds is zero.

Note that the validity period is defined as the period of time from notBefore through notAfter, inclusive. This means that a hypothetical certificate with a notBefore date of 9 June 2021 at 03:42:01 and a notAfter date of 7 September 2021 at 03:42:01 becomes valid at the beginning of the :01 second, and only becomes invalid at the :02 second, a period that is 90 days plus 1 second. So for a 90-day, notAfter must actually be 03:42:00.

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 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.

[17.](#) Certificate Overhead

In a public key-based key exchange, certificates and public keys are a major contributor to the size of the overall handshake. For example, in a regular TLS 1.3 handshake with minimal ECC certificates and no intermediate CA utilizing the secp256r1 curve with mutual authentication, around 40% of the entire handshake payload is consumed by the two exchanged certificates.

Hence, it is not surprising that there is a strong desire to reduce the size of certificates and certificate chains. This has lead to various standardization efforts. Here is a brief summary of what

options an implementer has to reduce the bandwidth requirements of a public key-based key exchange:

- * Use elliptic curve cryptography (ECC) instead of RSA-based certificate due to the smaller certificate size.
- * Avoid deep and complex CA hierarchies to reduce the number of intermediate CA certificates that need to be transmitted.
- * Pay attention to the amount of information conveyed inside certificates.
- * Use session resumption to reduce the number of times a full handshake is needed. Use Connection IDs [[I-D.ietf-tls-dtls-connection-id](#)], when possible, to enable long-lasting connections.
- * Use the TLS cached info [[RFC7924](#)] extension to avoid sending certificates with every full handshake.
- * Use client certificate URLs [[RFC6066](#)] instead of full certificates for clients.
- * Use certificate compression as defined in [[I-D.ietf-tls-certificate-compression](#)].
- * Use alternative certificate formats, where possible, such as raw public keys [[RFC7250](#)] or CBOR-encoded certificates [[I-D.ietf-cose-cbor-encoded-cert](#)].

The use of certificate handles, as introduced in cTLS [[I-D.ietf-tls-ctls](#)], is a form of caching or compressing certificates as well.

Whether to utilize any of the above extensions or a combination of them depends on the anticipated deployment environment, the availability of code, and the constraints imposed by already deployed infrastructure (e.g., CA infrastructure, tool support).

[18.](#) Ciphersuites

```
// As soon as the ongoing discussion around CCM_8 deprecation
// settles, provide summary and capture the consensus.
```

[19.](#) Open Issues

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

[20.](#) Security Considerations

This entire document is about security.

[21.](#) Acknowledgements

We would like to thank Ben Kaduk and John Mattsson.

[22.](#) 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

[23.](#) References

[23.1.](#) 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-](#)

[dtls13-43](#), 30 April 2021,
<<https://datatracker.ietf.org/doc/html/draft-ietf-tls-dtls13-43>>.

- [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/rfc/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/rfc/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/rfc/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/rfc/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/rfc/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/rfc/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/rfc/rfc8446>>.

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

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

23.2. Informative References

[I-D.ietf-cose-cbor-encoded-cert]

Mattsson, J. P., Selander, G., Raza, S., Höglund, J., and M. Furuheid, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, [draft-ietf-cose-cbor-encoded-cert-02](#), 12 July 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-02>>.

[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-16](#), 27 January 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-suit-architecture-16>>.

[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, <<https://datatracker.ietf.org/doc/html/draft-ietf-tls-certificate-compression-10>>.

[I-D.ietf-tls-ctls]

Rescorla, E., Barnes, R., and H. Tschofenig, "Compact TLS 1.3", Work in Progress, Internet-Draft, [draft-ietf-tls-](#)

[ctls-04](#), 25 October 2021,
<<https://datatracker.ietf.org/doc/html/draft-ietf-tls-ctls-04>>.

[I-D.ietf-tls-dtls-connection-id]

Rescorla, E., Tschofenig, H., Fossati, T., and A. Kraus, "Connection Identifiers for DTLS 1.2", Work in Progress, Internet-Draft, [draft-ietf-tls-dtls-connection-id-13](#), 22 June 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-tls-dtls-connection-id-13>>.

[I-D.ietf-tls-esni]

Rescorla, E., Oku, K., Sullivan, N., and C. A. Wood, "TLS Encrypted Client Hello", Work in Progress, Internet-Draft, [draft-ietf-tls-esni-13](#), 12 August 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-13>>.

[I-D.ietf-uta-rfc7525bis]

Sheffer, Y., Holz, R., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", Work in Progress, Internet-Draft, [draft-ietf-uta-rfc7525bis-03](#), 25 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-uta-rfc7525bis-03>>.

[I-D.irtf-cfrg-hpke]

Barnes, R. L., Bhargavan, K., Lipp, B., and C. A. Wood, "Hybrid Public Key Encryption", Work in Progress,

Internet-Draft, [draft-irtf-cfrg-hpke-12](#), 2 September 2021, <<https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-hpke-12>>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", [RFC 6066](#), DOI 10.17487/RFC6066, January 2011, <<https://www.rfc-editor.org/rfc/rfc6066>>.

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in

Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", [RFC 7250](#), DOI 10.17487/RFC7250, June 2014, <<https://www.rfc-editor.org/rfc/rfc7250>>.

- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", [RFC 7858](#), DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/rfc/rfc7858>>.
- [RFC7924] Santesson, S. and H. Tschofenig, "Transport Layer Security (TLS) Cached Information Extension", [RFC 7924](#), DOI 10.17487/RFC7924, July 2016, <<https://www.rfc-editor.org/rfc/rfc7924>>.
- [RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", [RFC 8094](#), DOI 10.17487/RFC8094, February 2017, <<https://www.rfc-editor.org/rfc/rfc8094>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", [RFC 8484](#), DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/rfc/rfc8484>>.

Authors' Addresses

Hannes Tschofenig
Arm Limited

Email: Hannes.Tschofenig@gmx.net

Thomas Fossati
Arm Limited

Email: Thomas.Fossati@arm.com