Workgroup: UTA
Internet-Draft:
draft-ietf-uta-tls13-iot-profile-08
Updates: 7925 (if approved)
Published: 22 October 2023
Intended Status: Standards Track
Expires: 24 April 2024
Authors: H. Tschofenig T. Fossati M. Richardson
Linaro Sandelman Software Works

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.

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 24 April 2024.

Table of Contents

1. Introduction

This document defines a profile of DTLS 1.3 [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 Furthermore, many outdated ciphersuites have been omitted from the TLS/DTLS 1.3 specification.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).

For use of external pre-shared keys [RFC9258] makes the following recommendation:

> Applications SHOULD provision separate PSKs for (D)TLS 1.3 and prior versions.

Where possible, the importer interface defined in [RFC9258] MUST be used for external PSKs. This ensures that external PSKs used in (D)TLS 1.3 are bound to a specific key derivation function (KDF) and hash function.

The SNI extension is discussed in this document and the justification for implementing and using the ALPN extension can be found in [RFC9325].

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 then 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 ECH (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.

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 overhead associated with this privacy feature.

Note that in industrial IoT deployments the use of ECH may not be an option because network administrators inspect DNS traffic generated by IoT devices in order to detect malicious behaviour.

Besides, to avoid leaking DNS lookups from network inspection altogether further protocols are needed, including DoH [RFC8484] and DPRIVE [RFC7858] [RFC8094]. For use of such techniques in managed networks, the reader is advised to keep up to date with the protocols defined by the Adaptive DNS Discovery (add) working group [ADD].

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

Appendix E.5 of [TLS13] establishes that:

> Application protocols MUST NOT use 0-RTT data without a profile that defines its use. That profile needs to identify which messages or interactions are safe to use with 0-RTT and how to handle the situation when the server rejects 0-RTT and falls back to 1-RTT.

At the time of writing, no such profile has been defined for CoAP [CoAP]. Therefore, 0-RTT MUST NOT be used by CoAP applications.

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.

The content is also better aligned with the IEEE 802.1AR [_8021AR] specification, which introduces the terms Initial Device Identifier (IDevID) and Locally Significant Device Identifiers (LDevIDs). IDevIDs and LDevIDs are Device Identifier (DevID) and a DevID consists of

- a private key,
- a certificate (containing the public key and the identifier certified by the certificate's issuer), and
- a certificate chain up to a trust anchor. The trust anchor is is usually the root certificate).

The IDevID is typically provisioned by a manufacturer and signed by the manufacturer CA. It is then used to obtain operational certificates, the LDevIDs, from the operator or owner of the device. Some protocols also introduce an additional hierarchy with application instance certificates, which are obtained for use with specific applications.

IDevIDs are primarily used with device onboarding or bootstrapping protocols, such as the Bootstrapping Remote Secure Key Infrastructure (BRSKI) protocol [RFC8995] or by LwM2M Bootstrap [LwM2M]. Hence, the use of IDevIDs is limited in purpose even though they have a long lifetime, or do not expire at all. While some bootstrapping protocols use TLS (and therefore make use of the IDevID as part of client authentication) there are other bootstrapping protocols that do not use TLS/DTLS for client authentication, such as FIDO Device Onboarding (FDO) [FDO]. In many cases, a profile for the certificate content is provided by those specifications. For these reasons, this specification focuses on the description of LDevIDs.

While the IEEE 802.1AR terminology is utilized in this document, this specification does not claim conformance to IEEE 802.1AR since such a compliance statement goes beyond the use of the terminology and the certificate content and would include the use of management protocols, fulfillment of certain hardware security requirements, and interfaces to access these hardware security modules. Placing these requirements on network equipment like routers may be appropriate but designers of constrained IoT devices have opted for different protocols and hardware security choices.

15.1. All Certificates

To avoid repetition, this section outlines requirements on X.509 certificates applicable to all PKI entities.

15.1.1. Version

Certificates MUST be of type X.509 v3. Note that TLS 1.3 allows to convey payloads other than X.509 certificates in the Certificate message. The description in this section only focuses on X.509 v3 certificates and leaves the description of other formats to other sections or even other specifications.

15.1.2. Serial Number

CAs MUST generate non-sequential serial numbers greater than zero (0) up to 20 octects from a cryptographically secure pseudo-random number generator. The serial number MUST be unique for each certificate issued by a given CA (i.e., the issuer name and the serial number uniquely identify a certificate).

This requirement is aligned with [RFC5280].

15.1.3. Signature

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

Note: In contrast to IEEE 802.1AR this specification does not require end entity certificates, subordinate CA certificates, and CA certificates to use the same signature algorithm. Furthermore, this specification does not utlize RSA for use with constrained IoT devices and networks.

15.1.4. Issuer

The issuer field MUST contain a non-empty distinguished name (DN) of the entity that has signed and issued the certificate in accordance to [RFC5280].

15.1.5. Validity

In IoT deployment scenarios it is often expected that the IDevIDs have no maximum validity period. For this purpose the use of a special value for the notAfter date field, the GeneralizedTime value of 99991231235959Z, is utilized. If this is done, then CA certificates and certificates of subordinate CAs cannot have a maximum validity period either. Hence, it requires careful consideration whether it is appropriate to issue IDevID certificates with no maximum validity period.

LDevID certificates are, however, issued by the operator or owner, and may be renewed at a regular interval using protocols, such as Enrollment over Secure Transport (EST) [RFC7030] or the Certificate Management Protocol (CMP) [I-D.ietf-lamps-lightweight-cmp-profile]. It is therefore RECOMMENDED to limit the lifetime of these LDevID

certificates using the notBefore and notAfter fields, as described in Section 4.1.2.5 of [RFC5280]. 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.

For devices without a reliable source of time we advise the use of a device management solution, which typically includes a certificate management protocol, to 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. Subject Public Key Info

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]. This specification assumes that devices supported one of the following algorithms:

- *id-ecPublicKey with secp256r1,
- *id-ecPublicKey with secp384r1, and
- *id-ecPublicKey with secp521r1.

There is no requirement to use CA certificates and certificates of subordinate CAs to use the same algorithm as the end-entity certificate. Certificates with longer lifetime may well use a cryptographic stronger algorithm.

15.1.7. 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 [RFC9019]. 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 thenecessary action to allow IoT devices to remain operational.

The CRL distribution points extension has been defined in RFC 5280 to identify how CRL information is obtained. The authority information access extension indicates how to access information like the online certificate status service (OCSP). Both extensions SHOULD NOT be set. If set, they MUST NOT be marked critical.

15.2. Root CA Certificate

This section outlines the requirements for root CA certificates.

15.2.1. Subject

[RFC5280] defines the Subject field as follows: "The subject field identifies the entity associated with the public key stored in the subject public key field." RFC 5280 adds "If the subject is a CA then the subject field MUST be populated with a non-empty distinguished name matching the contents of the issuer field in all certificates issued by the subject CA."

The Subject field MUST be present and MUST contain the commonName, the organizationName, and the countryName attribute and MAY contain an organizationalUnitName attribute.

15.2.2. Authority Key Identifier

Section 4.2.1.1 of [RFC5280] defines the Authority Key Identifier as follows: "The authority key identifier extension provides a means of identifying the public key corresponding to the private key used to sign a certificate. This extension is used where an issuer has multiple signing keys."

The Authority Key Identifier extension MAY be omitted. If it is set, it MUST NOT be marked critical, and MUST contain the subjectKeyIdentifier of this certificate.

[Editor's Note: Do we need to set the Authority Key Identifier in the CA certificate?]

15.2.3. Subject Key Identifier

Section 4.2.1.2 of [RFC5280] defines the Subject Key Identifier as follows: "The subject key identifier extension provides a means of identifying certificates that contain a particular public key."

The Subject Key Identifier extension MUST be set, MUST NOT be marked critical, and MUST contain the key identifier of the public key contained in the subject public key info field.

[Editor's Note: Do we need to set the Subject Key Identifier in the CA certificate?]

15.2.4. Key Usage

[RFC5280] defines the key usage field as follows: "The key usage extension defines the purpose (e.g., encipherment, signature, certificate signing) of the key contained in the certificate."

The Key Usage extension SHOULD be set. If it is set, it MUST be marked critical and the keyCertSign or cRLSign purposes MUST be set. Additional key usages MAY be set depending on the intended usage of the public key.

[Editor's Note: Should we harden the requirement to "The Key Usage extension MUST be set.]

[RFC5280] defines the extended key usage as follows: "This extension indicates one or more purposes for which the certified public key may be used, in addition to or in place of the basic purposes indicated in the key usage extension."

This extendedKeyUsage extension MUST NOT be set.

15.2.5. Basic Constraints

[RFC5280] states that "The Basic Constraints extension identifies whether the subject of the certificate is a CA and the maximum depth of valid certification paths that include this certificate. The cA boolean indicates whether the certified public key may be used to verify certificate signatures."

For the pathLenConstraint RFC 5280 makes further statements:

* "The pathLenConstraint field is meaningful only if the cA boolean is asserted and the key usage extension, if present, asserts the keyCertSign bit. In this case, it gives the maximum number of non-self-issued intermediate certificates that may follow this certificate in a valid certification path."

*"A pathLenConstraint of zero indicates that no non-self-issued intermediate CA certificates may follow in a valid certification path."
*"Where pathLenConstraint does not appear, no limit is imposed."
*"Conforming CAs MUST include this extension in all CA certificates that contain public keys used to validate digital signatures on certificates and MUST mark the extension as critical in such certificates."

The Basic Constraints extension MUST be set, MUST be marked critical, the cA flag MUST be set to true and the pathLenConstraint MUST be omitted.

[Editor's Note: Should we soften the requirement to: "The pathLenConstraint field SHOULD NOT be present."]

15.3. Subordinate CA Certificate

This section outlines the requirements for subordinate CA certificates.

15.3.1. Subject

The Subject field MUST be set and MUST contain the commonName, the organizationName, and the countryName attribute and MAY contain an organizationalUnitName attribute.

15.3.2. Authority Key Identifier

The Authority Key Identifier extension MUST be set, MUST NOT be marked critical, and MUST contain the subjectKeyIdentifier of the CA that issued this certificate.

15.3.3. Subject Key Identifier

The Subject Key Identifier extension MUST be set, MUST NOT be marked critical, and MUST contain the key identifier of the public key contained in the subject public key info field.

15.3.4. Key Usage

The Key Usage extension MUST be set, MUST be marked critical, the keyCertSign or cRLSign purposes MUST be set, and the digitalSignature purpose SHOULD be set.

The extendedKeyUsage extensed MAY be set depending on the intended usage of the public key.

[Editor's Note: Should we harden the requirement to "The extendedKeyUsage MUST NOT be present."]

15.3.5. Basic Constraints

The Basic Constraints extension MUST be set, MUST be marked critical, the cA flag MUST be set to true and the pathLenConstraint MUST be set to 0.

[Editor's Note: Should we soften the requriement to "The pathLenConstraint field MAY be present."]

15.3.6. CRL Distribution Point

The CRL Distribution Point extension SHOULD NOT be set. If it is set, it MUST NOT be marked critical and MUST identify the CRL relevant for this certificate.

15.3.7. Authority Information Access

The Authority Information Access extension SHOULD NOT be set. If it is set, it MUST NOT be marked critical and MUST identify the location of the certificate of the CA that issued this certificate and the location it provides an online certificate status service (OCSP).

15.4. End Entity Certificate

This section outlines the requirements for end entity certificates.

15.4.1. Subject

The requirement in Section 4.4.2 of [RFC7925] to only use EUI-64 for end entity certificates as a Subject name is lifted.

Two fields are typically used to encode a device identifer, namely the Subject and the subjectAltName fields. Protocol specifications tend to offer recommendations what identifiers to use and the deployment situation is fragmented.

The Subject field MAY include a unique device serial number. If the serial number is included, it MUST be encoded in the serialNumber attribute.

[RFC5280] defines: "The subject alternative name extension allows identities to be bound to the subject of the certificate. These identities may be included in addition to or in place of the identity in the subject field of the certificate."

The subject alternative name extension MAY be set. If it is set, it MUST NOT be marked critical, except when the subject DN contains an empty sequence.

If the EUI-64 format is used to identify the subject of an end entity 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'.

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. The subjectAltName MUST NOT contain multiple names.

Note: The IEEE 802.1AR recommends to encode information about a Trusted Platform Module (TPM), if present, in the HardwareModuleName. This specification does not follow this recommendation.

15.4.2. Authority Key Identifier

The Authority Key Identifier extension MUST be set, MUST NOT be marked critical, and MUST contain the subjectKeyIdentifier of the CA that issued this certificate.

15.4.3. Subject Key Identifier

The Subject Key Identifier SHOULD NOT be included in end-entity certificates. If it is included, then the Subject Key Identifier extension MUST NOT be marked critical, and MUST contain the key identifier of the public key contained in the subject public key info field.

[Editor's Note: Should we harden the requirement and state: "The Subject Key Identifier MUST NOT be included in end-entity certificates."]

15.4.4. Key Usage

The key usage extension MUST be set and MUST be marked as critical. For signature verification keys the digitialSignature key usage purpose MUST be specified. Other key usages are set according to the intended usage of the key.

If enrollment of new certificates uses server-side key generation, encrypted delivery of the private key is required. In such cases the key usage keyEncipherment or keyAgreement MUST be set because the encrypted delivery of the newly generated key involves encryption or agreement of a symmetric key. On-device key generation is, however, the preferred approach.

The extendedKeyUsage MUST be present and contain at least one of id-kp-serverAuth or id-kp-clientAuth.

16. 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 subordinate 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. Below is a brief summary of what options an implementer has to reduce the bandwidth requirements of a public key-based key exchange. Note that many of the standardized extensions are not readily available in TLS/DTLS stacks since optimizations typically get implemented last.

- Use elliptic curve cryptography (ECC) instead of RSA-based certificate due to the smaller certificate size. This document recommends the use of elliptic curve cryptography only.
- Avoid deep and complex CA hierarchies to reduce the number of subordinate CA certificates that need to be transmitted and processed. See [I-D.irtf-t2trg-taxonomy-manufacturer-anchors] for a discussion about CA hierarchies.
- 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 [RFC9146], 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. When applications perform TLS client authentication via DNS-Based Authentication of Named Entities (DANE) TLSA records then the [I-D.ietf-dance-tls-clientid] specification may be used to reduce the packets on the wire. Note: The term "TLSA" does not stand for anything; it is just the name of the RRtype, as explained in [RFC668].
- Use certificate compression as defined in [RFC8879].
- 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).

17. Ciphersuites

Section 4.5.3 of [DTLS13] flags AES-CCM with 8-octet authentication tags (CCM_8) as unsuitable for general use with DTLS. In fact, due to its low integrity limits (i.e., a high sensitivity to forgeries), endpoints that negotiate ciphersuites based on such AEAD are susceptible to a trivial DoS. (See also Section 5.3 and 5.4 of [I-D.irtf-cfrg-aead-limits] for further discussion on this topic, as well as references to the analysis supporting these conclusions.)

Specifically, [DTLS13] warns that:

> "TLS_AES_128_CCM_8_SHA256 MUST NOT be used in DTLS without additional
> safeguards against forgery. Implementations MUST set usage limits for
> AEAD_AES_128_CCM_8 based on an understanding of any additional forgery
> protections that are used."

Since all the ciphersuites mandated by [RFC7925] and [CoAP] are based on CCM_8, there is no stand-by ciphersuite to use for applications that want to avoid the security and availability risks associated with CCM_8 while retaining interoperability with the rest of the ecosystem.

In order to ameliorate the situation, this document RECOMMENDS that implementations support the following two ciphersuites:

- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- *TLS_ECDHE_ECDSA_WITH_AES_128_CCM

and offer them as their first choice. These ciphersuites provide confidentiality and integrity limits that are considered acceptable in the most general settings. For the details on the exact bounds of both ciphersuites see Section 4.5.3 of [DTLS13]. Note that the GCM-based ciphersuite offers superior interoperability with cloud services at the cost of a slight increase in the wire and peak RAM footprints.

When the GCM-based ciphersuite is used with TLS 1.2, the recommendations in Section 6.2.1 of [RFC9325] related to deterministic nonce generation apply. In addition, the integrity limits on key usage detailed in Section 4.4 of [RFC9325] also apply.

18. Fault Attacks on Deterministic Signature Schemes

A number of passive side-channel attacks as well as active fault-injection attacks (e.g., [Ambrose2017]) have been demonstrated that allow a malicious third party to gain information about the signing

key if a fully deterministic signature scheme (e.g., [RFC6979] ECDSA or EdDSA [RFC8032]) is used.

Most of these attacks assume physical access to the device and are therefore especially relevant to smart cards as well as IoT deployments with poor or non-existent physical security.

In this security model, it is recommended to combine both randomness and determinism, for example, as described in [I-D.mattsson-cfrg-det-sigs-with-noise].

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. IANA Considerations

This document makes no requests to IANA.

22. References

22.1. Normative References

[DTLS13] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/rfc/rfc9147>.

[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>.

[RFC9258] Benjamin, D. and C. A. Wood, "Importing External Pre-Shared Keys (PSKs) for TLS 1.3", RFC 9258, DOI 10.17487/RFC9258, July 2022, <https://www.rfc-editor.org/rfc/rfc9258>.

[TLS13] 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>.

22.2. Informative References

[ADD] IETF, "Adaptive DNS Discovery (add) Working Group", September 2023, <https://datatracker.ietf.org/wg/add/about/>.

[Ambrose2017] Ambrose, C., Bos, J. W., Fay, B., Joye, M., Lochter, M., and B. Murray, "Differential Attacks on Deterministic Signatures", 2017, <https://eprint.iacr.org/2017/975.pdf>.

[CoAP] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/

RFC7252, June 2014, <https://www.rfc-editor.org/rfc/rfc7252>.

[FDO] FIDO Alliance, "FIDO Device Onboard Specification 1.1", April 2022, <https://fidoalliance.org/specifications/download-iot-specifications/>.

[I-D.ietf-cose-cbor-encoded-cert] Mattsson, J. P., Selander, G., Raza, S., Höglund, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-ietf-cose-cbor-encoded-cert-07, 20 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-07>.

[I-D.ietf-dance-tls-clientid] Huque, S. and V. Dukhovni, "TLS Extension for DANE Client Identity", Work in Progress, Internet-Draft, draft-ietf-dance-tls-clientid-02, 12 May 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-dance-tls-clientid-02>.

[I-D.ietf-lamps-lightweight-cmp-profile] Brockhaus, H., von Oheimb, D., and S. Fries, "Lightweight Certificate Management Protocol (CMP) Profile", Work in Progress, Internet-Draft, draft-ietf-lamps-lightweight-cmp-profile-21, 17 February 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-lightweight-cmp-profile-21>.

[I-D.ietf-tls-ctls] Rescorla, E., Barnes, R., Tschofenig, H., and B. M. Schwartz, "Compact TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-ctls-08, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-ctls-08>.

[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-17, 9 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-17>.

[I-D.irtf-cfrg-aead-limits] Günther, F., Thomson, M., and C. A. Wood, "Usage Limits on AEAD Algorithms", Work in Progress, Internet-Draft, draft-irtf-cfrg-aead-limits-07, 31 May 2023, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-aead-limits-07>.

[I-D.irtf-cfrg-hpke] Barnes, R., 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>.

[I-D.irtf-t2trg-taxonomy-manufacturer-anchors] Richardson, M., "A Taxonomy of operational security considerations for manufacturer installed keys and Trust Anchors", Work in Progress, Internet-Draft, draft-irtf-t2trg-taxonomy-manufacturer-anchors-02, 6 August 2023, <https://datatracker.ietf.org/doc/html/draft-irtf-t2trg-taxonomy-manufacturer-anchors-02>.

[I-D.mattsson-cfrg-det-sigs-with-noise] Mattsson, J. P., Thormarker, E., and S. Ruohomaa, "Deterministic ECDSA and EdDSA Signatures with Additional Randomness", Work in Progress, Internet-Draft, draft-mattsson-cfrg-det-sigs-with-noise-04, 15 February 2022, <https://datatracker.ietf.org/doc/html/draft-mattsson-cfrg-det-sigs-with-noise-04>.

[LwM2M] OMA SpecWorks, "Lightweight Machine to Machine (LwM2M) V. 1.2.1 Technical Specification: Transport Bindings", December 2022, <https://openmobilealliance.org/release/LightweightM2M/V1_2_1-20221209-A/>.

[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>.

[RFC668] "Not Issued", RFC 668, <https://www.rfc-editor.org/rfc/rfc668>.

[RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August 2013, <https://www.rfc-editor.org/rfc/rfc6979>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/rfc/rfc7030>.

[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>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/rfc/rfc8032>.

[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>.

[RFC8879] Ghedini, A. and V. Vasiliev, "TLS Certificate Compression", RFC 8879, DOI 10.17487/RFC8879, December 2020, <https://www.rfc-editor.org/rfc/rfc8879>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

[RFC9019] Moran, B., Tschofenig, H., Brown, D., and M. Meriac, "A Firmware Update Architecture for Internet of Things", RFC 9019, DOI 10.17487/RFC9019, April 2021, <https://www.rfc-editor.org/rfc/rfc9019>.

[RFC9146] Rescorla, E., Ed., Tschofenig, H., Ed., Fossati, T., and A. Kraus, "Connection Identifier for DTLS 1.2", RFC 9146, DOI 10.17487/RFC9146, March 2022, <https://www.rfc-editor.org/rfc/rfc9146>.

[RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/rfc/rfc9325>.

[_8021AR] IEEE, "IEEE Standard for Local and metropolitan area networks – Secure Device Identity, IEEE 802.1AR-2018", August 2018, <https://1.ieee802.org/security/802-1ar>.

Acknowledgments

We would like to thank Ben Kaduk, Hendrik Brockhaus, John Mattsson and Michael Richardson.

Authors' Addresses

Hannes Tschofenig

Email: Hannes.Tschofenig@gmx.net

Thomas Fossati
Linaro

Email: Thomas.Fossati@linaro.com

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca