

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: June 4, 2021

S. Raza  
J. Hoeglund  
RISE AB  
G. Selander  
J. Mattsson  
Ericsson AB  
M. Furuheid  
Nexus Group  
December 01, 2020

CBOR Encoding of X.509 Certificates (CBOR Certificates)  
draft-mattsson-cose-cbor-cert-compress-05

## Abstract

This document specifies a CBOR encoding of X.509 certificates. The resulting certificates are called CBOR Certificates. The CBOR encoding supports a large subset of [RFC 5280](#), while at the same time significantly reduces the size of certificates compatible with [RFC 7925](#) and IEEE 802.1AR (DevIDs). When used to re-encode DER encoded X.509 certificates, the CBOR encoding can in many cases reduce the size of [RFC 7925](#) profiled certificates with over 50%. The CBOR encoding can also be used to encode "natively signed" CBOR certificates, which does not require re-encoding for the signature to be verified. The document also specifies COSE headers as well as a TLS certificate type for CBOR certificates.

## 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 June 4, 2021.

## Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

## Copyright Notice

Copyright (c) 2020 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. 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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Notational Conventions . . . . .	<a href="#">4</a>
<a href="#">3.</a>	CBOR Encoding . . . . .	<a href="#">4</a>
<a href="#">3.1.</a>	Message Fields . . . . .	<a href="#">5</a>
<a href="#">3.2.</a>	Encoding of Extensions . . . . .	<a href="#">8</a>
<a href="#">4.</a>	Compliance Requirements for Constrained IoT . . . . .	<a href="#">10</a>
<a href="#">5.</a>	Legacy Considerations . . . . .	<a href="#">11</a>
<a href="#">6.</a>	Expected Certificate Sizes . . . . .	<a href="#">11</a>
<a href="#">7.</a>	Natively Signed CBOR Certificates . . . . .	<a href="#">12</a>
<a href="#">8.</a>	Security Considerations . . . . .	<a href="#">12</a>
<a href="#">9.</a>	IANA Considerations . . . . .	<a href="#">13</a>
<a href="#">9.1.</a>	CBOR Certificate Types Registry . . . . .	<a href="#">13</a>
<a href="#">9.2.</a>	CBOR Attribute Type Registry . . . . .	<a href="#">14</a>
<a href="#">9.3.</a>	CBOR Extension Type Registry . . . . .	<a href="#">14</a>
<a href="#">9.4.</a>	CBOR Extended Key Usage Registry . . . . .	<a href="#">15</a>
<a href="#">9.5.</a>	CBOR General Name Registry . . . . .	<a href="#">16</a>
<a href="#">9.6.</a>	CBOR Certificate Signature Algorithms Registry . . . . .	<a href="#">16</a>
<a href="#">9.7.</a>	CBOR Certificate Public Key Algorithms Registry . . . . .	<a href="#">17</a>
<a href="#">9.8.</a>	COSE Header Parameters Registry . . . . .	<a href="#">18</a>
<a href="#">9.9.</a>	TLS Certificate Types Registry . . . . .	<a href="#">19</a>
<a href="#">10.</a>	References . . . . .	<a href="#">19</a>
<a href="#">10.1.</a>	Normative References . . . . .	<a href="#">19</a>
<a href="#">10.2.</a>	Informative References . . . . .	<a href="#">20</a>
<a href="#">Appendix A.</a>	Example CBOR Certificates . . . . .	<a href="#">21</a>
<a href="#">A.1.</a>	Example <a href="#">RFC 7925</a> profiled X.509 Certificate . . . . .	<a href="#">21</a>
<a href="#">A.2.</a>	Example HTTPPS X.509 Certificate . . . . .	<a href="#">25</a>

<a href="#">Appendix B. X.509 Certificate Profile, ASN.1 . . . . .</a>	<a href="#">27</a>
<a href="#">Acknowledgments . . . . .</a>	<a href="#">29</a>
<a href="#">Authors' Addresses . . . . .</a>	<a href="#">29</a>

---

## Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

### [1.](#) Introduction

One of the challenges with deploying a Public Key Infrastructure (PKI) for the Internet of Things (IoT) is the size and parsing of X.509 public key certificates [\[RFC5280\]](#), since those are not optimized for constrained environments [\[RFC7228\]](#). More compact certificate representations are desirable. Due to the current PKI usage of DER encoded X.509 certificates, keeping compatibility with DER encoded X.509 is necessary at least for a transition period. However, the use of a more compact encoding with the Concise Binary Object Representation (CBOR) [\[RFC7049\]](#) reduces the certificate size significantly which has known performance benefits in terms of decreased communication overhead, power consumption, latency, storage, etc.

CBOR is a data format designed for small code size and small message size. CBOR builds on the JSON data model but extends it by e.g. encoding binary data directly without base64 conversion. In addition to the binary CBOR encoding, CBOR also has a diagnostic notation that is readable and editable by humans. The Concise Data Definition Language (CDDL) [\[RFC8610\]](#) provides a way to express structures for protocol messages and APIs that use CBOR. [\[RFC8610\]](#) also extends the diagnostic notation.

CBOR data items are encoded to or decoded from byte strings using a type-length-value encoding scheme, where the three highest order bits of the initial byte contain information about the major type. CBOR supports several different types of data items, in addition to integers (int, uint), simple values (e.g. null), byte strings (bstr), and text strings (tstr), CBOR also supports arrays [] of data items, maps {} of pairs of data items, and sequences of data items. For a complete specification and examples, see [\[RFC7049\]](#), [\[RFC8610\]](#), and [\[RFC8742\]](#).

[RFC 7925](#) [\[RFC7925\]](#) and IEEE 802.1AR [\[IEEE-802.1AR\]](#) specify certificate profiles for Internet of Things deployments which can be

applied to lightweight certificate based authentication with, e.g., TLS [RFC8446], DTLS [I-D.ietf-tls-dtls13], COSE [RFC8152], EDHOC [I-D.ietf-lake-edhoc] or Compact TLS 1.3 [I-D.ietf-tls-ctls]. This document specifies a CBOR encoding based on [X.509-IoT], which can support large parts of [RFC5280]. The encoding support all [RFC7925] and IEEE 802.1AR [IEEE-802.1AR] profiled X.509 certificates. Two variants are defined using the same CBOR encoding and differing only in what is being signed:

- o An invertible CBOR re-encoding of DER encoded X.509 certificates [RFC5280], which can be reversed to obtain the original DER encoded X.509 certificate.

---

Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

- o Natively signed CBOR certificates, which further optimizes the performance in constrained environments but is not backwards compatible with [RFC5280], see [Section 7](#).

This document specifies COSE headers for use of the CBOR certificates with COSE, see [Section 9.8](#). The document also specifies a TLS certificate type for use of the CBOR certificates with TLS (with or without additional TLS certificate compression), see [Section 9.9](#).

## [2](#). Notational Conventions

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.

This specification makes use of the terminology in [RFC5280], [RFC7049], [RFC7228], and [RFC8610].

## [3](#). CBOR Encoding

This section specifies the content and encoding for CBOR certificates, with the overall objective to produce a very compact representation supporting large parts of [RFC5280], and everything in [RFC7925] and [IEEE-802.1AR]. In the CBOR encoding, static fields are elided, elliptic curve points and time values are compressed, OID are replaced with short integers, and redundant encoding is removed. Combining these different components reduces the certificate size

significantly, which is not possible with general purpose compressions algorithms, see Figure 2.

The CBOR certificate can be either a CBOR re-encoding of a DER encoded X.509 certificate, in which case the signature is calculated on the DER encoded ASN.1 data in the X.509 certificate, or a natively signed CBOR certificate, in which case the signature is calculated directly on the CBOR encoded data (see [Section 7](#)). In both cases the certificate content is adhering to the restrictions given by [\[RFC5280\]](#). The re-encoding is known to work with DER encoded certificates but might work with other canonical encodings. The re-encoding does not work for BER encoded certificates.

In the encoding described below, the order of elements in arrays are always encoded in the same order as the elements or the corresponding SEQUENCE or SET in the DER encoding.

### [3.1.](#) Message Fields

The X.509 fields and their CBOR encodings are listed below, and used in the definition of CBOR Certificates, see Figure 1.

CBOR certificates are defined in terms of DER encoded [\[RFC5280\]](#) X.509 certificates:

- o version. The 'version' field is encoded in the 'cborCertificateType' CBOR int. The field 'cborCertificateType' also indicates the type of the CBOR certificate. Currently, the type can be a natively signed CBOR certificate following X.509 v3 (cborCertificateType = 0) or a CBOR re-encoded X.509 v3 DER certificate (cborCertificateType = 1), see [Section 9.1](#).
- o serialNumber. The 'serialNumber' INTEGER value field is encoded as the unwrapped CBOR positive bignum (~biguint) 'certificateSerialNumber'. Any leading 0x00 byte (to indicate that the number is not negative) is therefore omitted.
- o signature. The 'signature' field is always the same as the 'signatureAlgorithm' field and therefore omitted from the CBOR

encoding.

- o issuer. In the general case, the sequence of 'RelativeDistinguishedName' is encoded as a CBOR array of CBOR arrays of Attributes, where each Attribute type and value is encoded as a (CBOR int, CBOR text string) pair. Each AttributeType is encoded as a CBOR int (see Figure 4). The AttributeType id-emailAddress is always an IA5String. For the other AttributeTypes, the sign is used to represent the character string type; positive for utf8String, negative for printableString. In natively signed CBOR certificates all strings are utf8String and the sign has no meaning. The string types teletexString, universalString, and bmpString are not supported. If Name contains a single Attribute containing an utf8String encoded 'common name' it is encoded as a CBOR text string. If the text string contains an EUI-64 of the form "HH-HH-HH-HH-HH-HH-HH-HH" where 'H' is one of the symbol '0'-'9' or 'A'-'F' it is encoded as a CBOR byte string of length 8 instead. EUI-64 mapped from a 48-bit MAC address (i.e. of the form "HH-HH-HH-HH-HH-HH-HH-HH") is encoded as a CBOR byte string of length 6.
- o validity. The 'notBefore' and 'notAfter' fields are encoded as unwrapped CBOR epoch-based date/time (~time) where the tag content is an unsigned integer. In POSIX time, leap seconds are ignored, with a leap second having the same POSIX time as the second before it. Compression of X.509 certificates with the time 23:59:60 UTC

is therefore not supported. Note that [RFC 5280](#) mandates encoding of dates through the year 2049 as UTCTime, and later dates as GeneralizedTime. The value "99991231235959Z" (no expiration date) is encoded as CBOR null.

- o subject. The 'subject' is encoded exactly like issuer.
- o subjectPublicKeyInfo. The 'algorithm' field is encoded as the CBOR int 'subjectPublicKeyAlgorithm' (see [Section 9.7](#)) or a CBOR OID tag [[I-D.ietf-cbor-tags-oid](#)]. Algorithms with parameters are not supported except id-ecPublicKey with named curves and the RSA algorithms that use parameters = NULL. For id-ecPublicKey the namedCurve parameter is encoded in the CBOR int. The 'subjectPublicKey' BIT STRING value field is encoded as a CBOR byte string. This specification assumes the BIT STRING has zero

unused bits and the unused bits byte is omitted. Uncompressed public keys of type `id-ecPublicKey` are point compressed as defined in Section 2.3.3 of [\[SECG\]](#). If a DER encoded certificate with a point compressed public key of type `id-ecPublicKey` is CBOR encoded, the octets `0xfe` and `0xfd` are used instead of `0x02` and `0x03` in the CBOR encoding to represent even and odd y-coordinate, respectively.

- o `issuerUniqueID`. Not supported.
- o `subjectUniqueID`. Not supported.
- o `extensions`. The 'extensions' field is encoded as a CBOR array where each extension is encoded as either a CBOR int (see [Section 9.3](#)) followed by an optional CBOR item of any type or a CBOR OID tag [\[I-D.ietf-cbor-tags-oid\]](#) followed by a CBOR bool encoding 'critical' and the DER encoded value of the 'extnValue' encoded as a CBOR byte string. If the array contains exactly two ints and the absolute value of the first int is 2, the array is omitted and the extensions is encoded as a single CBOR int with the absolute value of the second int and the sign of the first int. Extensions are encoded as specified in [Section 3.2](#). The extensions mandated to be supported by [\[RFC7925\]](#) and [\[IEEE-802.1AR\]](#) are given special treatment. An omitted 'extensions' field is encoded as an empty CBOR array.
- o `signatureAlgorithm`. The 'signatureAlgorithm' field is encoded as a CBOR int (see [Section 9.6](#)) or a CBOR OID tag [\[I-D.ietf-cbor-tags-oid\]](#). Algorithms with parameters are not supported except RSA algorithms that use parameters = NULL.
- o `signatureValue`. The 'signatureValue' BIT STRING value field is encoded as the CBOR byte string `issuerSignatureValue`. This

specification assumes the BIT STRING has zero unused bits and the unused bits byte is omitted. ECDSA signatures are given special treatment. For ECDSA signatures the SEQUENCE and INTEGER type and length fields are omitted and the two INTEGER value fields are padded to the fixed length  $L = \text{ceil}(\log_2(n) / 8)$ , where  $n$  is the size of the largest prime-order subgroup. For `secp256r1`, `secp384r1`, and `secp521r1`,  $L$  is 32, 48, and 66 respectively. For natively signed CBOR certificates the signatureValue is calculated

over the CBOR sequence TBSCertificate.

The following Concise Data Definition Language (CDDL) defines CBORCertificate and TBSCertificate, which are encoded as CBOR Sequences [[RFC8742](#)]. The member names therefore only have documentary value.



```

CBORCertificate = [
    TBSCertificate,
    issuerSignatureValue : bytes,
]

TBSCertificate = (
    cborCertificateType : int,
    certificateSerialNumber : CertificateSerialNumber,
    issuer : Name,
    validityNotBefore : Time,
    validityNotAfter : Time / null,
    subject : Name,
    subjectPublicKeyAlgorithm : Algorithm,
    subjectPublicKey : bytes,
    extensions : Extensions,
    issuerSignatureAlgorithm : Algorithm,
)

CertificateSerialNumber = ~biguint

Name = [ * [ + Attribute ] ] / text / bytes

Attribute = ( attributeType : int, attributeValue : text )

Time = ~time

Algorithm = int / oid

Extensions = [ * Extension ] / int

Extension = (
    extensionID : int / oid,
    ? critical : bool,          ; present if and only if extensionID is an oid
    extensionValue : any,      ; type known from extensionID
)

```

Figure 1: CDDL for CBORCertificate.

### [3.2.](#) Encoding of Extensions

EDITOR'S NOTE: The current specification encodes many common extensions with a DER encoded byte string. It should be discussed if more or all commonly active extensions should be natively encoded with CBOR. Would a specific CBOR encoding have to be specified for each extension or can a general CBOR encoding that apply to all remaining extensions be specified?

This section details the encoding of the 'extensions' field. The 'extensions' field is encoded as a CBOR array where each extensionID is encoded as either a CBOR int or a CBOR OID tag. If 'extensionID' is encoded as an int (see [Section 9.3](#)), the sign is used to encode if the extension is critical and the 'critical' field is omitted. Critical extensions are encoded with a positive sign and non-critical extensions are encoded with a negative sign.

The 'extnValue' OCTET STREAM value field is encoded as the CBOR byte string 'extensionValue' except for the extensions specified below. The 'extensionValue' for the extensions mandated to be supported by [\[RFC7925\]](#) and [\[IEEE-802.1AR\]](#) are encoded as follows:

- o basicConstraints. If 'cA' = false then extensionValue = -2, if 'cA' = true and 'pathLenConstraint' is not present then extensionValue = -1, and if 'cA' = true and 'pathLenConstraint' is present then extensionValue = pathLenConstraint.
- o keyUsage. The 'KeyUsage' BIT STRING is interpreted as an unsigned integer n in network byte order and encoded as a CBOR int.
- o extKeyUsage. extensionValue is encoded as an array of CBOR ints (see [Section 9.4](#)) or CBOR OID tags [\[I-D.ietf-cbor-tags-oid\]](#) where each int or OID tag encodes a key usage purpose. If the array contains a single int, the array is omitted.

ExtValueEKU = [ + int / oid ] / int

- o subjectAltName. extensionValue is encoded as an array of (int, any) pairs where each pair encodes a general name (see [Section 9.5](#)). If subjectAltName contains exactly one dNSName, the array and the int are omitted and extensionValue is the dNSName encoded as a CBOR text string. In addition to the general names defined in [\[RFC5280\]](#), the hardwareModuleName type of otherName has been given its own int due to its mandatory use in IEEE 802.1AR. When 'otherName + hardwareModuleName' is used, then [ oid, bytes ] is used to identify the pair ( hwType, hwSerialEntries ) directly as specified in [\[RFC4108\]](#).

GeneralNames = [ + GeneralName ] / text

GeneralName = ( GeneralNameType : int, GeneralNameValue : any )

- o authorityKeyIdentifier. extensionValue is encoded as an array where the value of the 'keyIdentifier' is encoded as a CBOR byte string, 'GeneralNames' is encoded like in subjectAltName, and 'AuthorityCertSerialNumber' is encoded as ~biguint exactly like

certificateSerialNumber.

## Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

KeyIdentifier = bytes

- o subjectKeyIdentifier. extensionValue is the value of the 'keyIdentifier' field encoded as a CBOR byte string.
- o cRLDistributionPoints. If the cRLDistributionPoints is a sequence of DistributionPointName, it is encoded like subjectAltName, with the difference that if cRLDistributionPoints contains exactly one uniformResourceIdentifier, the the array and the int are omitted and extensionValue is the uniformResourceIdentifier encoded as a CBOR text string.
- o authorityInfoAccess. If authorityInfoAccess consist of only uniformResourceIdentifiers it is encoded as an array of uris.

ExtValueAIA = [ + ( ocsp : 1 // caIssuers : 2 , uri : text ) ]

### 3.2.1. Example Encoding of Extensions

The examples below use values from [Section 9.3](#), [Section 9.4](#), and [Section 9.5](#):

- o A critical basicConstraints ('cA' = true) without pathLenConstraint is encoded as the two CBOR ints -1, -1.
- o A non-critical keyUsage with digitalSignature and keyAgreement asserted is encoded as the two CBOR ints 2, 17 ( $2^0 + 2^4 = 17$ ).
- o A non-critical extKeyUsage containing id-kp-codeSigning and id-kp-OCSPSigning is encoded as the CBOR int 3 followed by the CBOR array [ 3, 6 ].
- o A non-critical subjectAltName containing only the dNSName example.com is encoded as the CBOR int 4 followed by the CBOR text string "example.com".

Thus, the extension field of a certificate containing all of the above extensions in the given order would be encoded as the CBOR array [ -1, -1, 2, 17, 3, [ 3, 6 ], 4, "example.com" ].

#### [4.](#) Compliance Requirements for Constrained IoT

For general purpose applications, the normative requirements of [\[RFC5280\]](#) applies. This section describes the mandatory to implement algorithms and OIDs for constrained IoT application; the values of the OIDs including certificate fields and extensions, time format, attributes in distinguished names, etc.

Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

TODO: Write this section

#### [5.](#) Legacy Considerations

CBOR certificates can be deployed with legacy X.509 certificates and CA infrastructure. In order to verify the signature, the CBOR certificate is used to recreate the original X.509 data structure to be able to verify the signature.

For protocols like TLS/DTLS 1.2, where the handshake is sent unencrypted, the actual encoding and compression can be done at different locations depending on the deployment setting. For example, the mapping between CBOR certificate and standard X.509 certificate can take place in a 6LoWPAN border gateway which allows the server side to stay unmodified. This case gives the advantage of the low overhead of a CBOR certificate over a constrained wireless links. The conversion to X.509 within an IoT device will incur a computational overhead, however, measured in energy this is likely to be negligible compared to the reduced communication overhead.

For the setting with constrained server and server-only authentication, the server only needs to be provisioned with the CBOR certificate and does not perform the conversion to X.509. This option is viable when client authentication can be asserted by other means.

For protocols like IKEv2, TLS/DTLS 1.3, and EDHOC, where certificates are encrypted, the proposed encoding needs to be done fully end-to-end, through adding the encoding/decoding functionality to the server.

#### [6.](#) Expected Certificate Sizes

The CBOR encoding of the sample certificate given in [Appendix A](#) results in the numbers shown in Figure 2. After [\[RFC7925\]](#) profiling, most duplicated information has been removed, and the remaining text strings are minimal in size. Therefore, the further size reduction reached with general compression mechanisms will be small, mainly corresponding to making the ASN.1 encoding more compact. The zlib number was calculated with zlib-flate.

```
zlib-flate -compress < cert.der > cert.compressed
```

Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

	<a href="#">RFC 7925</a>	zlib	CBOR Certificate
Certificate Size	314	295	138

Figure 2: Comparing Sizes of Certificates (bytes)

7. Natively Signed CBOR Certificates

The difference between CBOR encoded X.509 certificate and natively signed CBOR certificate is that the signature is calculated over the CBOR encoding of the CBOR sequence TBSCertificate rather than the DER encoded ASN.1 data. This removes entirely the need for ASN.1 DER and base64 encodings which reduces the processing in the authenticating devices, and avoids known complexities and security issues with these encodings.

Natively signed CBOR certificates can be applied in devices that are only required to authenticate to natively signed CBOR certificate compatible servers. This is not a major restriction for many IoT deployments, where the parties issuing and verifying certificates can be a restricted ecosystem which not necessarily involves public CAs.

CBOR encoded X.509 certificates provides an intermediate step between

[RFC7925] or [IEEE-802.1AR] profiled X.509 certificates and natively signed CBOR certificates: An implementation of CBOR encoded X.509 certificates contains both the CBOR encoding of the X.509 certificate and the signature operations sufficient for natively signed CBOR certificates.

The natively signed approach based on DER encoded X.509 certificates described in this document has also other benefits. For example, a CA can use existing ASN.1 machinery to create a DER encoded certificate, the DER encoded certificate can then be transformed to CBOR before signing.

## 8. Security Considerations

The CBOR profiling of X.509 certificates does not change the security assumptions needed when deploying standard X.509 certificates but decreases the number of fields transmitted, which reduces the risk for implementation errors.

The use of natively signed CBOR certificates removes the need for ASN.1 encoding, which is a rich source of security vulnerabilities.

Conversion between the certificate formats can be made in constant time to reduce risk of information leakage through side channels.

The mechanism in this draft does not reveal any additional information compared to X.509. Because of difference in size, it will be possible to detect that this profile is used. The gateway solution described in [Section 5](#) requires unencrypted certificates and is not recommended.

## 9. IANA Considerations

This document creates several new registries under the new heading "CBOR Certificate". For all items, the 'Reference' field points to this document.

The expert reviewers for the registries defined in this document are expected to ensure that the usage solves a valid use case that could not be solved better in a different way, that it is not going to

duplicate one that is already registered, and that the registered point is likely to be used in deployments. They are furthermore expected to check the clarity of purpose and use of the requested code points. Experts should take into account the expected usage of entries when approving point assignment, and the length of the encoded value should be weighed against the number of code points left that encode to that size and the size of device it will be used on. Values in the interval [-24, 23] have a 1 byte encodings, other values in the interval [-256, 255] have a 2 byte encodings, and the remaning values in the interval [-65536, 65535] have 3 byte encodings.

9.1. CBOR Certificate Types Registry

IANA has created a new registry titled "CBOR Certificate Types" under the new heading "CBOR Certificate". For values in the interval [-24, 23] the registration procedure is "IETF Review" and "Expert Review". For all other values the registration procedure is "Expert Review". The columns of the registry are Value, Description, and Reference, where Value is an integer, and the other columns are text strings. The initial contents of the registry are:

Value	Description
0	Natively Signed CBOR Certificate following X.509 v3
1	CBOR re-encoding of X.509 v3 Certificate

Figure 3: CBOR Certificate Types

9.2. CBOR Attribute Type Registry

IANA has created a new registry titled "CBOR Attribute Type Registry" under the new heading "CBOR Certificate". The columns of the registry are Value, X.509 Attribute Type, and Reference, where Value is an integer, and the other columns are text strings. Only non-negative values can be registered. For values in the interval [0, 23] the registration procedure is "IETF Review" and "Expert Review". For all other values the registration procedure is "Expert Review". The initial contents of the registry are:

Value	X.509 Attribute Type
0	id-emailAddress
1	id-at-commonName
2	id-at-surname
3	id-at-serialNumber
4	id-at-countryName
5	id-at-localityName
6	id-at-stateOrProvinceName
7	id-at-organizationName
8	id-at-organizationalUnitName
9	id-at-title
10	id-at-givenName
11	id-at-initials
12	id-at-generationQualifier
13	id-at-dnQualifier
14	id-at-pseudonym
15	id-at-organizationIdentifier

Figure 4: CBOR Attribute Type Registry

### 9.3. CBOR Extension Type Registry

IANA has created a new registry titled "CBOR Extension Type Registry" under the new heading "CBOR Certificate". The columns of the registry are Value, X.509 Extension Type, and Reference, where Value is an integer, and the other columns are text strings. Only positive values can be registered. For values in the interval [1, 23] the registration procedure is "IETF Review" and "Expert Review". For all other values the registration procedure is "Expert Review". The initial contents of the registry are:

Value	X.509 Extension Type	extensionValue
1	id-ce-basicConstraints	int



2	id-ce-keyUsage	int
3	id-ce-extKeyUsage	ExtValueEKU
4	id-ce-subjectAltName	GeneralNames
5	id-ce-authorityKeyIdentifier	ExtValueAKI
6	id-ce-subjectKeyIdentifier	KeyIdentifier
7	id-ce-certificatePolicies	bytes
8	id-ce-cRLDistributionPoints	GeneralNames
9	id-pe-authorityInfoAccess	ExtValueAIA
10	SCT List (1.3.6.1.4.1.11129.2.4.2)	bytes
248	id-ce-nameConstraints	bytes
249	id-ce-policyConstraints	bytes
250	id-ce-inhibitAnyPolicy	bytes
251	id-ce-policyMappings	bytes
252	id-ce-issuerAltName	GeneralNames
253	id-ce-subjectDirectoryAttributes	bytes
254	id-ce-freshestCRL	bytes
255	id-pe-subjectInfoAccess	bytes

Figure 5: CBOR Extension Type Registry

#### 9.4. CBOR Extended Key Usage Registry

IANA has created a new registry titled "CBOR Extended Key Usage Registry" under the new heading "CBOR Certificate". The columns of the registry are Value, Extended Key Usage Purpose, and Reference, where Value is an integer, and the other columns are text strings. For values in the interval  $[-24, 23]$  the registration procedure is "IETF Review" and "Expert Review". For all other values the registration procedure is "Expert Review". The initial contents of the registry are:

Value	Extended Key Usage
0	anyExtendedKeyUsage
1	id-kp-serverAuth
2	id-kp-clientAuth
3	id-kp-codeSigning
4	id-kp-emailProtection
5	id-kp-timeStamping
6	id-kp-OCSPSigning

Figure 6: CBOR Extended Key Usage Registry

### 9.5. CBOR General Name Registry

IANA has created a new registry titled "CBOR General Name Registry" under the new heading "CBOR Certificate". The columns of the registry are Value, General Name, and Reference, where Value is an integer, and the other columns are text strings. For values in the interval  $[-24, 23]$  the registration procedure is "IETF Review" and "Expert Review". For all other values the registration procedure is "Expert Review". The initial contents of the registry are:

Value	General Name Type	Value
-1	otherName + hardwareModuleName	[ oid, bytes ]
0	otherName	[ oid, bytes ]
1	rfc822Name	text
2	dNSName	text
4	directoryName	Name
6	uniformResourceIdentifier	text
7	iPAddress	bytes
8	registeredID	oid

Figure 7: CBOR General Name Registry

### 9.6. CBOR Certificate Signature Algorithms Registry

IANA has created a new registry titled "CBOR Certificate Signature Algorithms" under the new heading "CBOR Certificate". For values in the interval  $[-24, 23]$  the registration procedure is "IETF Review" and "Expert Review". For all other values the registration procedure is "Expert Review". The columns of the registry are Value, X.509 Algorithm, and Reference, where Value is an integer, and the other

columns are text strings. The initial contents of the registry are:

Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

Value	X.509 Signature Algorithm
-256	sha1WithRSAEncryption
-255	ecdsa-with-SHA1
1	sha256WithRSAEncryption
2	sha384WithRSAEncryption
3	sha512WithRSAEncryption
4	id-RSASSA-PSS-SHAKE128
5	id-RSASSA-PSS-SHAKE256
6	ecdsa-with-SHA256
7	ecdsa-with-SHA384
8	ecdsa-with-SHA512
9	id-ecdsa-with-shake128
10	id-ecdsa-with-shake256
11	id-Ed25519
12	id-Ed448
13	id-alg-hss-lms-hashsig
14	id-alg-xmss
15	id-alg-xmssmt
245	sha224WithRSAEncryption
246	id-rsassa-pkcs1-v1_5-with-sha3-224
247	id-rsassa-pkcs1-v1_5-with-sha3-256
248	id-rsassa-pkcs1-v1_5-with-sha3-384
249	id-rsassa-pkcs1-v1_5-with-sha3-512
251	ecdsa-with-SHA224
252	id-ecdsa-with-sha3-224
253	id-ecdsa-with-sha3-256
254	id-ecdsa-with-sha3-384
255	id-ecdsa-with-sha3-512

Figure 8: CBOR Certificate Signature Algorithms

### 9.7. CBOR Certificate Public Key Algorithms Registry

IANA has created a new registry titled "CBOR Certificate Public Key Algorithms" under the new heading "CBOR Certificate". For values in the interval  $[-24, 23]$  the registration procedure is "IETF Review" and "Expert Review". For all other values the registration procedure

is "Expert Review". The columns of the registry are Value, X.509 Algorithm, and Reference, where Value is an integer, and the other columns are text strings. The initial contents of the registry are:

## Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

Value	X.509 Public Key Algorithm
0	rsaEncryption
1	id-ecPublicKey + secp256r1
2	id-ecPublicKey + secp384r1
3	id-ecPublicKey + secp521r1
4	id-X25519
5	id-X448
6	id-Ed25519
7	id-Ed448
8	id-alg-hss-lms-hashsig
9	id-alg-xmss
10	id-alg-xmssmt

Figure 9: CBOR Certificate Public Key Algorithms

### 9.8. COSE Header Parameters Registry

This document registers the following entries in the "COSE Header Parameters" registry under the "CBOR Object Signing and Encryption (COSE)" heading. The formatting and processing are the same as the corresponding x5bag, x5chain, x5t, and x5u defined in [\[I-D.ietf-cose-x509\]](#) except that the certificates are CBOR encoded instead of DER encoded. Note that certificates can also be identified with a 'kid' header parameter by storing 'kid' and the associated bag or chain in a dictionary.

Name	Label	Value Type	Description
c5bag	TBD1	COSE_X509	An unordered bag of

			CBOR certificates
c5chain	TBD2	COSE_X509	An ordered chain of CBOR certificates
c5t	TBD3	COSE_CertHash	Hash of an CBOR certificate
c5u	TBD4	uri	URI pointing to a CBOR certificate

## 9.9. TLS Certificate Types Registry

This document registers the following entry in the "TLS Certificate Types" registry under the "Transport Layer Security (TLS) Extensions" heading. The new certificate type can be used with additional TLS certificate compression [[I-D.ietf-tls-certificate-compression](#)].

EDITOR'S NOTE: The TLS registrations should be discussed and approved by the TLS WG at a later stage. When COSE WG has adopted work on CBOR certificates, it could perhaps be presented in the TLS WG. The TLS WG might e.g. want a separate draft in the TLS WG.

Value	Name	Recommended	Comment
TBD5	CBOR Certificate	Y	

## 10. References

### 10.1. Normative References

[I-D.ietf-cbor-tags-oid]

Bormann, C. and S. Leonard, "Concise Binary Object Representation (CBOR) Tags for Object Identifiers", [draft-ietf-cbor-tags-oid-03](#) (work in progress), November 2020.

[I-D.ietf-cose-x509]

Schaad, J., "CBOR Object Signing and Encryption (COSE): Header parameters for carrying and referencing X.509 certificates", [draft-ietf-cose-x509-07](#) (work in progress), September 2020.

[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>>.

[RFC4108] Housley, R., "Using Cryptographic Message Syntax (CMS) to Protect Firmware Packages", [RFC 4108](#), DOI 10.17487/RFC4108, August 2005, <<https://www.rfc-editor.org/info/rfc4108>>.

[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>>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", [RFC 7049](#), DOI 10.17487/RFC7049, October 2013, <<https://www.rfc-editor.org/info/rfc7049>>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", [RFC 8152](#), DOI 10.17487/RFC8152, July 2017, <<https://www.rfc-editor.org/info/rfc8152>>.

[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>>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", [RFC 8610](#), DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.

[RFC8742] Bormann, C., "Concise Binary Object Representation (CBOR)

Sequences", [RFC 8742](https://www.rfc-editor.org/info/rfc8742), DOI 10.17487/RFC8742, February 2020, <<https://www.rfc-editor.org/info/rfc8742>>.

[SECG] "Elliptic Curve Cryptography, Standards for Efficient Cryptography Group, ver. 2", 2009, <<https://secg.org/sec1-v2.pdf>>.

## 10.2. Informative References

- [I-D.ietf-lake-edhoc]  
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", [draft-ietf-lake-edhoc-02](#) (work in progress), November 2020.
- [I-D.ietf-tls-certificate-compression]  
Ghedini, A. and V. Vasiliev, "TLS Certificate Compression", [draft-ietf-tls-certificate-compression-10](#) (work in progress), January 2020.
- [I-D.ietf-tls-ctls]  
Rescorla, E., Barnes, R., and H. Tschofenig, "Compact TLS 1.3", [draft-ietf-tls-ctls-01](#) (work in progress), November 2020.
- [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-dtls13-39](#) (work in progress), November 2020.

- [I-D.ietf-uta-tls13-iot-profile]  
Tschofenig, H. and T. Fossati, "TLS/DTLS 1.3 Profiles for the Internet of Things", [draft-ietf-uta-tls13-iot-profile-00](#) (work in progress), June 2020.
- [IEEE-802.1AR]  
Institute of Electrical and Electronics Engineers, ., "IEEE Standard for Local and metropolitan area networks-Secure Device Identity", IEEE Standard 802.1AR-2018 , August 2018, <[https://standards.ieee.org/standard/802\\_1AR-2018.html](https://standards.ieee.org/standard/802_1AR-2018.html)>.

- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", [RFC 7228](#), DOI 10.17487/RFC7228, May 2014, <<https://www.rfc-editor.org/info/rfc7228>>.
- [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>>.
- [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>>.
- [X.509-IoT] Forsby, F., Furuher, M., Papadimitratos, P., and S. Raza, "Lightweight X.509 Digital Certificates for the Internet of Things.", Springer, Cham. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 242., July 2018, <[https://doi.org/10.1007/978-3-319-93797-7\\_14](https://doi.org/10.1007/978-3-319-93797-7_14)>.

## [Appendix A](#). Example CBOR Certificates

### [A.1](#). Example [RFC 7925](#) profiled X.509 Certificate

Example of [[RFC7925](#)] profiled X.509 certificate parsed with OpenSSL.

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 128269 (0x1f50d)

Signature Algorithm: ecdsa-with-SHA256



```

Issuer: CN=RFC test CA
Validity
    Not Before: Jan  1 00:00:00 2020 GMT
    Not After : Feb  2 00:00:00 2021 GMT
Subject: CN=01-23-45-FF-FE-67-89-AB
Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
    Public-Key: (256 bit)
    pub:
        04:b1:21:6a:b9:6e:5b:3b:33:40:f5:bd:f0:2e:69:
        3f:16:21:3a:04:52:5e:d4:44:50:b1:01:9c:2d:fd:
        38:38:ab:ac:4e:14:d8:6c:09:83:ed:5e:9e:ef:24:
        48:c6:86:1c:c4:06:54:71:77:e6:02:60:30:d0:51:
        f7:79:2a:c2:06
    ASN1 OID: prime256v1
    NIST CURVE: P-256
X509v3 extensions:
    X509v3 Key Usage:
        Digital Signature
Signature Algorithm: ecdsa-with-SHA256
    30:44:02:20:44:5d:79:8c:90:e7:f5:00:dc:74:7a:65:4c:ec:
    6c:fa:6f:03:72:76:e1:4e:52:ed:07:fc:16:29:4c:84:66:0d:
    02:20:5a:33:98:5d:fb:d4:bf:dd:6d:4a:cf:38:04:c3:d4:6e:
    bf:3b:7f:a6:26:40:67:4f:c0:35:4f:a0:56:db:ae:a6

```

The DER encoding of the above certificate is 314 bytes.

```

30 82 01 36 30 81 DE A0 03 02 01 02 02 03 01 F5 0D 30 0A 06 08 2A 86
48 CE 3D 04 03 02 30 16 31 14 30 12 06 03 55 04 03 0C 0B 52 46 43 20
74 65 73 74 20 43 41 30 1E 17 0D 32 30 30 31 30 31 30 30 30 30 30
5A 17 0D 32 31 30 32 30 32 30 30 30 30 30 5A 30 22 31 20 30 1E 06
03 55 04 03 0C 17 30 31 2D 32 33 2D 34 35 2D 46 46 2D 46 45 2D 36 37
2D 38 39 2D 41 42 30 59 30 13 06 07 2A 86 48 CE 3D 02 01 06 08 2A 86
48 CE 3D 03 01 07 03 42 00 04 B1 21 6A B9 6E 5B 3B 33 40 F5 BD F0 2E
69 3F 16 21 3A 04 52 5E D4 44 50 B1 01 9C 2D FD 38 38 AB AC 4E 14 D8
6C 09 83 ED 5E 9E EF 24 48 C6 86 1C C4 06 54 71 77 E6 02 60 30 D0 51
F7 79 2A C2 06 A3 0F 30 0D 30 0B 06 03 55 1D 0F 04 04 03 02 07 80 30
0A 06 08 2A 86 48 CE 3D 04 03 02 03 47 00 30 44 02 20 44 5D 79 8C 90
E7 F5 00 DC 74 7A 65 4C EC 6C FA 6F 03 72 76 E1 4E 52 ED 07 FC 16 29
4C 84 66 0D 02 20 5A 33 98 5D FB D4 BF DD 6D 4A CF 38 04 C3 D4 6E BF
3B 7F A6 26 40 67 4F C0 35 4F A0 56 DB AE A6

```

#### [A.1.1.](#) Example CBOR Certificate Encoding

The CBOR encoding of the same X.509 certificate is shown below in CBOR diagnostic format.

/This defines a CBOR Sequence ([RFC 8742](#)):/

```
1,
h'01f50d',
"RFC test CA",
1577836800,
1612224000,
h'0123456789AB',
1,
h'02B1216AB96E5B3B3340F5BDF02E693F16213A04525ED44450
  B1019C2DFD3838AB',
1,
6,
h'445D798C90E7F500DC747A654CEC6CFA6F037276E14E52ED07
  FC16294C84660D5A33985DFBD4BFDD6D4ACF3804C3D46EBF3B
  7FA62640674FC0354FA056DBAEA6
```

The size of the CBOR encoding (CBOR sequence) is 138 bytes.

```
01
43 01 F5 0D
6B 52 46 43 20 74 65 73 74 20 43 41
1A 5E 0B E1 00
1A 60 18 96 00
46 01 23 45 67 89 AB
01
58 21 02 B1 21 6A B9 6E 5B 3B 33 40 F5 BD F0 2E 69 3F 16 21 3A 04 52
5E D4 44 50 B1 01 9C 2D FD 38 38 AB
01
06
58 40 44 5D 79 8C 90 E7 F5 00 DC 74 7A 65 4C EC 6C FA 6F 03 72 76 E1
4E 52 ED 07 FC 16 29 4C 84 66 0D 5A 33 98 5D FB D4 BF DD 6D 4A CF 38
04 C3 D4 6E BF 3B 7F A6 26 40 67 4F C0 35 4F A0 56 DB AE A6
```

#### [A.1.2.](#) Example: Natively Signed CBOR Certificate

The corresponding natively signed CBOR certificate in CBOR diagnostic format is identical, except for `cborCertificateType` and `signatureValue`.

## Internet-Draft CBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

/This defines a CBOR Sequence ([RFC 8742](#)):/

```

0,
h'01f50d',
"RFC test CA",
1577836800,
1612224000,
h'0123456789AB',
1,
h'02B1216AB96E5B3B3340F5BDF02E693F16213A04525ED44450
  B1019C2DFD3838AB',
1,
6,
h'B27A0B781455F71B68290F6C2EC9A897F18FDE9B6C59575953
  BC67268AB0E4DDE99D273E04E4715383AB2257C6AAA35284E5
  ED18BDB91247E9F2C433136480B9'
```

The size of the CBOR encoding (CBOR sequence) is 138 bytes.

```

00
43 01 F5 0D
6B 52 46 43 20 74 65 73 74 20 43 41
1A 5E 0B E1 00
1A 60 18 96 00
46 01 23 45 67 89 AB
01
58 21 02 B1 21 6A B9 6E 5B 3B 33 40 F5 BD F0 2E 69 3F 16 21 3A 04 52
5E D4 44 50 B1 01 9C 2D FD 38 38 AB
01
06
58 40 B2 7A 0B 78 14 55 F7 1B 68 29 0F 6C 2E C9 A8 97 F1 8F DE 9B 6C
59 57 59 53 BC 67 26 8A B0 E4 DD E9 9D 27 3E 04 E4 71 53 83 AB 22 57
C6 AA A3 52 84 E5 ED 18 BD B9 12 47 E9 F2 C4 33 13 64 80 B9
```

### [A.1.3.](#) Example: Key Pair

All of the the above example certificate were signed and can be verified with the following key pair where the private key is encoded as in COSE [[RFC8152](#)].

---

Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

issuerPublicKeyAlgorithm :

1 (id-ecPublicKey + secp256r1)

issuerPublicKey :

h'02AE4CDB01F614DEFC7121285FDC7F5C6D1D42C95647F061BA0080DF678867845E'

issuerPrivateKey :

h'DC66B3415456D649429B53223DF7532B942D6B0E0842C30BCA4C0ACF91547BB2'

## [A.2.](#) Example HTTPS X.509 Certificate

The DER encoding of the tools.ietf.org certificate is 1647 bytes.

```
30 82 06 6b 30 82 05 53 a0 03 02 01 02 02 09 00 a6 a5 5c 87 0e 39 b4
0e 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0b 05 00 30 81 c6 31 0b 30 09
06 03 55 04 06 13 02 55 53 31 10 30 0e 06 03 55 04 08 13 07 41 72 69
7a 6f 6e 61 31 13 30 11 06 03 55 04 07 13 0a 53 63 6f 74 74 73 64 61
6c 65 31 25 30 23 06 03 55 04 0a 13 1c 53 74 61 72 66 69 65 6c 64 20
54 65 63 68 6e 6f 6c 6f 67 69 65 73 2c 20 49 6e 63 2e 31 33 30 31 06
03 55 04 0b 13 2a 68 74 74 70 3a 2f 2f 63 65 72 74 73 2e 73 74 61 72
66 69 65 6c 64 74 65 63 68 2e 63 6f 6d 2f 72 65 70 6f 73 69 74 6f 72
79 2f 31 34 30 32 06 03 55 04 03 13 2b 53 74 61 72 66 69 65 6c 64 20
53 65 63 75 72 65 20 43 65 72 74 69 66 69 63 61 74 65 20 41 75 74 68
6f 72 69 74 79 20 2d 20 47 32 30 1e 17 0d 32 30 31 30 30 31 31 39 33
38 33 36 5a 17 0d 32 31 31 31 30 32 31 39 33 38 33 36 5a 30 3e 31 21
30 1f 06 03 55 04 0b 13 18 44 6f 6d 61 69 6e 20 43 6f 6e 74 72 6f 6c
20 56 61 6c 69 64 61 74 65 64 31 19 30 17 06 03 55 04 03 0c 10 2a 2e
74 6f 6f 6c 73 2e 69 65 74 66 2e 6f 72 67 30 82 01 22 30 0d 06 09 2a
86 48 86 f7 0d 01 01 01 05 00 03 82 01 0f 00 30 82 01 0a 02 82 01 01
00 b1 e1 37 e8 eb 82 d6 89 fa db f5 c2 4b 77 f0 2c 4a de 72 6e 3e 13
60 d1 a8 66 1e c4 ad 3d 32 60 e5 f0 99 b5 f4 7a 7a 48 55 21 ee 0e 39
12 f9 ce 0d ca f5 69 61 c7 04 ed 6e 0f 1d 3b 1e 50 88 79 3a 0e 31 41
16 f1 b1 02 64 68 a5 cd f5 4a 0a ca 99 96 35 08 c3 7e 27 5d d0 a9 cf
f3 e7 28 af 37 d8 b6 7b dd f3 7e ae 6e 97 7f f7 ca 69 4e cc d0 06 df
5d 27 9b 3b 12 e7 e6 fe 08 6b 52 7b 82 11 7c 72 b3 46 eb c1 e8 78 b8
```

0f cb e1 eb bd 06 44 58 dc 83 50 b2 a0 62 5b dc 81 b8 36 e3 9e 7c 79  
b2 a9 53 8a e0 0b c9 4a 2a 13 39 31 13 bd 2c cf a8 70 cf 8c 8d 3d 01  
a3 88 ae 12 00 36 1d 1e 24 2b dd 79 d8 53 01 26 ed 28 4f c9 86 94 83  
4e c8 e1 14 2e 85 b3 af d4 6e dd 69 46 af 41 25 0e 7a ad 8b f2 92 ca  
79 d9 7b 32 4f f7 77 e8 f9 b4 4f 23 5c d4 5c 03 ae d8 ab 3a ca 13 5f  
5d 5d 5d a1 02 03 01 00 01 a3 82 02 e1 30 82 02 dd 30 0c 06 03 55 1d  
13 01 01 ff 04 02 30 00 30 1d 06 03 55 1d 25 04 16 30 14 06 08 2b 06  
01 05 05 07 03 01 06 08 2b 06 01 05 05 07 03 02 30 0e 06 03 55 1d 0f  
01 01 ff 04 04 03 02 05 a0 30 3d 06 03 55 1d 1f 04 36 30 34 30 32 a0  
30 a0 2e 86 2c 68 74 74 70 3a 2f 2f 63 72 6c 2e 73 74 61 72 66 69 65  
6c 64 74 65 63 68 2e 63 6f 6d 2f 73 66 69 67 32 73 31 2d 32 34 32 2e  
63 72 6c 30 63 06 03 55 1d 20 04 5c 30 5a 30 4e 06 0b 60 86 48 01 86  
fd 6e 01 07 17 01 30 3f 30 3d 06 08 2b 06 01 05 05 07 02 01 16 31 68

---

Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

74 74 70 3a 2f 2f 63 65 72 74 69 66 69 63 61 74 65 73 2e 73 74 61 72  
66 69 65 6c 64 74 65 63 68 2e 63 6f 6d 2f 72 65 70 6f 73 69 74 6f 72  
79 2f 30 08 06 06 67 81 0c 01 02 01 30 81 82 06 08 2b 06 01 05 05 07  
01 01 04 76 30 74 30 2a 06 08 2b 06 01 05 05 07 30 01 86 1e 68 74 74  
70 3a 2f 2f 6f 63 73 70 2e 73 74 61 72 66 69 65 6c 64 74 65 63 68 2e  
63 6f 6d 2f 30 46 06 08 2b 06 01 05 05 07 30 02 86 3a 68 74 74 70 3a  
2f 2f 63 65 72 74 69 66 69 63 61 74 65 73 2e 73 74 61 72 66 69 65 6c  
64 74 65 63 68 2e 63 6f 6d 2f 72 65 70 6f 73 69 74 6f 72 79 2f 73 66  
69 67 32 2e 63 72 74 30 1f 06 03 55 1d 23 04 18 30 16 80 14 25 45 81  
68 50 26 38 3d 3b 2d 2c be cd 6a d9 b6 3d b3 66 63 30 2b 06 03 55 1d  
11 04 24 30 22 82 10 2a 2e 74 6f 6f 6c 73 2e 69 65 74 66 2e 6f 72 67  
82 0e 74 6f 6f 6c 73 2e 69 65 74 66 2e 6f 72 67 30 1d 06 03 55 1d 0e  
04 16 04 14 ad 8a b4 1c 07 51 d7 92 89 07 b0 b7 84 62 2f 36 55 7a 5f  
4d 30 82 01 06 06 0a 2b 06 01 04 01 d6 79 02 04 02 04 81 f7 04 81 f4  
00 f2 00 77 00 f6 5c 94 2f d1 77 30 22 14 54 18 08 30 94 56 8e e3 4d  
13 19 33 bf df 0c 2f 20 0b cc 4e f1 64 e3 00 00 01 74 e5 ac 71 13 00  
00 04 03 00 48 30 46 02 21 00 8c f5 48 52 ce 56 35 43 39 11 cf 10 cd  
b9 1f 52 b3 36 39 22 3a d1 38 a4 1d ec a6 fe de 1f e9 0f 02 21 00 bc  
a2 25 43 66 c1 9a 26 91 c4 7a 00 b5 b6 53 ab bd 44 c2 f8 ba ae f4 d2  
da f2 52 7c e6 45 49 95 00 77 00 5c dc 43 92 fe e6 ab 45 44 b1 5e 9a  
d4 56 e6 10 37 fb d5 fa 47 dc a1 73 94 b2 5e e6 f6 c7 0e ca 00 00 01  
74 e5 ac 72 3c 00 00 04 03 00 48 30 46 02 21 00 a5 e0 90 6e 63 e9 1d  
4f dd ef ff 03 52 b9 1e 50 89 60 07 56 4b 44 8a 38 28 f5 96 dc 6b 28  
72 6d 02 21 00 fc 91 ea ed 02 16 88 66 05 4e e1 8a 2e 53 46 c4 cc 51  
fe b3 fa 10 a9 1d 2e db f9 91 25 f8 6c e6 30 0d 06 09 2a 86 48 86 f7  
0d 01 01 0b 05 00 03 82 01 01 00 14 04 3f a0 be d2 ee 3f a8 6e 3a 1f  
78 8e a0 4c 35 53 0f 11 06 1f ff 60 a1 6d 0b 83 e9 d9 2a db b3 3f 9d  
b3 d7 e0 59 4c 19 a8 e4 19 a5 0c a7 70 72 77 63 d5 fe 64 51 0a d2 7a

```
d6 50 a5 8a 92 38 ec cb 2f 0f 5a c0 64 58 4d 5c 06 b9 73 63 68 27 8b
89 34 dc 79 c7 1d 3a fd 34 5f 83 14 41 58 49 80 68 29 80 39 8a 86 72
69 cc 79 37 ce e3 97 f7 dc f3 95 88 ed 81 03 29 00 d2 a2 c7 ba ab d6
3a 8e ca 09 0b d9 fb 39 26 4b ff 03 d8 8e 2d 3f 6b 21 ca 8a 7d d8 5f
fb 94 ba 83 de 9c fc 15 8d 61 fa 67 2d b0 c7 db 3d 25 0a 41 4a 85 d3
7f 49 46 37 3c f4 b1 75 d0 52 f3 dd c7 66 f1 4b fd aa 00 ed bf e4 7e
ed 01 ec 7b e4 f6 46 fc 31 fd 72 fe 03 d2 f2 65 af 4d 7e e2 81 9b 7a
fd 30 3c f5 52 f4 05 34 a0 8a 3e 19 41 58 c8 a8 e0 51 71 84 09 15 ae
ec a5 77 75 fa 18 f7 d5 77 d5 31 cc c7 2d
```

#### [A.2.1.](#) Example CBOR Certificate Encoding

The CBOR encoding of the X.509 certificate is shown below in CBOR diagnostic format.

---

Internet-DraftCBOR Encoding of X.509 Certificates (CBOR CerDecember 2020

/This defines a CBOR Sequence ([RFC 8742](#)):/

```
1,
h'A6A55C870E39B40E',
[
  [-4, "US"],
  [-6, "Arizona"],
  [-5, "Scottsdale"],
  [-7, "Starfield Technologies, Inc."],
  [-8, "http://certs.starfieldtech.com/repository/"],
  [-1, "Starfield Secure Certificate Authority - G2"]
],
1601581116,
1635881916,
[
  [-8, "Domain Control Validated"],
  [1, "*.tools.ietf.org"]
],
0,
h'3082010A0282010100B1E137E8EB82D689FADBF5C24B77F02C4ADE726E3E1360D1A8661EC4A
[
```

```

-1, -2,
  3, [ 1, 2 ],
-2, 5,
  8, "http://crl.starfieldtech.com/sfig2s1-242.crl",
  7, h'305A304E060B6086480186FD6E01071701303F303D06082B060105050702011631687
  9, [ 1, "http://ocsp.starfieldtech.com/", 2, "http://certificates.starfiel
  5, h'254581685026383D3B2D2CBECD6AD9B63DB36663',
  4, [ 2, "*.tools.ietf.org", 2, "tools.ietf.org" ],
  6, h'AD8AB41C0751D7928907B0B784622F36557A5F4D',
  10, h'0481F400F2007700F65C942FD1773022145418083094568EE34D131933BFDF0C2F200
],
1,
h'14043FA0BED2EE3FA86E3A1F788EA04C35530F11061FFF60A16D0B83E9D92ADBB33F9DB3D7E

```

The size of the CBOR encoding (CBOR sequence) is 1332 bytes.

## [Appendix B](#). X.509 Certificate Profile, ASN.1

EDITOR'S NOTE: The ASN.1 below is not up to date with the rest of the specification. The below ASN.1 for [RFC 7925](#) profile should be in [draft-ietf-uta-tls13-iot-profile](#) instead. If CBOR Certificates support a large subset of [RFC 5280](#), we should probably not duplicate all the ASN.1 in that document. Should be discussed what kind and how much (if any) ASN.1 this document needs. If possible, one option would be to have ASN.1 for the restrictions compared to [RFC 5280](#).

IOTCertificate DEFINITIONS EXPLICIT TAGS ::= BEGIN

```

Certificate ::= SEQUENCE {
    tbsCertificate      TBSCertificate,
    signatureAlgorithm   AlgorithmIdentifier,
    signatureValue       BIT STRING
}

```

```

TBSCertificate ::= SEQUENCE {
    version             [0] INTEGER {v3(2)},
    serialNumber         INTEGER (1..MAX),
    signature            AlgorithmIdentifier,
    issuer               Name,
    validity              Validity,
}

```

```

    subject          Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    extensions       [3] Extensions OPTIONAL
}

Name ::= SEQUENCE SIZE (1) OF DistinguishedName

DistinguishedName ::= SET SIZE (1) OF CommonName

CommonName ::= SEQUENCE {
    type          OBJECT IDENTIFIER (id-at-commonName),
    value         UTF8String
}

Validity ::= SEQUENCE {
    notBefore     UTCTime,
    notAfter      UTCTime
}

SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm      AlgorithmIdentifier,
    subjectPublicKey BIT STRING
}

AlgorithmIdentifier ::= SEQUENCE {
    algorithm      OBJECT IDENTIFIER,
    parameters    ANY DEFINED BY algorithm OPTIONAL }
}

Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension

Extension ::= SEQUENCE {
    extnId         OBJECT IDENTIFIER,
    critical       BOOLEAN DEFAULT FALSE,

```

```

    extnValue       OCTET STRING
}

id-at-commonName   OBJECT IDENTIFIER ::=
    {joint-iso-itu-t(2) ds(5) attributeType(4) 3}

END

```



## Acknowledgments

The authors want to thank Henk Birkholz, Carsten Bormann, Russ Housley, Olle Johansson, Benjamin Kaduk, Ilari Liusvaara, Laurence Lundblade, Thomas Peterson, Michael Richardson, Stefan Santesson, Jim Schaad, Fraser Tweedale, and Rene Struik for reviewing and commenting on intermediate versions of the draft.

## Authors' Addresses

Shahid Raza  
RISE AB

Email: [shahid.raza@ri.se](mailto:shahid.raza@ri.se)

Joel Hoeglund  
RISE AB

Email: [joel.hoglund@ri.se](mailto:joel.hoglund@ri.se)

Goeran Selander  
Ericsson AB

Email: [goran.selander@ericsson.com](mailto:goran.selander@ericsson.com)

John Preuss Mattsson  
Ericsson AB

Email: [john.mattsson@ericsson.com](mailto:john.mattsson@ericsson.com)

Martin Furuhed  
Nexus Group

Email: [martin.furuhed@nexusgroup.com](mailto:martin.furuhed@nexusgroup.com)