CBOR Compression of RFC 7925 Profiled X.509 Certificates

draft-raza-cose-cbor-cert-cert-compress-04
draft-mattsson-tls-cbor-cert-cert-compress-00
draft-mattsson-cose-cbor-cert-cert-compress-00

UTA, IETF 107, April 2020
X.509 certificates take up a large part of the total number of bytes in TLS 1.3 handshakes. Especially in cTLS.

Draft registers a code point to use the algorithm ‘CBOR Compression’ in draft-ietf-tls-certificate-compression.

Uses draft-raza-ace-cbor-certificates to compress certificates by encoding them from DER to CBOR.

The aim is to be compatible with all RFC 7925 profiled certificates.

General purpose compression algorithms (without dictionary) is not able to compress RFC 7925 X.509 certs much at all.

COSE WG is re-chartering, current plan is to work on CBOR compression of RFC 7925 (and maybe IEEE 802.11AR)
CBOR Compressed RFC 7925 X.509 certificates
draft-raza-ace-cbor-certificates-04

CBOR compression brings:

1) Compactness

2) Compatibility with and migration path from X.509

3) Smaller footprint than general compression algorithms.

4) Re-use of CBOR already used by COSE, but support for ASN.1 DER is still needed.

<table>
<thead>
<tr>
<th></th>
<th>RFC 7925 X.509</th>
<th>zlib</th>
<th>CBOR Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate Size</td>
<td>314 bytes</td>
<td>295</td>
<td>136</td>
</tr>
</tbody>
</table>
Example CBOR Compression of RFC 7925 X.509

```
{
1,
h'128269',
"RFC test CA",
1577836800,
161224000,
h'0123456789AB',
h'02ae4cdeb0f614d8fca7121285fde7c7f5c6d1d42c95d747f0e61ba0080df678867845e',
5,
h'373873EF8781B88297EF2351FA5CCF62DA4E44740DC2A2E6A3 C6C882A3238D9C3AD9353BA788683B06BB48FECA16EA711717 34C675C5332B2AF1CB733810A1FC'
}
```
• Is CBOR compression interesting for TLS 1.3 / cTLS?
• OK to make TLS IANA registration in COSE WG?
• Should draft-tschofenig-uta-tls13-profile mandate a X.509 profile?
• Should the RFC 7925 certificate profile be updated in any way and in that case where? E.g.
  • An ASN.1 schema would be extremely beneficial.
  • Is the encoding of EUI-64 as a X.509 text string specified somewhere?
  • Is ‘pathLenConstraint’ mandatory to support? In that case, are there any minimum length that is mandatory to support?
  • For ‘BasicConstraints’, the only valid options are "Present and true", and "Absent and therefore false". For the bool ‘critical’ in ‘expansions’, both "Present and false" and "Absent and therefore false" seems to be valid. Is this intentional?