CBOR Encoded X.509 Certificates
(C509 Certificates)

draft-ietf-cose-cbor-encoded-cert-02

IETF 111 COSE WG
Since IETF 110

- Adopted, now version -02
- Support for a large subset of RFC 5280 and all its extensions for certificates
- Support for certificates compatible with the following profiles:
  - RFC 7925, IEEE 802.1AR (DevID), CNSA, CAB Forum, RPKI, GSMA eUICC
- Updated based on review by Ilari, and input from Russ and Michael
  - Thanks!
- Expected Certificate Sizes
  - RFC 7925, RPKI, and HTTPS chains and bags for CBOR and TLS
- 509 Certificate Signing Request and Certificate Revocation List
- New section on CA C509 Processing and Certificate Issuance
- Open source code for generating C509
  - Rust: https://github.com/cose-wg/CBOR-certificates/tree/master/c509
  - C code in progress from Fraunhofer AISEC
Extensions: CBOR encoding fully supported

- **SubjectKeyIdentifier** = KeyIdentifier
  KeyIdentifier = bytes

- **KeyUsage** = int

- **PolicyMappings** = [ + (issuerDomainPolicy: ~oid, subjectDomainPolicy: ~oid) ]

- **BasicConstraints** = int

- **PolicyConstraints** = [ requireExplicitPolicy: uint / null, inhibitPolicyMapping: uint / null ]

- **ExtKeyUsageSyntax** = [ 2* KeyPurposeId ] / KeyPurposeId
  KeyPurposeId = int / ~oid

- **InhibitAnyPolicy** = uint
Extensions: CBOR encoding partly supported

- **SubjectAltName** = GeneralNames / text
  GeneralNames = [ + GeneralName ]
  GeneralName = ( GeneralNameType : int, GeneralNameValue : any )

- **IssuerAltName** = GeneralNames / text

- **CRLDistributionPoints** = [ + DistributionPointName ]
  DistributionPointName = [ 2* text ] / text

- **FreshestCRL** = CRLDistributionPoints

- **AuthorityInfoAccessSyntax** = [ + AccessDescription ]
  AccessDescription = ( accessMethod: int / ~oid, uri: text )

- **SubjectInfoAccessSyntax** = AuthorityInfoAccessSyntax
Extensions: CBOR encoding partly supported

- Authority Key Identifier
- Certificate Policies
- Name Constraints
- Subject Directory Attributes
- AS Resources (autonomousSysIds)
- AS Resources v2 (autonomousSysIds-v2)
- IP Resources (id-pe-ipAddrBlocks)
- IP Resources v2 (id-pe-ipAddrBlocks-v2)
- Signed Certificate Timestamp
C509 CSR

- based on and compatible with RFC 2986
- reusing the formatting for C509

C509CertificateSigningRequest = [  
  TBSCertificateSigningRequest,  
  subjectProofOfPossessionValue: any,  
]

TBSCertificateSigningRequest = (  
  c509CertificateSigningRequestType: int,  
  subject: Name,  
  subjectPublicKeyAlgorithm: AlgorithmIdentifier,  
  subjectPublicKey: any,  
  extensionsRequest : Extensions,  
  subjectProofOfPossessionAlgorithm: AlgorithmIdentifier,  
)

Two c509CertificateSigningRequestType values defined:
- 0 requests a c509CertificateType = 0
- 1 requests a c509CertificateType = 1
C509 CRL

- based on and compatible with RFC5280
- reusing the formatting for C509

C509CertificateRevocationList = [
  TBSCertificateRevocationList,
  issuerSignatureValue : any,
]

RevokedCertificates = [
  userCertificate: CertificateSerialNumber,
  revocationDate: Time,
  crlEntryExtensions: Extensions,
]

TBSCertificateRevocationList = (  
  C509CertificateRevocationListType: int,
  issuer: Name,
  thisUpdate: Time,
  nextUpdate: Time,
  revokedCertificates: RevokedCertificates,
  crlExtensions: Extensions,
  issuerSignatureAlgorithm: AlgorithmIdentifier,
)
## Sizes of Certificate Chains in COSE

<table>
<thead>
<tr>
<th>Size in bytes (length of chain)</th>
<th>COSE_X509</th>
<th>COSE_C509</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://tools.ietf.org/html/rfc7925">RFC 7925</a> profiled IoT Certificate (1)</td>
<td>317</td>
<td>139</td>
</tr>
<tr>
<td>ECDSA HTTPS Certificate Chain (2)</td>
<td>2193</td>
<td>1394</td>
</tr>
<tr>
<td>RSA HTTPS Certificate Chain (4)</td>
<td>5175</td>
<td>3934</td>
</tr>
</tbody>
</table>
## Sizes of Certificate Chains with TLS

<table>
<thead>
<tr>
<th>(length of chain)</th>
<th>X509</th>
<th>X509 + Brotli</th>
<th>C509</th>
<th>C509 + Brotli</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 7925 Cert (1)</td>
<td>327</td>
<td>324</td>
<td>151</td>
<td>167</td>
</tr>
<tr>
<td>RPKI Cert (1)</td>
<td>20991</td>
<td>9134</td>
<td>8660</td>
<td>5668</td>
</tr>
<tr>
<td>HTTPS Chain (2)</td>
<td>2204</td>
<td>1455</td>
<td>1414</td>
<td>1063</td>
</tr>
<tr>
<td>HTTPS Chain (4)</td>
<td>5190</td>
<td>3244</td>
<td>3958</td>
<td>2845</td>
</tr>
<tr>
<td>HTTPS Bag (8)</td>
<td>11578</td>
<td>3979</td>
<td>8882</td>
<td>3519</td>
</tr>
</tbody>
</table>
#98 Compression of chains

  — comp = 0 \(\Rightarrow\) no compression
  — other values of comp are use for a compressed chain conveyed in a bstr

— Would allow significant compression, compare values from TLS:

+-------------------+-------------------+
|                    | C509 Certificate  |
+-------------------+-------------------+
| ECDSA HTTPS Chain | 1409              |
+-------------------+-------------------+
| RSA HTTPS Chain   | 3957              |
+-------------------+-------------------+
Next steps

— Close Github issues (many are already resolved)
— Example compression of DevID certificate(s) that Michael Richardsson has provided.
— Decide if some additional compression of chains/bags should be supported #98
— Specify file format using draft-richardson-cbor-file-magic
— More tests on existing certificates
— Test compression of CRL
— More reviews, in particular encoding of extensions