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,
 ]
```

Two c509CertificateSigningRequestType values defined:

- 0 requests a c509CertificateType = 0
- 1 requests a c509CertificateType = 1

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

C509 CRL

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

```
C509CertificateRevocationList = [
      TBSCertificateRevocationList,
      issuerSignatureValue : any,
   ]
```

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

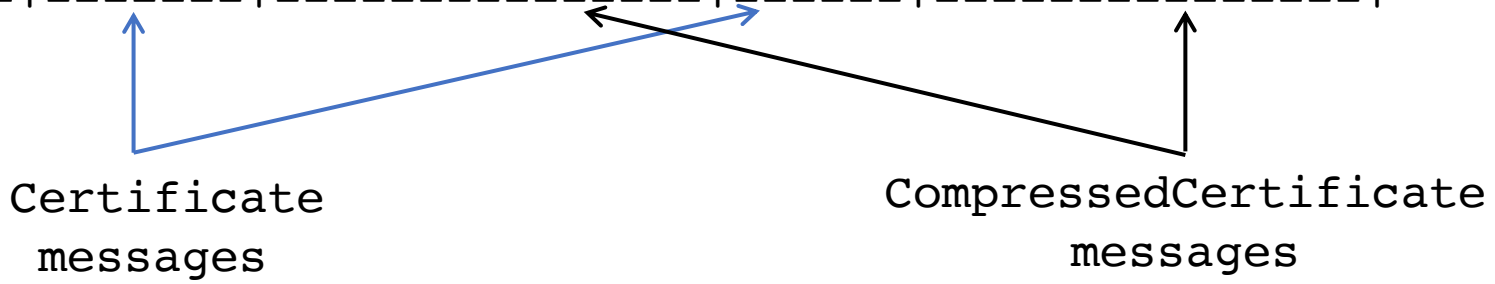
```
TBSCertificateRecovationList = (
      C509CertificateRevocationListType: int,
      issuer: Name,
      thisUpdate: Time,
      nextUpdate: Time,
      revokedCertificates: RevokedCertificates,
      crlExtensions: Extensions,
      issuerSignatureAlgorithm: AlgorithmIdentifier,
)
```

Sizes of Certificate Chains in COSE

Size in bytes (length of chain)	COSE_X509	COSE_C509
RFC 7925 profiled IoT Certificate (1)	317	139
ECDSA HTTPS Certificate Chain (2)	2193	1394
RSA HTTPS Certificate Chain (4)	5175	3934

Sizes of Certificate Chains with TLS

(length of chain)	X509	X509 + Brotli	C509	C509 + Brotli
RFC 7925 Cert (1)	327	324	151	167
RPKI Cert (1)	20991	9134	8660	5668
HTTPS Chain (2)	2204	1455	1414	1063
HTTPS Chain (4)	5190	3244	3958	2845
HTTPS Bag (8)	11578	3979	8882	3519



#98 Compression of chains

- Candidate definition: COSE_C509 = [comp : int, C509Certificate / [2* C509Certificate] / bytes]
  - comp = 0 → 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	C509 + Brotli
ECDSA HTTPS Chain	1409	1058
RSA HTTPS Chain	3957	2841

- Discussion: Should we define this?
- What compression can be achieved with draft-bormann-cbor-packed?

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