CBOR Encoded X.509 (C509)

draft-ietf-cose-cbor-encoded-certs-07

John Preuß Mattsson, Göran Selander, Ericsson AB
Shahid Raza, Joel Höglund, RISE AB
Martin Furuhed, Nexus Group

IETF 118, COSE WG, November 07, 2023

C509 Recap / Crash

- C509 is a compact CBOR encoding of X.509
- See top level CDDL →

- Can be used natively
  - **Type** = 0
  - Compact all-CBOR

- Or as compression of legacy X.509
  - **Type** = 1
  - Compression

```
C509Certificate = [
   TBSCertificate,
   issuerSignatureValue : any,
]


TBSCertificate = (
   c509CertificateType: int,
   certificateSerialNumber: CertificateSerialNumber,
   issuer: Name,
   validityNotBefore: Time,
   validityNotAfter: Time,
   subject: Name,
   subjectPublicKeyAlgorithm: AlgorithmIdentifier,
   subjectPublicKey: any,
   extensions: Extensions,
   issuerSignatureAlgorithm: AlgorithmIdentifier,
)
```

Ex.: C509 encoding of raa16376.cert from drip-dki

```
1,                        / X.509 v3, signature on DER encoding/
h'b6e6f5911185c478',      / certificateSerialNumber /
h'002001003000000005',    / issuer /
1684108800,               / notBefore /
1716508800,               / notAfter /
h'002001003ffe000005',    / subject /
10,                       / subjectPublicKeyAlgorithm = Ed25519 /
h'df7e64cc1bfdcb65835437b37b6110d56fedb81443f58d53df8094e0e2828d23',
[                         / extensions /
  1, h'2001003FFE000005F970A4D7FD0E14A5', / subjectKeyIdentifier /
  7, h'20010030000000052AEB9ADC1CE8B1EC', / authorityKeyIdentifier /
 -4, -1,                  / critical basicConstraints CA = True /
 -2, 1                    / critical keyUsage = digitalSignature /
],
12,                       / issuerSignatureAlgorithm = Ed25519 /
h'ab0f4085e0951b2be2dffaa9f5039d57ec5070a14cee3457d7edee591ec5528559
7b3d905ff76e79810b49c2ea6c713b6cad4a7c081abeb0f5619644da02510b'
```

CBOR diagnostic notation, plain hex is 183 bytes (X.509 is 331 bytes)

Changes -06 → -07 1(2)

- More efficient encoding of byte string Common Name of issuer / subject
  - Added byte disambiguating between byte string and EUI-64/48
  - Requested for use in draft-moskowitz-drip-dki

- Support of Certificate (Signing) Request for static DH public keys
  - Based on RFC 6955
    - Support for non-signature proof-of-possession
    - Requires a public Diffie-Hellman key of the verifier distributed out-of-band
    - Methods added to C509 Signature Algorithms Registry

- Certificate request attributes are supported using the extensionsRequest structure
  - ChallengePassword
    - Supports optimised encoding as byte strings

Changes -06 → -07 2(2)

- Added int assignments of extensions and attributes requested by Lijun Liao
- Fixed errors in EKU table detected by Brian Sipos
- Updated legacy and security considerations
- Updated examples, including annotations
- Revocation related content moved to separate draft
  - **Adoption call needed?**

What formats of CSR vs. requested Cert to support?

- C509 comes in two types:
  - 0: signature over CBOR
  - 1: signature over DER encoded X.509
- Similarly, the CBOR encoded CSR may be signed over CBOR or over DER

Request vs. Response	C509 with signature over CBOR	C509 with signature over DER	X.509
CBOR encoded CSR with signature over CBOR	X		
CBOR encoded CSR with CSR signature over DER		X	

- **Proposal:** Support
  - c509CertificateRequestType = 0: all signed-CBOR; native C509 case
  - c509CertificateRequestType = 1: all signed-ASN.1; legacy interop case

Next steps

- Remaining issues
  - Define file format
  - Define media types, including CBOR encoded CSR
  - Include IEEE-802.1AR example (if found)
- Update code to latest version
- WGLC?