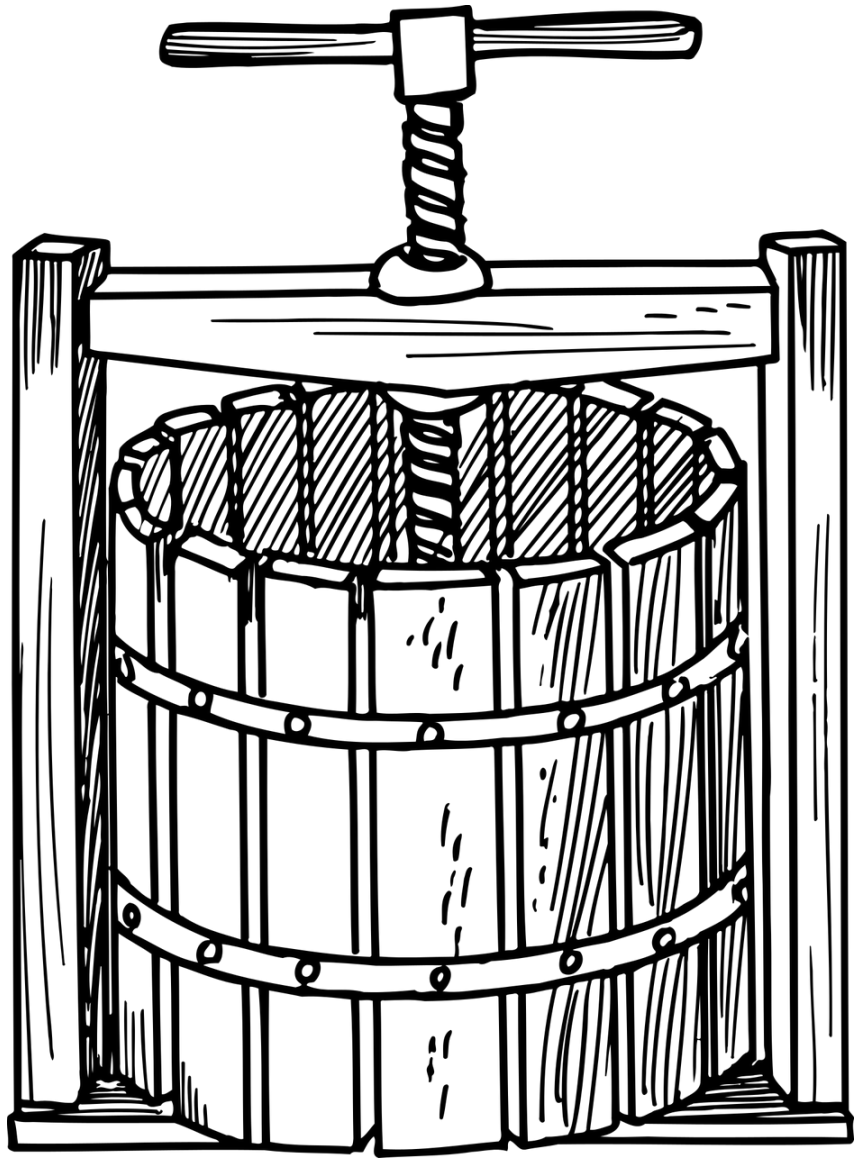
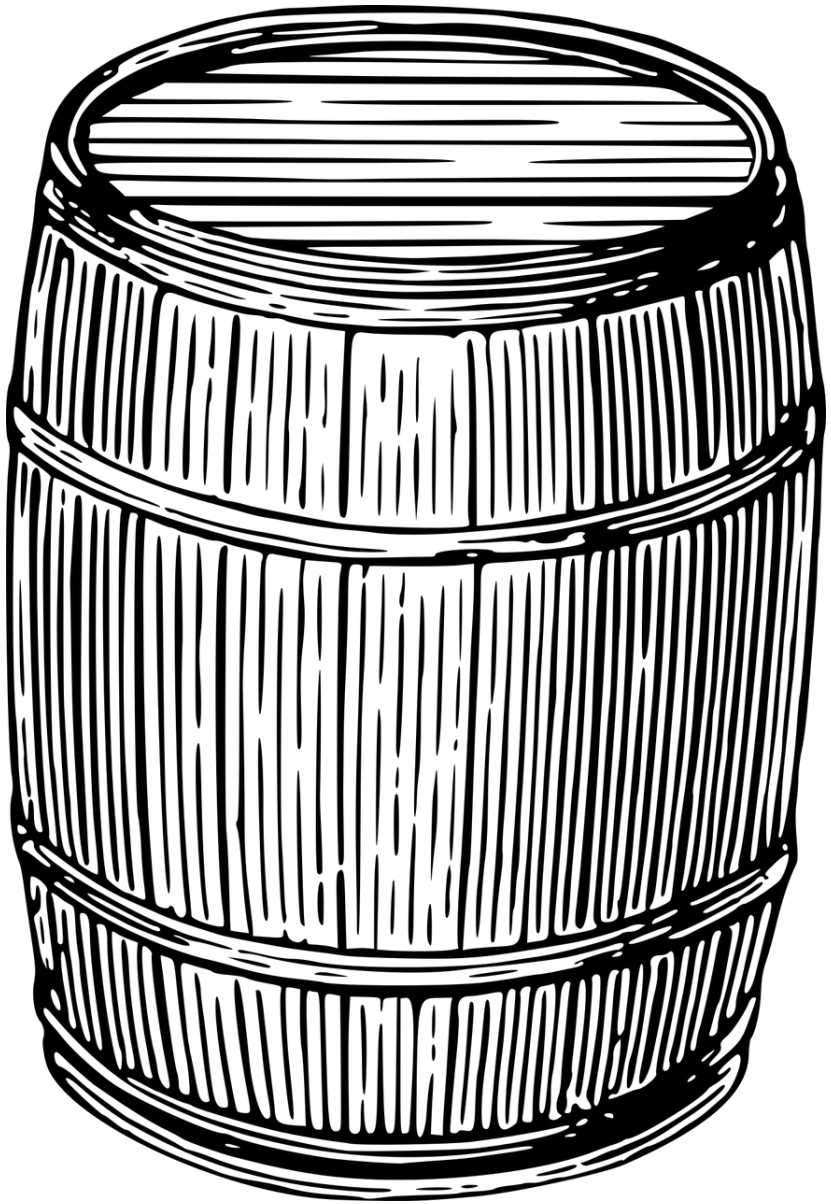


CBOR Encoding of X.509 Certificates (CBOR Certificates)

draft-mattsson-cose-cbor-cert-compress-05

COSE, John Preuß Mattsson



Changes from -03 to -05

Changes from -03 to -05

High level changes:

- Rephrased "compress" in the whole document. File name will be changed at a later point.
- Added reference to IEEE 802.1AR DevID and made optimization for IEEE 802.1AR DevID.
- Added "v3" in several places. Made it clear that X.509 version in included in the CBOR certificate type.
- Removed several not needed optimizations to simplify.

- Unwrapped CBOR positive bignum (~biguint) used for serialnumber.

- Change time to unwrapped CBOR epoch time (~time). This saves 1 byte for time ≥ 2050
- Added that the value "99991231235959Z" (no expiration date) is encoded as CBOR null. This is quite commonly used in IoT certificates. This saves 4 bytes when used.

- Use draft-ietf-cbor-tags-oid for OIDs and OID CDDL. Removed the word “relative“.

- Omitted 'signature' instead of 'singatureAlgorithm'. Now the algorithm comes just before the signature value.

Changes from -03 to -05

Attributes:

— Added id-at-organizationIdentifier
— Changed so utf8Strings have a positive sign.
— Added support of emailAddress / IA5String in issuer and subject.
— Fixed ambiguity with hex. Draft now only compresses EUI-64 with capital letters of the form "HH-HH-HH-HH-HH-HH-HH-HH"

Extensions:

— Added support of OtherName
— Added optimization for the hardwareModuleName type of otherName, which is mandatory to use in IEEE 802.1AR DevID
— Added support for registeredID. This was trivial and registeredID is used e.g. by the GSMA eUICC PKI Certificate Policy (SGP 14).
— Added CBOR encoding for AuthorityKeyIdentifier and subjectKeyIdentifier which are heavily used by IEEE 802.1AR DevID (at least one of them are mandated depending on type of cert).
— Added CBOR encoding for cRLDistributionPoints and authorityInfoAccess, which are both used in basically all HTTPS certificates on the web.

Changes from -03 to -05

- Moved SHA-1 signature algorithms to negative 2 byte encodings: -256 and -255
- Changes CDDL to allow / not allow encoding of empty SEQUENCE (OF) / SET (OF)
- Added a public private ECDSA key pair turning the example certificates into test vectors. An implementation can now create and verify the example RFC 7925 certificates.
- Added a note that certificates can be identified with 'kid' by storing them in a dictionary.
- Added considerations for expert reviewers.

Plans and discussions for -06

Plans and discussions for -06 (or later)

- Add references to other X.509 profiles:
  - CAB Baseline Requirements
  - CNSA X.509 Profile (RFC 8603)

- Add more references to protocols where Large certificate chains are also problematic and CBOR certificates might be useful:
  - EAP-TLS (draft-ietf-emu-eap-tls13, draft-ietf-emu-eaptlscert) where authenticators typically drop an EAP session after only 40 - 50 round-trips.
  - QUIC (draft-ietf-quic-transport) where the latency increases significantly unless the server only send less than three times as many bytes as received prior to validating the client address.

- Example encoding of IEEE 802.1AR DevID. We request input on example certificate.

- Example encoding of CAB Baseline ECDSA HTTPS X.509 Certificate (ietf.org)

- Brotli seems to be the compression algorithm used in practice. Should replace zlib example with Brotli. Preferably the draft should have a IoT certificate and HTTPS certificate chain examples with DER, Brotli, CBOR, CBOR+Brotli.

- Remove ASN.1 appendix.

Plans and discussions for -06 (or later)

- subject private key for the example certificates. This allows the example/test vector certificates to be used in EDHOC test vectors.

- More deployment guidance for IoT, comment that it would be good to discuss how different algorithms affect size.

- We tested the encoding on a set of HTTPS certificates. Quite a lot of attributes other than the ones RFC 5280 recommends support of in use:
  - Street Address
  - Postal Code
  - Business Category
  - Jurisdiction of Incorporation Country Name
  - ...

  - Plan to add street address and postal code to IANA registry as these are mentioned in CAB baseline requirements.
  - Plan to add (oid, bytes) encoding to support all other obscure attributes.

```
Attribute = ( attributeType : int, attributeValue : text ) // ( attributeType : ~oid, attributeValue : bytes )

AlgorithmIdentifier = int / ~oid
```

Plans and discussions for -06 (or later)

— Change Algorithms registries so that the int encodes the whole AlgorithmIdentifier. This simplifies things and allows algorithms with parameters to be registered. Examples:

Value	X.509 Public Key Algorithms	
0	Name:	RSA
	OID:	1.2.840.113549.1.1.1
	Parameters:	NULL
	DER:	30 0d 06 09 2a 86 48 86 f7 0d 01 01 01 05 00
	Comments:	Compressed subjectPublicKey
1	Name:	EC Public Key (Weierstraß) with secp256r1
	OID:	1.2.840.10045.2.1
	Parameters:	namedCurve = secp256r1 (1.2.840.10045.3.1.7)
	DER:	30 13 06 07 2A 86 48 CE 3D 02 01 06 08 2A 86 48 CE 3D 03 01 07
	Comments:	Point compressed subjectPublicKey
8	Name:	X25519 (Montgomery)
	OID:	1.3.101.110
	Parameters:	Absent
	DER:	30 05 06 03 2B 65 6E
	Comments:	
16	Name:	HSS / LMS
	OID:	1.2.840.113549.1.9.16.3.17
	Parameters:	Absent
	DER:	30 0D 06 0B 2A 86 48 86 F7 0D 01 09 10 03 11
	Comments:	

Value	X.509 Signature Algorithms	
-256	Name:	RSASSA-PKCS1-v1_5 with SHA-1
	OID:	1.2.840.113549.1.1.5
	Parameters:	NULL
	DER:	30 0D 06 09 2A 86 48 86 F7 0D 01 01 05 05 00
	Comments:	Don't use
0	Name:	ECDSA with SHA-256
	OID:	1.2.840.10045.4.3.2
	Parameters:	Absent
	DER:	30 0A 06 08 2A 86 48 CE 3D 04 03 02
	Comments:	Compressed signature value
12	Name:	Ed25519
	OID:	1.3.101.112
	Parameters:	Absent
	DER:	30 05 06 03 2B 65 70
	Comments:	
26	Name:	RSASSA-PSS with SHA-256
	OID:	1.2.840.113549.1.1.10
	Parameters:	SHA-256, MGF-1 with SHA-256, saltLength = 32
	DER:	30 41 06 09 2A 86 48 86 F7 0D 01 01 0A 30 34 A0 0F 30 0D 06 09 60 86 48 01 65 03 04 02 01 05 00 A1 1C 30 1A 06 09 2A 86 48 86 F7 0D 01 01 08 30 0D 06 09 60 86 48 01 65 03 04 02 01 05 00 a2 03 02 01 20
	Comments:	

Plans and discussions for -06 (or later)

- Do the format need to be adjusted to make implementation with some CBOR encoders and decoders easier?
- Laruance Lundblade: Some CBOR decoders don't allow access a sub-part of the encoded CBOR so it can be input into the signature algorithm. Suggest to wrap TBSCertificate in a byte string

```
bytes .cborseq [ TBSCertificate ]
```

- Stefan Hristozov : Easy to calculate offset with tinycbor and his own Rust implementation
- Joel Höglund: Small on the wire size is important.
- Carsten Bormann: One valid use of CBOR is to prefix a single CBOR data item to a byte sequence (not itself packaged as a CBOR string).
- Wrapping TBSCertificate in a byte string would add 2-3 additional bytes.

How to progress until next meeting

—Reviews

—Implementations

—Discussion on the list