DISPATCH
@IETF123

CBOR::Core

CBOR Cross Platform Profile

https://datatracker.ietf.org/doc/draft-rundgren-cbor-core/

Anders Rundgren, Independent

Motivation: Make CBOR Mainstream

CBOR has to date *mainly* been confined to embedded systems, making JSON the default for mainstream computing platforms like *Mobile phones*, *Web servers*, and *Web browsers*.

However, CBOR is considerably more advanced than JSON, including additional data types and compact binary encoding. There is also a fully compatible textual counterpart intended for *logging*, *test data*, and *configuration files*.

Through the use of *deterministic encoding*, cryptographic constructs beyond what the JOSE stack offer, are made possible.

`CBOR::Core` is a proposal for a *unified* and *interoperable* cross platform profile.

Fixed and Deterministic Encoding Format – No options needed

`CBOR::Core` exposes CBOR as a set of *common data types* and associated *high-level APIs*. The mandated encoding format is aligned with CBOR BCP.

Encode

```
let cbor = CBOR.Map()
    .set(CBOR.Int(1), CBOR.Float(45.7))
    .set(CBOR.Int(2), CBOR.String("Hi there!")).encode();
console.log(CBOR.toHex(cbor));
a201fb4046d9999999999a02694869207468657265721
```

Decode

```
let map = CBOR.decode(cbor);  // Use result from encode
console.log(map.toString());  // Diagnostic notation
{
  1: 45.7,
  2: "Hi there!"
}
console.log("Value=" + map.get(CBOR.Int(1)).getFloat64());
Value=45.7
```

Type checking API

Non-normative JavaScript Examples

Embedded Signatures – Enabled by Deterministic Encoding

Embedded Signatures operate over “raw” CBOR data, *enabling signatures to become a part of messages*. COSE and JOSE standards do the *opposite*.

```
1010(["https://example.com/myobject", {  # Object type Id (optional)
  1: "Hello signed world!",               # Application/message data
  2: [4.7, true, h'012345'],              #           -"-
  simple(99): {                 # Embedded Signature container label (reserved)
    1: -50,                     # COSE Signature Algorithm (Ed25519)
    4: {                        # COSE Public Key (Ed25519)
      1: 1,
      -1: 6,
      -2: h'fe49acf5b92b6e923594f2e83368f680ac924be93cf533aecaf802e37757f8c9'
    },                          # Embedded Signature value covering the entire object
    6: h'5de56d88d5ef3f7052b1e79f323fa585c6a6e5b5bfedb697cd61fe9515bf2578
         f2af03b149cbf43b5ab7176fa784cf4fbb66ba25943d1879a21ad57bc3c81d0f'
  }
}])
```

Non-normative (*but authentic*) example provided in *Diagnostic Notation*

IETF

References

Current Internet-Draft	https://www.ietf.org/archive/id/draft-rundgren-cbor-core-11.html
JavaScript Implementation	https://github.com/cyberphone/CBOR.js
Android/Java Implementation	https://github.com/cyberphone/android-cbor
Embedded Signatures	https://test.webpki.org/csf-lab/home

IETF

Hoped dispatching outcome

Since no IETF WG has shown interest in this work, I'm hoping on either making `CBOR::Core` an AD-sponsored item, alternatively publishing it through the Independent Stream.

I am actively seeking a coeditor as well as implementers.