COSE Overview

Jim Schaad
August Cellars

Willing Changes

- No crypto compatibility
- Use of CBOR idioms
- Partial change of naming schemes

Reuse

- Algorithm registries
- Header parameter registries
- Some basic paradigms

Bormann Overview

- Replace all base64 with binary
- Use CBOR rather than JSON
- Use integers for some map names
- Encoding uses array not periods

Bormann Example

- {1: h'436f6e74656e7420537472696e67', 2: [{3: h'74eea831242ee36b86558a3cdf128721d970c932422490986c995f5d348801af47a5c4a2d4138aab09bea2735ae1864ef51f18bc4483dcc7dd867bb004fa9a70041863d0c124ad9787bafe1e030158781e7d4a4e44bb9207dbca92e9619f5af893e8b837c0742825149066e13c108260305326a6cebfb376fc312dc3a80bee3340eeb78c570bc4c558d499fb00a3d09ebc256b123b98145725c1a0f49bf324a67f5bf754adec393b1d09be4e3d4295f7615d82acb47709b875de0a2f3915b996958638e982a3950802644d3c6ca1c6536dd70375e37b21065336983be88c26376c984c6c726af2db700c2ade35d74ad055bb0aa45ab067f71397399c4c64f89a', 4: {"kid": "bilbo.baggins@hobbiton.example", "alg": "PS256"}}, {3: h'00dc64e5e793e461e428b22d3636036e74fd304add0da24d49d77ca289018c2be01cb3c20867c989dd5ef9df7c86aee2bbbb410a2f3e3c4f5e6a21ead686f27853ed01a4a999e0b790dfcfd5081773632560 47c457cb96cda2e6dc5b4d0698791884b5871107091a1246e950bcaf37e078c4819ccbe8b7f522f32743f0cad318230b0d69', 4: {"kid": "bilbo.baggins@hobbiton.example", "alg": "ES512"}}]}

Schaad Overview

- Use CBOR rather than JSON
- Replace all base64 with binary
- Replace some maps with arrays
- Separate MAC and Signature
- Add recipient management for MAC
- Use same structure for encrypted body and recipients

Example

- [0, null, null, h'436f6e74656e742053747269e67', [[null, {"kid": "bilbo.baggins@hobbiton.example", "alg": "PS256"}, h'702efa196fc5182fae561d04392f9f4b15b19d6c8bbdb46e1445892a4a0f914cf7147486808bc1363e0d67dd2d862bbf86b954b675b57c88898bcad7aa427e1509e18154c9772bb3492761cf9728e339683f25ca2ae75f9d855e6613e8a2aecf75dcff9d2ab07801124a862cd57327df0c4fc4990adf79e8c59382b4e3f50e00a4c8880a649562af8a2c28b0768d825df299f8013d5bc242ebf7d2aab38da337e668981571129c325b8896deb799c5556ef0797a45060f0203418aa97ee9ed403b759afa97a9e976dddd0b027892796 5bd0aa02e7d57959d7a15cb6c9a9b9786c0f5c5f9c0d80971bf9c1db153fdf360a8207a5f0761420711512a008d7c2562'], [null, {"kid": "bilbo.baggins@hobbiton.example", "alg": "ES512"}, h'00b998c956c97802477478b9d7a9be897d14c17403716c359d5b7f21002c72089eef32f03f2f0fe3f69d778d820975dad728027ab680c2e7556d130aa7657ffbf08501f793d28ceef5b2d2da63cdbafc60c12ac919a1ee7714797ef4bb8d7b68ee1d6dd3bb41fd7910cf90ae5c008b919f49decf830e5fe348208ae6a37a3a00a76f9a17']]]

Size Comparisons

- Overheads only – ignores contents

	JOSE	Bormann	Schaad
1 Signer	64 + base64	13	13
1 MAC recipient	64 + base64	13	14
1 Encryption Recip	94 + base64	14	14

Array vs Map

- Array cost is 1 byte for empty places
- Map cost is 1 byte for extant places
- Signature body– 4 items 2 will always be filled
- Signature item – 3 items 2 will always be filled
- Encrypt body – 6 items 4 will always be filled
- Recipient – 6 items 2 will always be filled
- Recipient structures are only losers.

Use integer string substitution

- Can do tag substitution only
- Can do both tag and value substitution
  - Can use both positive and negative range with sign as a separation
- Issues on recipients of dealing with both tagging types. Will be needed in the future if integer tagging is assigned to new strings after some rollout

Message type identification

- Use integer for message type – replace “typ” element
- Assumes “typ” will not be used for profiling of the top level structure
- Could use new tagging to replace array element

Separate MAC and Signature

- Easier to discuss security properties of separated
- Easier to allow for key management
- Cleans up terminology
- They just aren’t the same thing!

Single Encryption Structure

- Simplifies code
- Allows for full AEAD key wrap algorithms
- Allows for future change from composite to Chinese menu for algorithm specification

Move Auth Tag to CipherText

- Decreases output size
- Cleaner for algorithms which don’t have a distinct tag
- No indication that more libraries do it one way than the other

Separate Attributes in two layers

- JOSE only supports this for unauthenticated attributes in encryption
- For signature, both layers authenticated by signature
- For mac and encryption, each layer is separately authenticated
- Attributes apply only to the layer they are in

Algorithm Additions/Changes

- AES-CCM-8
  - Also implicit IV?
- HIP work on AES as basic algorithm
  - Use AES in encrypt mode only
  - Try and use MAC mode where hashes would be used
  - HKDF extractor function difficult in 2010
  - Assumes EC or similar for key management

Discussion Questions

- Who wants to actually work on this?
- Preference for which approach
- Preference for forum
  - Individual, JOSE, new working group
- Missing features, controversial features?
- Independence from JOSE documents

Structures for COSE

```
COSE_Sign : {
      protected : bstr | null;
      unprotected : map(tstr, .) | null;
      payload : bstr | null;
      signatures: COSE_signature_a* | COSE_signature;
      }
COSE_signature : {
      protected : bstr | null;
      unprotected : map(tstr, .) | null;
       signature : bstr;
      }
*COSE_signature_a : COSE_signature;
```

Structures for COSE

```
COSE_encrypt {
    protected : bstr | null; # Contains map(tstr, .)
    unprotected : map(tstr, .) | null;
    iv : bstr | null;
    aad : bstr | null;
    ciphertext : bstr | null;
    recipients : COSE_encrypt_a* | COSE_encrypt |
    null; }
* COSE_encrypt_a : COSE_encrypt
```

Structures for COSE

```
COSE_mac : { protected : bstr | null;
    unprotected" : map(tstr, .) | null;
    payload : bstr; tag : bstr;
    recipients : COSE_encrypt_a* | COSE_encrypt | null;
    }
```