HPKE Summary

draft-ietf-cose-hpke

COSE List
IETF 117, San Francisco
July 24, 2023
What Does It Do?

- Enable HPKE to be used with COSE
- Adds support to Keys and Envelopes
Algorithm Discussion

- “alg”
  - present in keys (COSE Key and JSON Web Keys)
  - present in envelopes (COSE_Encrypt and JWE)
Proposal 1: "A la Carte"

```
{
  kty: "EC",
crv: "P-256",
  // alg is optional
  alg: "HPKE-v1-Base(-KEM)",
  // unknown parameters MUST be ignored
  hkc: {
    kem: 0x0010,
    kdfs: [0x0001],
    aeads: [0x0001]
  },
  x: "-eZX…fsxHPc",
y: "BG…JtlnFl",
}
```

// payload: "This is the content", aad: ""
//
16([
  h'a10120', // alg = HPKE-v1-BASE
  {
    4: h'3031', // kid
    -4: [     // HPKE_sender_info
      16,     // kem = DHKEM(P-256, HKDF-SHA256)
      1,      // kdf = HKDF-SHA256
      1,      // aead = AES-128-GCM
      h'048c…e7', // enc
    ],
  },
],
/ encrypted plaintext /
  h'ee2…13',
})

"Envelope" with single recipient & single layer

Key representation
Proposal 2: "Ciphersuites"

```json
{
  kty: "EC",
crv: "P-256",
  // alg is optional
  alg: "TBD_42",
  x: "-eZX...fsxHPc",
y: "BG...JtInFl",
}

// payload: "This is the content", aad: ""
//
// 16([
  h'a10120', // alg = TBD_42
  {        // HPKE_sender_info
    4: h'3031', // kid
    -4: [     // enc
      h'048c...e7', // enc
    ],
  },
  // encrypted plaintext /
  h'ee2...13',
])

Key representation

"Envelope" with single recipient & single layer

TBD_42 = HPKE-v1-BASE + DHKEM(P-256, HKDF-SHA256) + HKDF-SHA256 + AES-128-GCM
Next Steps

● Consensus decision on:
  ● Should “alg” fully specify HPKE using an integer / string (aka ciphersuite approach)?

(Yes / No)
Topic: Encoding of "enc"

- "enc"
  - present in envelopes (COSE_Encrypt and JWE)
  - contains output of HPKE algorithm
Proposal 3: Opaque Enc

// payload: "This is the content", aad: ""

//
16({
  h'a10120', // alg = HPKE-v1-BASE
  { 4: h'3031', // kid
      -4: [
          // HPKE_sender_info
            16, // kem = DHKEM(P-256, HKDF-SHA256)
            1, // kdf = HKDF-SHA256
            1, // aead = AES-128-GCM
            h'048c…e7', // enc
        ],
    },
  // encrypted plaintext
  h'ee2…13',
})

"Envelope" with single recipient & single layer in the style of
Proposal 4: enc represented as COSE Key / JWK

// payload: "This is the content", aad: ""

//
16([
  h'a10120', // alg = TBD_42
  {
    4: h'3031', // kid
    // HPKE_sender_info / Ephemeral public key for the sender
    // enc (represented as COSE Key / JWK)
    -1: {
      kty: "EC",
      crv: "P-256",
      x: "-eZX...fsxHPc",
      y: "BG...JtlnFl",
    }
  },
  // encrypted plaintext
  h'ee2...13',
])

"Envelope" according to https://datatracker.ietf.org/doc/html/draft-ietf-cose-hpke-02
Next Steps

● Consensus decision on:
● Should “enc” be opaque in COSE / JOSE (Yes / No).
Support for more than HPKE Base Mode

- HPKE supports three authenticated variants in addition to the unauthenticated variant.
- COSE-HPKE drafts only supported base mode.
- The complete list of modes is:
  - Base Mode
  - PSK Mode
  - Auth Mode
  - Auth-PSK Mode
Next Steps

- Consensus decision on:
  - Should COSE HPKE specification support all modes (Yes / No)?
Context Information and HPKE

- What information should be included in the key derivation function?
- HPKE offers to pass information via the info construct and the COSE specification has defined a context information structure in Section 5.2 of RFC 9053.

Challenge: COSE-HPKE is not the final "application".
- Hence, there is some optionality.
- Makes interoperability more difficult.

What should be included?
- Nothing: [Empty String for Info Value · Issue #25 · cose-wg/HPKE · GitHub](https://github.com/cose-wg/HPKE/issues/25)
- Content aligned with other documents
- COSE-HPKE specific?
- Something else?
Confidentiality without Integrity

- COSE specifies encryption (COSE_Encrypt) independently from authentication.
- COSE-HPKE currently mandates the use of COSE_Sign1 / COSE_MAC
- Should the confidentiality and integrity be treated independently?
COSE HPKE for COSE_MAC

- COSE-HPKE specification focused on the use of HPKE for use with COSE_Encrypt
- Should it also be specified for COSE_MAC?
Use of HPKE API

- HPKE specification describes two APIs,
  - a one-shot API (see Section 6 of RFC 9180)
  - a generic API (see Section 5.1.1 and 5.2 of RFC 9180)
- COSE-HPKE specification describes the use of the one-shot API.
- The focus and the description of the API is not necessary for the use in COSE.
- Should the specification be API agnostic?
Other Issues?

Please keep track of issues at Issues · cose-wg/HPKE · GitHub