Use of Hybrid Public-Key Encryption (HPKE) with CBOR Object Signing and Encryption (COSE)

draft-ietf-cose-hpke-15

Hannes Tschofenig, Orie Steele, Daisuke Ajitomi, Laurence Lundblade

WGLC Issued

- Chairs issued a WGLC on the document early June.
- Thanks to the respondents
  - Thomas Fossati
  - Aritra Banerjee
  - Michael Prorock
  - Akira Tsukamoto
  - Marco Tiloca
  - Sophie Schmieg
  - Renzo Navas
  - Ilari Liusvaara
  - Liuchunchi (Peter)
- Lots of feedback provided.

Updates

- Draft version -15 provided an update of several issues, including
  - Updated abstract,
  - Fixed incorrectly referenced Enc_structure instead of Recipient_structure,
  - Reference to „LAMPS attack“,
  - Added map of ciphersuite labels to their textual description, and
  - Editorial improvements.
- See diff version here.

Todo List

1. Updating the examples
2. Closing the discussion on providing context info
3. COSE_Mac Use Cases
4. Relationship to the Fully Specified Algorithms draft
5. Multiple invocations of Open() / Seal() on the same context

Todo List

1. Updating the examples
2. Closing the discussion on providing context info
3. COSE_Mac Use Cases
4. ~~Relationship to the Fully Specified Algorithms draft~~
5. ~~Multiple invocations of Open() / Seal() on the same context~~

HPKE algorithsms are fully specified

There are no multiple invocations on the same context.

Examples

- Update examples based on recipient_structure.
- Add PSK-mode example
- Add authentication, for example using COSE_Sign.
- Enhance recipient structures with context information.

Example of recipient structure:

```
[
        / protected h'a10126' / << {
          / alg / 1:42 /  HPKE-4 (Assumed: 42) /
          / salt / -20: 'aabbccddeeffgghh'
          / PartyV identity / -24:'bob@example.com',
         } >>,
        / unprotected / {
          / kid / 4:'11'
        },
          / ek /
          -4: h'd1afbdc95b0e735676f6bca34f
               be50f2822259ac09bfc3c500f1
               4a05de9b2833',
      },
      / ciphertext containing encrypted CEK /
      h'079b443ec6dfcda6a5f8748aff3875146a8ed
        40359e1279b545166385d8d9b59',
],
```

Context Info

- For HPKE Key Encryption Mode the **Recipient_structure** is used as input to the **aad** parameter of the HPKE Seal operation.
- HPKE Seal **info** parameter defaults to the empty string.

```
Recipient_structure = [
    context: "Recipient",
    next_layer_alg: int/tstr,
    recipient_protected_header: empty_or_serialize_map,
    recipient_aad: bstr
]
```

- **next_layer_alg** prevents the “LAMPS attack” (and thereby defines a purpose-built solution for HPKE of cose-cek-hkdf).
- **recipient_aad** contains any additional context the application wishes to protect. Defaults to the empty string.
- **recipient_protected_header** contains the protected headers from the COSE_recipient (see example from previous slide).

COSE_Mac Use Cases

- COSE defines a number of Content Key Distribution Methods.
- These methods establish a symmetric key, which are most commonly used for encrypting payloads.
- COSE HPKE focuses mainly on the encryption feature but also talks about the use of COSE_Mac where the content is only integrity protected but not confidentiality protected.
- It can be used with Base mode and with PSK mode.
- COSE_Mac with long-term key for authentication + COSE HPKE in Base mode establishing the Mac key to protect the payload corresponds to HPKE in PSK mode.
- Do we need both variants? What are the use cases for the unauthenticated variant?

Next Steps

- Authors will take dilligence to address the remaining issues discussed today.
- Update code.
- Submit draft update.
- Have second working group last call (?)