COSE HPKE

Hannes Tschofenig, Russ Housley, Brendan Moran

Status Update

- Call for adoption finished (after some delay) and the initial draft version was submitted as draft-ietf-cose-hpke-00.txt
- Draft repository was created at https://github.com/cose-wg/HPKE/
- A lot of discussions on the list about the content of the -01 version.

Re-use of HPKE Algorithm Registry

Re-use of HPKE Algorithm Registry

- HPKE spec defines three relevant registries:
  - AEAD IDs (AES-128-GCM, AES-256-GCM, ChaCha20Poly1305)
  - KEM IDs (DHKEM(P-256, HKDF-SHA256), DHKEM(X25519, HKDF-SHA256), DHKEM(X448, HKDF-SHA512), etc.)
  - KDF IDs (HKDF-SHA256, HKDF-SHA384, HKDF-SHA512)
- Argument: Whenever a new algorithm is added to the IANA HPKE registry it becomes automatically available also for the COSE HPKE spec.
- Several people asked for it (Goeran, John, Ilari).
- Open issue: where to place the parameters in the overall COSE structure.

Layering of COSE-HPKE

Layer structure in draft-ietf-cose-hpke-00.txt

```
96(                                                                    // COSE ENCRYPT
    [
        h'A10101',                                                     // protected field with alg=AES-GCM-128
        {                                                              // unprotected field with...
            5: h'26682306D4FB28CA01B43B80'                             //     iv
        },
        null,                                                          // detached ciphertext
        [                                                              // recipients array
            h'',                                                       // empty protected field
            {                                                          // unprotected field with ...
                1: -3                                                  //     alg=AES-128-KW
            },
            h'FA55A50CF110908DA6443149F2C2062011A7D8333A72721A',       // CEK encrypted with KEK
            [                                                          // recipients array
                h'A1013818',                                           // protected field with alg=ECDH-ES + HKDF-256
                {                                                      // unprotected field with ...
                  -1: h'A40102200121582 05F...979D5168718766510C445',  //    ephemeral structure
                   4: h'6B69642D31'                                    //    kid
                },
                null                                                   // empty ciphertext
            ]
        ]
    ]
)
```

Proposed layering

- Proposal for new layering structure by Ilari.
  - Smaller size
  - Matches the AES-KW structure (which also has 2 layers only)

```
96(
  [
    // protected field with alg=AES-GCM-128
    h'A10101',
    {   // unprotected field with iv
      5: h'26682306D4FB28CA01B43B80'
    },
    // null because of detached ciphertext
    null,
    [  // COSE_recipient_outer
     / protected / h'a1013818' / {
          \ hpke-alg \ 1:16 \ HPKE/P-256+HKDF-256 \
          \ hpke-aead-id \  -2:1    //    AES-128-GCM
     } / ,
     / unprotected / {
       // HPKE encapsulated key
       / ephemeral / -1:{
            / kty / 1:2,
            / crv / -1:1,
            / x / -2:h'98f50a4ff6c05861c8...90bbf91d6280',
            / y / -3:true
       },
      // kid for recipient static ECDH public key
      / kid / 4:'meriadoc.brandybuck@buckland.example'
       },
       // Encrypted CEK
       h'FA55A50CF110908DA6443149F2C2062011A7D8333A72721A',
       ]
    ]
  ]
)
```

Layer 0

Layer 1

COSE_Key Definition

COSE_Key Definition

- Suggestion by Ilari is to define a new KTY for use of HPKE in the COSE_Key structure.
- Pros:
  - Good idea when other changes to the key format are made as well.
- Con:
  - It is a new structure (although cost is low)
- Open issue: whether a new key format (e.g. based on PQC algorithms) can “automatically” become available also to COSE is unclear because additional, or new parameters may need to be defined.

Extra Features for HPKE KTY structure

- Compressed points (which are currently not available in COSE).
  - Should we only support compressed points or both?
- Ability to carry private keys in the COSE_Key structure
  - Useful for configuration
  - Similar to the approach taken in Encrypted Client Hello (draft-ietf-tls-esni) and PEM file format for ECH (draft-farrell-tls-pemesni)

Next steps

- New draft in time for the IETF meeting
- Reference implementation (to generate examples)
  - Hackathon project?
- Update corresponding SUIT firmware encryption draft.