COSE HPKE

H. Tschofenig, O. Steele, D. Ajitomi, L. Lundblade

draft-ietf-cose-hpke-09

HPKE

- SealBase(pkR, info, aad, pt) function is used to encrypt a plaintext pt to a recipient's public key (pkR).

- The "info" parameter can be used to influence the generation of keys
- The "aad" parameter provides additional authenticated data to the AEAD algorithm in use.

- Slide focuses on „info“ parameter.

History on Context Information COSE

- COSE RFC 9053, Section 5.2 follows what is said in SP800-56A.
- It defines the famous COSE_KDF_Context structure but leaves it open to applications using COSE to populate the structure with meaningful values.
- PartyUInfo.identity and PartyVInfo.identity is describes as containing roles („server“ and „client“) or identities defined by the application layer naming system, such as DNS.
- Parameters can be explicitly carried in the headers of COSE messages, such as salt, nonce or even PartyUInfo and PartyVInfo.

```
PartyInfo = (
    identity : bstr / nil,
    nonce : bstr / int / nil,
    other : bstr / nil
)

COSE_KDF_Context = [
    AlgorithmID : int / tstr,
    PartyUInfo : [ PartyInfo ],
    PartyVInfo : [ PartyInfo ],
    SuppPubInfo : [
        keyDataLength : uint,
        protected : empty_or_serialized_map,
        ? other : bstr
    ],
    ? SuppPrivInfo : bstr
]
```

History on Context Information
draft-ietf-cose-hpke-00

- Re-used COSE RFC-defined COSE_KDF_Context but profiled it
  - PartyUInfo.identity corresponds to the kid found in the COSE_Sign_Tagged or COSE_Sign1_Tagged structure (when a digital signature is used).
  - PartyVInfo.identity corresponds to the kid used for the respective recipient from the inner-most recipients array.
  - The value in the AlgorithmID field corresponds to the alg parameter in the protected structure in the inner-most recipients array.
  - keyDataLength is set to the number of bits of the desired output value.protected refers to the protected structure of the inner-most array.
- Does not mandate COSE_KDF_Context by the

History on Context Information
draft-ietf-cose-hpke-02

- Removed the COSE_KDF_Context.
    - In the absence of an application profile standard specifying otherwise a COSE-HPKE-compliant application MUST use an empty "info" parameter.
- Argument: Easier for interop testing and for developer.

History on Context Information
draft-ietf-cose-hpke-03

- Added the COSE_KDF_Context back again.
- Application-specific profiles of this specification MAY mandate the use of the info and the aad parameters.

History on Context Information
draft-ietf-cose-hpke-04

- Interop problems!
- Enhanced language:
  - Application profiles of this specification MAY populate the fields of the COSE_KDF_Context structure or MAY use a different structure as input to the "info" parameter. If no content for the "info" parameter is not supplied, it defaults to a zero-length byte string.

History on Context Information
draft-ietf-cose-hpke-09

- The COSE_KDF_Context MUST NOT be used in COSE-HPKE.
- Instead, a new structure is defined:
  - next_layer_alg MUST match the alg parameter in the next lower COSE layer.
  - "recipient_protected_header" contains the protected headers from COSE_recipient.
  - "recipient_aad" contains any additional context the application wishes to protect. If none, it is a zero-length string. This is distinct from the external_aad for the whole COSE encrypt. It is per-recipient. Since it is not a header, it may be secret data that is not transmitted.

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

LAMPS

- Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS) - RFC 9629.
- Notes:
  - wrap identifies a key-encryption algorithm used to encrypt the CEK
  - ukm is optional user keying material; For example, the ukm value could include a nonce, application-specific context information, or an identifier for the originator.

```
CMSORIforKEMOtherInfo ::= SEQUENCE {
  wrap KeyEncryptionAlgorithmIdentifier,
  kekLength INTEGER (1..65535),
  ukm [0] EXPLICIT UserKeyingMaterial OPTIONAL }
```

IETF#120 and Mailing List

- See discussion thread here: https://mailarchive.ietf.org/arch/msg/cose/-ayRTpMhr5kHp9ppA91__I3XWE8/
- Lots of input (but some of it is focused on PQC issues)
- Sophie:
  - "you cannot trust the information in the headers“ with the following qualification: “The one example I know of header information safely being used in the verification is the use of a key ID, in the very specific scenario that the key ID allows the selection between multiple different *equally trusted* keys.”
  - Problem: kid is optional parameter and alg is assumed to be present to drive the rest of the processing.
- COSE Key does not allow us to express the encryption and key exchange algorithms a key is supposed to be used for.

Next Steps

- My recommendation:
  - Use a context information structure that relies on data that is known to parties out of band.
  - Ideally rely on COSE Key-encoded information available to both parties.
  - Do not exchange all information inband.
- We need to make it easier for implementers.
- Not only a challenge for COSE HPKE.