Use of HPKE with JOSE

draft-rha-jose-hpke-encrypt

Tirumaleswar Reddy, Hannes Tschofenig, Orie Steele, Aritra Banerjee, Michael B. Jones

IETF 119, Brisbane
What Does It Do?

- HPKE Direct Encryption mode: A kind of “Direct Encryption” to a single recipient.
- HPKE Key Encryption mode: A kind of “Key Encryption” for symmetric encryption to multiple recipients.
Why Do It?

● The HPKE specification provides a variant of public key encryption of arbitrary-sized plaintexts for a recipient public key.

● HPKE (Hybrid Public Key Encryption) emerged in the IETF as a prominent public key encryption scheme
  ▶️ [https://www.rfc-editor.org/rfc/rfc9180.html](https://www.rfc-editor.org/rfc/rfc9180.html) (Developed by CFRG in IRTF)
  ▶️ Used by several protocols Oblivious HTTP, Encrypted Client Hello in TLS, MLS


● HPKE interfaces are friendly to hybrid encryption
Ciphersuite Registration

- The Cipher suite (fully-specified algorithms) approach was finalized for the COSE HPKE draft, rather than the a-la-carte approach
- HPKE-<Mode>-<KEM>-<KDF>-<AEAD>
  - HPKE-Base-P256-SHA256-AES128GCM
- Three authenticated variants including PSK, Auth, and Auth_psk are defined in HPKE
- The "KEM", "KDF", and "AEAD" values are taken from the HPKE IANA registry (Hybrid Public Key Encryption (HPKE) (iana.org))
# JOSE HPKE Serializations and Modes

<table>
<thead>
<tr>
<th>Name</th>
<th>Recipients</th>
<th>Serializations</th>
<th>Content Encryption Key</th>
<th>Similar to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Encryption</td>
<td>1</td>
<td>Compact, JSON</td>
<td>Derived from HPKE</td>
<td>Direct Key Agreement</td>
</tr>
<tr>
<td>Key Encryption</td>
<td>1 or More</td>
<td>Compact, JSON</td>
<td>Encrypted by HPKE</td>
<td>Key Agreement with Key Wrapping</td>
</tr>
</tbody>
</table>
HPKE Encryption

- Invoke SetupBase() to create HPKE context
- In HPKE Direct Encryption mode, the plaintext "pt" passed into Seal() is the content to be encrypted.
- In HPKE Key Encryption mode, the plaintext "pt" passed into Seal() is the CEK.
HPKE Decryption

- Invoke SetupBaseR to create the HPKE context
- Open() to decrypt ciphertext: Output is plaintext or CEK.
HPKE Direct Encryption

```
{
"alg": "dir",
"enc": "HPKE-Base-P256-SHA256-AES128GCM",
"epk": {
    "kty": "EK",
    "ek": "BGNkj…U9thXA"
}
}
```

direct encryption from draft-steele-jose-cose-hpke-cookbook

Sender’s Encapsulated Ephemeral Public Key

HPKE Cipher Suite

Direct Encryption
HPKE Key Encryption

```
{
  "protected": "eyJlbmMiOiJBMTI4R0NNIn0",
  "ciphertext": "S0qqrM3xXPUavbmL9LQ...M57DX3cXCK3TKHqd4hZE8rSNDuUia",
  "iv": "AzaXpooLg3ZxEASQ",
  "aad": "8J-SgCBhYWQ",
  "tag": "S0omWw35S0H7tyEHsmGLDw",
  "recipients": [
    {
      "encrypted_key": "yDVZLs07-ecy_GCgEluwn9U723TCHNAzeYRRQPOfpHM",
      "header": {
        "alg": "HPKE-Base-P256-SHA256-AES128GCM",
        "epk": {
          "kty": "EK",
          "ek": "uPpjglLnXDKg9uQ4kt9tHCs3PUzPxQs"
        }
      }
    }
  ]
}
```

**HPKE Cipher Suite**

**{"enc": "A128GCM"}**

**Sender's Encapsulated Ephemeral Public Key**
Next Steps

- Consider for WG adoption
- Comments and suggestions are welcome