Workgroup: CBOR Object Signing and Encryption
Internet-Draft:
draft-ajitomi-cose-cose-key-jwk-hpke-kem-01
Published: 4 February 2023
Intended Status: Standards Track
Expires: 8 August 2023
Authors: D. Ajitomi
Independent

COSE Key and JSON Web Key Representation for Key Encapsulation Mechanism (KEM) of Hybrid Public Key Encryption (HPKE)

Abstract

This document defines an additional key parameter and a new key type for CBOR Object Signing and Encryption (COSE) Key and JSON Web Key (JWK) to represent a Key Encapsulated Mechanism (KEM) key and its associated information for Hybrid Public Key Encryption (HPKE).

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://dajiaji.github.io/i-d-cose-cose-key-jwk-hpke-kem/draft-ajitomi-cose-cose-key-jwk-hpke-kem.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ajitomi-cose-cose-key-jwk-hpke-kem/.

Discussion of this document takes place on the CBOR Object Signing and Encryption Working Group mailing list (mailto:cose@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/cose/. Subscribe at https://www.ietf.org/mailman/listinfo/cose/.

Source for this draft and an issue tracker can be found at https://github.com/dajiaji/i-d-cose-cose-key-jwk-hpke-kem.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Hybrid Public Key Encryption (HPKE) [RFC9180], published by the Internet Research Task Force (IRTF), has already been adopted in several communication protocol specifications such as TLS Encrypted Client Hello (ECH), Oblivious DNS over HTTPS (ODoH) and Oblivious HTTP (OHTTP). HPKE itself is communication protocol independent and can be widely used as a standard scheme for public key based end-to-end encryption in various applications, not only in communication protocols.

In HPKE, the sender of a ciphertext needs to know in advance not only the recipient public key, but also the HPKE mode, the KEM associated with the key, and the set of supported KDF and AEAD algorithms. The data structure of this information (hereafter referred to as HPKE key configuration information) is defined in each communication protocol specification that uses HPKE. For example, the ECH defines it as a structure called HpkeKeyConfig. When using HPKE in an application, it is necessary to define the data structure corresponding to the HpkeKeyConfig and how the information is transferred from the recipient to the sender. If the data structure and the publication method for the HPKE key configuration information were standardized, it would be easier to use HPKE in applications.

This document defines how to represent a KEM key for HPKE and the HPKE key configuration information in JSON Web Key (JWK) [RFC7517] and COSE_Key defined in CBOR Object Signing and Encryption (COSE) Structures and Process [RFC9052]. Specifically, this document defines (1) a common key parameter for defining the HPKE key configuration information in existing key types that can be used for key derivation and (2) a generic key type for HPKE that can also be used to represent post-quantum KEM keys to be specified in the future. By using the generic key type for HPKE, all KEM keys registered in the IANA HPKE registry can be represented in JWK and COSE_Key without the need to define cryptographic algorithm-specific key types and parameters such as for EC or RSA as defined in [RFC7518] and [RFC9053].

The ability to include HPKE-related information in JWK, which is widely used not only as the public key representation but also as the key publication method (via the JWK Set endpoint) at the application layer, and its binary representation, COSE_Key, will facilitate the use of HPKE in a wide variety of web applications and communication systems for constrained devices.

2. Conventions and Definitions

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Common Key Parameter for HPKE Key Configuration

The HPKE key configuration information is defined as a common key parameter of JWK and COSE_Key. The parameter can be specified in the key that can be used for key derivation. In addition, the handling of existing key parameters is also defined.

3.1. JWK Parameter

3.1.1. "hkc" (HPKE Key Configuration) Parameter

The "hkc" (KPKE key configuration) parameter identifies the KEM for the recipient key and the set of KDF and AEAD algorithms supported by the recipient. A JWK used for HPKE KEM **MUST** have this parameter. It **MUST** contain the object consisting of the following three attributes.

- "kem": The HPKE KEM identifier, which is a two-byte value registered in the IANA HPKE registry.

- "kdfs": The array of the HPKE KDF identifiers supported by the recipient. The KDF identifier is also a two-byte value registered in the IANA HPKE registry.

- "aeads": The array of the HPKE AEAD identifiers supported by the recipient. The AEAD identifier is also a two-byte value registered in the IANA HPKE registry.

The "hkc" parameter can be used with existing "EC" [RFC7518] and "OKP" [RFC8037] keys and the keys for future post-quantum KEMs.

3.1.2. Restrictions on the Use of Existing Key Parameters

The restrictions on the use of existing common key parameters in a JWK for HPKE KEM are as follows:

- "alg": The parameter **MUST** be present and contains one of the following values:
  - "HPKE-v1-Base"
  - "HPKE-v1-PSK"
  - "HPKE-v1-Auth"
  - "HPKE-v1-AuthPSK"
- "use": The parameter **SHOULD NOT** be specified. If specified, it **MUST** be "enc".
- "key_ops": The parameter **SHOULD NOT** be specified. If specified, it **MUST** include "deriveKey" and/or "deriveBits".
- etc.

3.2. COSE Key Common Parameter

3.2.1. hkc (HPKE Key Configuration) Parameter

The HPKE key configuration parameter for COSE_Key is defined as follows:

- hkc (HPKE Key Configuration): The parameter **MUST** contain an array structure named HPKE_Key_Configuration, which contains the same information as "hkc" in JWK above. The CDDL grammar describing the HPKE_Key_Configuration structure is:

```
HPKE_Key_Configuration = [
    kem: uint,               ; KEM identifier
    kdfs: uint / [+uint],    ; KDF identifiers
    aeads: uint / [+uint],   ; AEAD identifiers
]
```

Name	CBOR Type	Value Registry	Description
kem	uint	HPKE KEM Identifiers	The KEM identifier bound to the key
kdfs	uint / [+uint]	HPKE KDF Identifiers	The KDF identifiers supported by the recipient
aeads	uint / [+uint]	HPKE AEAD Identifiers	The AEAD identifiers supported by the recipient

Table 1: HPKE Key Configuration Parameters

The hkc parameter can be used with existing OKP and EC2 keys [RFC9053] and the keys for future post-quantum KEMs.

3.2.2. Restrictions on the Use of Existing Key Parameters

The restrictions on the use of existing common key parameters in a COSE_Key for the HPKE KEM are as follows:

- *alg(3): The parameter **MUST** be present and contains one of the following values:
  - -HPKE-v1-Base (T.B.D.)
  - -HPKE-v1-PSK (T.B.D.)
  - -HPKE-v1-Auth (T.B.D.)
  - -HPKE-v1-AuthPSK (T.B.D.)
- *key_ops(4): The parameter **SHOULD NOT** be specified. If specified, it **MUST** include "derive key"(7) and/or "derive bits"(8).
- *etc.

4. Generic Key Type for HPKE KEM

A generic key type for the HPKE KEM keys including a post-quantum KEM defined in the future is defined. Even KEM keys that can be represented by existing key types can use the generic key type defined here.

4.1. Key Type for JWK

A new generic key type (kty) value "HPKE-KEM" is defined to represent the private and public key used for the HPKE KEM. A key with this kty has the following parameters:

- *The parameter "kty" **MUST** be "HPKE-KEM".
- *The parameter "hkc" **MUST** be present and contains the HPKE Key Configuration defined in Section 3.1.1.
- *The parameter "pub" **MUST** be present and contains the public key encoded using the base64url [RFC4648] encoding.
- *The parameter "priv" **MUST** be present if the key is private key and contains the private key encoded using the base64url [RFC4648] encoding.

4.2. Key Type for COSE_Key

A new generic kty(1) value HPKE-KEM(T.B.D.) is defined to represent the private and public key used for the HPKE KEM. A key with this kty has the following parameters:

- The parameter kty(1) **MUST** be HPKE-KEM(T.B.D).

- The parameter hkc(T.B.D.) **MUST** be present and contains the HPKE Key Configuration defined in Section 3.2.1.

- The parameter pub(-1) **MUST** be present and contains the public key encoded in a byte string (bstr type).

- The parameter priv(-2) **MUST** be present if the key is private key and contains the private key encoded in a byte string (bstr type).

5. Security Considerations

TODO

6. IANA Considerations

TODO

7. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/rfc/rfc7517>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/rfc/rfc7518>.

[RFC8037] Liusvaara, I., "CFRG Elliptic Curve Diffie-Hellman (ECDH) and Signatures in JSON Object Signing and Encryption (JOSE)", RFC 8037, DOI 10.17487/RFC8037, January 2017, <https://www.rfc-editor.org/rfc/rfc8037>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC9052]
Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/rfc/rfc9052>.

[RFC9053] Schaad, J., "CBOR Object Signing and Encryption (COSE): Initial Algorithms", RFC 9053, DOI 10.17487/RFC9053, August 2022, <https://www.rfc-editor.org/rfc/rfc9053>.

[RFC9180] Barnes, R., Bhargavan, K., Lipp, B., and C. Wood, "Hybrid Public Key Encryption", RFC 9180, DOI 10.17487/RFC9180, February 2022, <https://www.rfc-editor.org/rfc/rfc9180>.

Appendix A. Examples

A.1. JWK for DHKEM(P-256, KDF-SHA256) Public Key with Key Type "EC"

```
{
    "kty": "EC",
    "kid": "01",
    "crv": "P-256",
    "alg": "HPKE-v1-Base",
    "hkc": {
        "kem": 0x010,
        "kdfs": [0x001, 0x002, 0x003],
        "aeads": [0x001, 0x002]
    },
    "x": "-eZXC6nV-xgthy8zZMCN8pcYSeE2XfWWqckA2fsxHPc",
    "y": "BGU5soLgsu_y7GN2I3EPUXS9EZ7Sw0qif-V70JtInFI"
}
```

A.2. JWK for DHKEM(X25519, KDF-SHA256) Public Key with Key Type "OKP"

```
{
    "kty": "OKP",
    "kid": "01",
    "crv": "X25519",
    "alg": "HPKE-v1-Base",
    "hkc": {
        "kem": 0x020,
        "kdfs": [0x001, 0x002, 0x003],
        "aeads": [0x001, 0x002]
    },
    "x": "y3wJq3uXPHeoCO4FubvTc7VcBuqpvUrSvU6ZMbHDTCI"
}
```

A.3. JWK for DHKEM(X25519, KDF-SHA256) Private Key with Key Type "HPKE-KEM"

```
{
    "kty": "HPKE-KEM",
    "kid": "01",
    "alg": "HPKE-v1-Base",
    "hkc": {
        "kem": 0x020,
        "kdfs": [0x001, 0x002, 0x003],
        "aeads": [0x001, 0x002]
    },
    "pub": "y3wJq3uXPHeoCO4FubvTc7VcBuqpvUrSvU6ZMbHDTCI",
    "priv": "vsJ1oX5NNi0IGdwGldiac75r-Utmq3Jq4LGv48Q_Qc4"
}
```

A.4. COSE_Key for DHKEM(P-256, KDF-SHA256) Public Key with Key Type EC2(2)

```
{
    1:2,           // EC2
    2:'01',
    3:-1(T.B.D),  // HPKE-v1-Base
    -1:1,          // P-256
    6(T.B.D): [   // hkc (HPKE Key Configuration)
        0x0010,                    // KEM identifier
        [0x0001, 0x0002, 0x0003],  // supported KDF identifiers
        [0x0001, 0x0002]           // supported AEAD identifiers
    ],
    -2:h'65eda5a12577c2bae829437fe338701a10aaa375e1bb5b5de108de439c08551
    -3:h'1e52ed75701163f7f9e40ddf9f341b3dc9ba860af7e0ca7ca7e9eecd0084d19
}
```

A.5. COSE_Key for DHKEM(X25519, KDF-SHA256) Public Key with Key Type OKP(1)

```
{
    1:1,           // OKP
    2:'01',
    3:-1(T.B.D),  // HPKE-v1-Base
    -1:4,          // X25519
    6(T.B.D): [   // hkc (HPKE Key Configuration)
        0x0020,                    // KEM identifier
        [0x0001, 0x0002, 0x0003],  // supported KDF identifiers
        [0x0001, 0x0002]           // supported AEAD identifiers
    ],
    -2:h'd75a980182b10ab7d54bfed3c964073a0ee172f3daa62325af021a68f707511
}
```

A.6. COSE_Key for DHKEM(X25519, KDF-SHA256) Private Key with Key Type HPKE-KEM(T.B.D)

```
{
    1:-1(T.B.D.),  // HPKE-KEM
    2:'01',
    3:-1(T.B.D),   // HPKE-v1-Base
    6(T.B.D): [    // hkc (HPKE Key Configuration)
        0x0020,                     // KEM identifier
        [0x0001, 0x0002, 0x0003],  // supported KDF identifiers
        [0x0001, 0x0002]           // supported AEAD identifiers
    ],
    -1:h'd75a980182b10ab7d54bfed3c964073a0ee172f3daa62325af021a68f707511
    -2:h'9d61b19deffd5a60ba844af492ec2cc44449c5697b326919703bac031cae7f6
}
```

Acknowledgments

We would like to thank Ilari Liusvaara, Laurence Lundblade and Orie Steele for their review feedback.

Author's Address

Daisuke Ajitomi
Independent

Email: dajiaji@gmail.com