Workgroup: COSE
Internet-Draft: draft-ietf-cose-hpke-00
Published: 10 January 2022
Intended Status: Standards Track
Expires: 14 July 2022
Authors: H. Tschofenig
Arm Limited
R. Housley
Vigil Security
B. Moran
Arm Limited

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

Abstract

This specification defines hybrid public-key encryption (HPKE) for use with CBOR Object Signing and Encryption (COSE).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 14 July 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November

10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction

Hybrid public-key encryption (HPKE) [I-D.irtf-cfrg-hpke] is a scheme that provides public key encryption of arbitrary-sized plaintexts given a recipient's public key. HPKE utilizes a non-interactive ephemeral-static Diffie-Hellman exchange to establish a shared secret, which is then used to encrypt plaintext.

The HPKE specification defines several features for use with public key encryption and a subset of those features is applied to COSE [RFC8152]. Since COSE provides constructs for authenticcation, those are not re-used from the HPKE specification. This specification uses the "base" mode (as it is called in HPKE specification language).

2. Conventions and Terminology

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.

This specification uses the following abbreviations and terms: - Content-encryption key (CEK), a term defined in RFC 2630 [RFC2630]. - Hybrid Public Key Encryption (HPKE) is defined in [I-D.irtf-cfrg-hpke]. - pkR is the public key of the recipient, as defined in [I-D.irtf-cfrg-hpke]. - skR is the private key of the recipient, as defined in [I-D.irtf-cfrg-hpke].

3. HPKE for COSE

3.1. Overview

The CDDL for the COSE_Encrypt structure, as used with this specification, is shown in Figure 1. The structures referenced below are found in the CDDL.

HPKE, when used with COSE, follows a three layer structure:

* Layer 0 (corresponding to the COSE_Encrypt structure) contains content encrypted with the CEK. This ciphertext may be detached. If not detached, then it is included in the COSE_Encrypt structure.

* Layer 1 (see COSE_recipient_outer structure) includes the encrypted CEK.

* Layer 2 (in the COSE_recipient_inner structure) contains parameters needed for HPKE to generate a shared secret used to encrypt the CEK from layer 1.

```
COSE_Encrypt_Tagged = #6.96(COSE_Encrypt)

SUIT_Encryption_Info = COSE_Encrypt_Tagged

; Layer 0
COSE_Encrypt = [
  Headers,
  ciphertext : bstr / nil,
  recipients : [+COSE_recipient_outer]
]

; Layer 1
COSE_recipient_outer = [
  protected   : bstr .size 0,
  unprotected : header_map, ; must contain alg
  encCEK      : bstr, ; CEK encrypted with HPKE-derived shared secret
  recipients  : [ + COSE_recipient_inner ]
]

; Layer 2
COSE_recipient_inner = [
  protected   : bstr .cbor header_map, ; must contain HPKE alg
  unprotected : header_map, ; must contain kid and ephemeral public key
  empty       : null,
  empty       : null
]

header_map = {
  Generic_Headers,
  * label =values,
}
```

Figure 1: CDDL for HPKE-based COSE_Encrypt Structure

The COSE_recipient_outer structure shown in Figure 1 includes the encrypted CEK (in the encCEK structure) and the COSE_recipient_inner structure, also shown in Figure 1, contains the ephemeral public key (in the unprotected structure).

3.2. HPKE Encryption with SealBase

The SealBase(pkR, info, aad, pt) function is used to encrypt a plaintext pt to a recipient's public key (pkR). For use in this specification, the plaintext "pt" passed into the SealBase is the CEK. The CEK is a random byte sequence of length appropriate for the encryption algorithm selected in layer 0. For example, AES-128-GCM requires a 16 byte key and the CEK would therefore be 16 bytes long.

The "info" parameter can be used to influence the generation of keys and the "aad" parameter provides additional authenticated data to

the AEAD algorithm in use. If successful, SealBase() will output a ciphertext "ct" and an encapsulated key "enc". The content of enc is the ephemeral public key.

The content of the info parameter is based on the 'COSE_KDF_Context' structure, which is detailed in Figure 2.

3.3. HPKE Decryption with Open

The recipient will use the OpenBase(enc, skR, info, aad, ct) function with the enc and ct parameters received from the sender. The "aad" and the "info" parameters are obtained via the context of the usage.

The OpenBase function will, if successful, decrypt "ct". When decrypted, the result will be the CEK. The CK is the symmetric key used to decrypt the ciphertext in the COSE_Encrypt structure.

3.4. Info Structure

This specification re-uses the context information structure defined in [RFC8152] for use with the HPKE algorithm. This payload becomes the content of the info parameter for the HPKE functions. For better readability of this specification the COSE_KDF_Context structure is repeated in Figure 2.

```
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
]
```

Figure 2: COSE_KDF_Context Data Structure for info parameter

Since this specification may be used in a number of different deployment environments flexibility for populating the fields in the COSE_KDF_Context structure is provided.

For better interoperability, the following recommended settings are provided:

* PartyUInfo.identity corresponds to the kid found in the COSE_Sign_Tagged or COSE_Sign1_Tagged structure (when a digital signature is used). When utilizing a MAC, then the kid is found in the COSE_Mac_Tagged or COSE_Mac0_Tagged structure.

* 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.

4. Example

An example of the COSE_Encrypt structure using the HPKE scheme is shown in Figure 3. It uses the following algorithm combination:

* AES-GCM-128 for encryption of detached ciphertext.

* AES-GCM-128 for encryption of the CEK.

* Key Encapsulation Mechanism (KEM): NIST P-256

* Key Derivation Function (KDF): HKDF-SHA256

```
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
            h'',          // empty protected field
            {             // unprotected field with ...
                 1: 1     //     alg=A128GCM
            },
            // Encrypted CEK
            h'FA55A50CF110908DA6443149F2C2062011A7D8333A72721A',
            / recipients / [  // COSE_recipient_inner
             [
               / protected / h'a1013818' / {
                   \ alg \ 1:TBD1 \ HPKE/P-256+HKDF-256 \
                 } / ,
               / 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'
               },
               // empty ciphertext
               / ciphertext / h''
             ]
            ]
        ]
     ]
)
```

Figure 3: COSE_Encrypt Example for HPKE

5. Security Considerations

This specification is based on HPKE and the security considerations of HPKE [I-D.irtf-cfrg-hpke] are therefore applicable also to this specification.

HPKE assumes that the sender is in possession of the public key of the recipient. A system using HPKE COSE has to assume the same assumptions and public key distribution mechanism is assumed to exist.

Since the CEK is randomly generated it must be ensured that the guidelines for random number generations are followed, see [RFC8937].

The SUIT_Encryption_Info structure shown in this document does not provide authentication. Hence, the SUIT_Encryption_Info structure has to be used in combination with other COSE constructs, such as the COSE_Sign or COSE_Sign1.

6. IANA Considerations

This document requests IANA to create new entries in the COSE Algorithms registry established with [RFC8152].

Name	Value	KDF	Ephemeral-Static	Key Wrap	Description
HPKE/P-256+HKDF-256	TBD1	HKDF - SHA-256	yes	none	HPKE with ECDH-ES (P-256) + HKDF-256
HPKE/P-384+HKDF-SHA384	TBD2	HKDF - SHA-384	yes	none	HPKE with ECDH-ES (P-384) + HKDF-384
HPKE/P-521+HKDF-SHA521	TBD3	HKDF - SHA-521	yes	none	HPKE with ECDH-ES (P-521) + HKDF-521
HPKE X25519 + HKDF-SHA256	TBD4	HKDF - SHA-256	yes	none	HPKE with ECDH-ES (X25519) + HKDF-256
HPKE X448 + HKDF-SHA512	TBD4	HKDF - SHA-512	yes	none	HPKE with ECDH-ES (X448) + HKDF-512

7. References

7.1. Normative References

[I-D.irtf-cfrg-hpke] Barnes, R. L., Bhargavan, K., Lipp, B., and C. A. Wood, "Hybrid Public Key Encryption", Work in Progress, Internet-Draft, draft-irtf-cfrg-hpke-12, 2 September 2021, <https://www.ietf.org/archive/id/draft-irtf-cfrg-hpke-12.txt>.

[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/info/rfc2119>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

[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/info/rfc8174>.

7.2. Informative References

[RFC2630] Housley, R., "Cryptographic Message Syntax", RFC 2630, DOI 10.17487/RFC2630, June 1999, <https://www.rfc-editor.org/info/rfc2630>.

[RFC8937] Cremers, C., Garratt, L., Smyshlyaev, S., Sullivan, N., and C. Wood, "Randomness Improvements for Security Protocols", RFC 8937, DOI 10.17487/RFC8937, October 2020, <https://www.rfc-editor.org/info/rfc8937>.

Appendix A. Acknowledgements

TBD: Add your name here.

Authors' Addresses

Hannes Tschofenig
Arm Limited

Email: hannes.tschofenig@arm.com

Russ Housley
Vigil Security, LLC

Email: housley@vigilsec.com

Brendan Moran
Arm Limited

Email: Brendan.Moran@arm.com