

Workgroup: Delay-Tolerant Networking  
Internet-Draft: draft-bsipos-dtn-bpsec-cose-03  
Published: 28 October 2020  
Intended Status: Standards Track  
Expires: 1 May 2021  
Authors: B. Sipos

RKF Engineering

## **DTN Bundle Protocol Security COSE Security Contexts**

### **Abstract**

This document defines a security context suitable for using CBOR Object Signing and Encryption (COSE) algorithms within Bundle Protocol Security (BPsec) integrity and confidentiality blocks. A profile of COSE is also defined for BPsec interoperation.

### **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 1 May 2021.

### **Copyright Notice**

Copyright (c) 2020 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

- [1. Introduction](#)
- [2. Requirements Language](#)
- [3. BPSec Security Context](#)
  - [3.1. COSE Integrity](#)
  - [3.2. COSE Confidentiality](#)
- [4. COSE Profile for BPSec](#)
  - [4.1. COSE Security Parameters](#)
  - [4.2. COSE Security Results](#)
  - [4.3. Interoperability Algorithms](#)
- [5. Implementation Status](#)
- [6. Security Considerations](#)
  - [6.1. Threat: BPSec Block Replay](#)
  - [6.2. Threat: Unidentifiable Key](#)
  - [6.3. Threat: Algorithm Vulnerabilities](#)
- [7. IANA Considerations](#)
  - [7.1. BPSec Security Contexts](#)
- [8. Acknowledgments](#)
- [9. References](#)
  - [9.1. Normative References](#)
  - [9.2. Informative References](#)
- [Appendix A. Examples](#)
  - [A.1. Symmetric Key COSE Mac0](#)
  - [A.2. RSA Keypair COSE Sign1](#)
  - [A.3. Symmetric Key COSE Encrypt0](#)
  - [A.4. Symmetric KEK COSE Encrypt](#)
- [Author's Address](#)

## 1. Introduction

The Bundle Protocol Security (BPSec) Specification [[I-D.ietf-dtn-bpsec](#)] defines structure and encoding for Block Integrity Block (BIB) and Block Confidentiality Block (BCB) types but does not specify any security contexts to be used by either of the security block types. The CBOR Object Signing and Encryption (COSE) specification [[RFC8152](#)] defines a structure, encoding, and algorithms to use for cryptographic signing and encryption.

This document describes how to use the algorithms and encodings of COSE within BPSec blocks to apply those algorithms to Bundle security in [Section 3](#). A bare minimum of interoperability algorithms and algorithm parameters is specified by this document in [Section 4](#).

This document does not address how those COSE algorithms are intended to be used within a larger security context. Examples of specific uses are provided in [Appendix A](#) to aid in implementation support of the interoperability algorithms.

## 2. Requirements Language

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. BPSec Security Context

This document specifies a single security context for use in both BPSec integrity and confidentiality blocks. This is done to save code points allocated to this specification and to simplify the encoding of COSE-in-BPSec; the BPSec block type uniquely defines the acceptable COSE messages which can be present and each COSE message is type-tagged to indicate its purpose and contents.

The COSE security context shall have the Security Context ID specified in [Section 7.1](#).

The existing CoAP Content-Format ID values in the CoRE registry [[IANA-CORE](#)] SHALL be used as BPSec Parameter ID and Result ID values within COSE security context (see tables within [Section 3.1](#) and [Section 3.2](#)). For Result ID values used to identify COSE messages, these code points are also identical to the existing COSE message-marking tags in Section 2 of [[RFC8152](#)]. This avoids the need for value-mapping between code points of the two registries.

When embedding COSE messages, the CBOR-tagged form SHALL NOT be used. The Result ID values already provide the same information as the COSE tags (using the same code points).

### 3.1. COSE Integrity

When used within a Block Integrity Block, COSE context SHALL allow only the Parameter IDs from [Table 1](#). Each integrity parameter value SHALL consist of the COSE structure indicated by [Table 1](#) in its decoded form.

Parameter ID	Parameter Structure	Reference
101	COSE_Key	<a href="#">[RFC8152]</a>
102	COSE_KeySet	<a href="#">[RFC8152]</a>

Table 1: COSE Integrity Parameters

When used within a Block Integrity Block, COSE context SHALL allow only the Result IDs from [Table 2](#). Each integrity result value SHALL consist of the COSE message indicated by [Table 2](#) in its decoded form.

Result ID	Result Structure	Reference
97	COSE_Mac	[RFC8152]
17	COSE_Mac0	[RFC8152]
98	COSE_Sign	[RFC8152]
18	COSE_Sign1	[RFC8152]

Table 2: COSE Integrity Results

Each integrity result SHALL use the "detached" payload form with nil payload value. The integrity result for COSE\_Mac and COSE\_Mac0 messages are computed by the procedure in Section 6.3 of [RFC8152]. The integrity result for COSE\_Sign and COSE\_Sign1 messages are computed by the procedure in Section 4.4 of [RFC8152].

[NOTE: This differs from base BPsec in that the entire block and the bundle primary is signed] The COSE "payload" used to generate a signature or MAC result SHALL be the canonically serialized target block, including the canonical block array structure. The COSE "protected attributes from the application" used to generate a signature or MAC result SHALL be either:

**For a primary block target:** An empty byte string.

**For a canonical block target:** The canonically serialized primary block of the bundle.

### 3.2. COSE Confidentiality

When used within a Block Confidentiality Block, COSE context SHALL allow only the Parameter IDs from Table 3. Each integrity parameter value SHALL consist of the COSE structure indicated by Table 3 in its decoded form.

Parameter ID	Parameter Structure	Reference
101	COSE_Key	[RFC8152]
102	COSE_KeySet	[RFC8152]

Table 3: COSE Integrity Parameters

When used within a Block Confidentiality Block, COSE context SHALL allow only the Result IDs from Table 4. Each confidentiality result value SHALL consist of the COSE message indicated by Table 4 in its decoded form.

Result ID	Result Structure	Reference
96	COSE_Encrypt	[RFC8152]
16	COSE_Encrypt0	[RFC8152]

Table 4: COSE Confidentiality Results

Only algorithms which support Authenticated Encryption with Authenticated Data (AEAD) SHALL be usable in the first (content) layer of a confidentiality result. Because COSE encryption with AEAD appends the authentication tag with the ciphertext, the size of the block-type-specific-data will grow after an encryption operation.

Each confidentiality result SHALL use the "detached" payload form with nil payload value. The COSE plaintext and ciphertext correspond exactly with the target block-type-specific-data. The confidentiality result for COSE\_Encrypt and COSE\_Encrypt0 messages are computed by the procedure in Section 5.3 of [[RFC8152](#)].

[NOTE: This differs from base BPSec in that AAD from the block and the bundle primary is used] The COSE "plaintext" used to generate an encrypt result SHALL be the block-type-specific-data of the target block, the decoded byte string itself (not including the encoded CBOR item header). The COSE "protected attributes from the application" used to generate an encrypt result SHALL be the concatenation of the following:

1. The canonically serialized primary block of the bundle.
2. The canonically serialized augmented target block, which has its block-type-specific-data substituted with an empty byte string.

#### **4. COSE Profile for BPSec**

This section contains requirements which apply to the use of COSE within BPSec across any security context use.

##### **4.1. COSE Security Parameters**

When necessary to support public key infrastructure (PKI) within BPSec, a BIB or BCB with a COSE context MAY contain one or more public keys or key identifiers. Because each context contains a single set of parameters which apply to all results in the same context, security acceptors SHALL treat all COSE keys as being related to the security source itself and potentially applying to every result.

##### **4.2. COSE Security Results**

When generating a BPSec result, security sources SHALL use encode COSE labels with a uint value. When processing a BPSec result, security acceptors MAY handle COSE labels with with a tstr value.

When used in a BPSec result, each COSE message SHALL contain an explicit algorithm identifier in the lower (content) layers. When available and not implied by the bundle source, a COSE message

SHOULD contain a key identifier in the highest (recipient) layer. When a key identifier is not available, BPsec acceptors SHOULD use the Security Source (if available) and the Bundle Source to imply which keys can be used for security operations. A BPsec security operation always occurs within the context of the immutable primary block with its parameters (specifically the Source Node ID) and the security block with its optional Security Source.

The algorithms required by this profile focuses on networks using shared symmetric-keys, with recommended algorithms for Elliptic Curve (EC) keypairs and RSA keypairs. The focus of this profile is to enable interoperation between security sources and acceptors on an open network, where more explicit COSE parameters make it easier for BPsec acceptors to avoid assumptions and avoid out-of-band parameters. The requirements of this profile still allow the use of potentially not-easily-interoperable algorithms and message/recipient configurations for use by private networks, where message size is more important than explicit COSE parameters.

#### 4.3. Interoperability Algorithms

[NOTE: The required list is identical to the [[I-D.ietf-dtn-bpsec-interop-sc](#)] list.] The set of integrity algorithms needed for interoperability is listed here. The full set of COSE algorithms available is managed at [[IANA-COSE](#)].

Implementations conforming to this specification SHALL support the symmetric keyed algorithms of [Table 5](#). Implementations capable of doing so SHOULD support the asymmetric keyed and key-encryption algorithms of [Table 5](#).

BPsec Block	COSE Layer	Name	Code	Implementation Requirements
Integrity	1	HMAC 256/256	5	Required
Integrity	1	ES256	-7	Recommended
Integrity	1	EdDSA	-8	Recommended
Integrity	1	PS256	-37	Recommended
Confidentiality	1	A256GCM	3	Required
Integrity or Confidentiality	2	A256KW	-5	Recommended
Integrity or Confidentiality	2	ECDH-ES + A256KW	-31	Recommended
Integrity or Confidentiality	2	RSAES-OAEP w/ SHA-256	-41	Recommended

Table 5: Interoperability Algorithms

The following are recommended key and recipient uses within COSE/ BPsec:

**Symmetric Key Integrity:**

When generating a BIB result from a symmetric key, implementations SHOULD use either a COSE\_Mac0 or a COSE\_Mac using the private key directly. When a COSE\_Mac is used with a direct key, the recipient layer SHOULD include a key identifier.

**EC Keypair Integrity:** When generating a BIB result from an EC keypair, implementations SHOULD use either a COSE\_Sign1 or a COSE\_Sign using the private key directly or a COSE\_Mac from a symmetric key with a layer-2 encryption of the symmetric key. When a COSE\_Sign or COSE\_Mac is used with EC keypair, the recipient layer SHOULD include a public key identifier.

**RSA Keypair Integrity:** When generating a BIB result from an RSA keypair, implementations SHOULD use either a COSE\_Sign1 or a COSE\_Sign using the private key directly or a COSE\_Mac from a symmetric key with a layer-2 key-wrap of the symmetric key. When a COSE\_Sign or COSE\_Mac is used with RSA keypair, the recipient layer SHOULD include a public key identifier. When a COSE\_Sign or COSE\_Sign1 is used with RSA keypair, the signature uses a maximum-length PSS salt in accordance with [[RFC8230](#)].

**Symmetric Key Confidentiality:** When generating a BCB result from an symmetric key, implementations SHOULD use a COSE\_Encrypt message with a recipient containing a key-wrapped CEK. When generating a BCB result from a symmetric key, implementations SHOULD NOT use COSE\_Encrypt0 or COSE\_Encrypt with direct content encryption key (CEK). Doing so risks key overuse and the vulnerabilities associated with large amount of ciphertext from the same key.

**EC Keypair Confidentiality:** When generating a BCB result from an EC keypair, implementations SHOULD use a COSE\_Encrypt message with a recipient containing a key-wrapped CEK.

**RSA Keypair Confidentiality:** When generating a BCB result from an RSA keypair, implementations SHOULD use a COSE\_Encrypt message with a recipient containing a key-wrapped CEK.

## 5. Implementation Status

[NOTE to the RFC Editor: please remove this section before publication, as well as the reference to [[RFC7942](#)] and [[github-dtn-bpsec-cose](#)].]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [[RFC7942](#)]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing

drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations can exist.

An example implementation of COSE over Blocks has been created as a GitHub project [[github-dtn-bpsec-cose](#)] and is intended to use as a proof-of-concept and as a possible source of interoperability testing. This example implementation only handles CBOR encoding/decoding and cryptographic functions, it does not construct actual BIB or BCB and does not integrate with a BP Agent.

## **6. Security Considerations**

This section separates security considerations into threat categories based on guidance of BCP 72 [[RFC3552](#)].

All of the security considerations of the underlying BPSec [[I-D.ietf-dtn-bpsec](#)] apply to these new security contexts.

### **6.1. Threat: BPSec Block Replay**

The bundle's primary block contains fields which uniquely identify a bundle: the Source Node ID, Creation Timestamp, and fragment parameters (see Section 4.2.2 of [[I-D.ietf-dtn-bpbis](#)]). These same fields are used to correlate Administrative Records with the bundles for which the records were generated. Including the primary block in the AAD for BPSec integrity and confidentiality binds the verification of the secured block to its parent bundle and disallows replay of any block with its BIB or BCB.

This profile of COSE limits the encryption algorithms to only AEAD in order to include the context of the encrypted data as AAD. If an agent mistakenly allows the use of non-AEAD encryption when decrypting and verifying a BCB, the possibility of block replay attack is present.

### **6.2. Threat: Unidentifiable Key**

The profile in [Section 4.3](#) recommends key identifiers when possible and the parameters in [Section 4.1](#) allow encoding public keys where available. If the application using a COSE Integrity or COSE Confidentiality context leaves out key identification data (in a COSE recipient structure), the security acceptor for those BPSec blocks only has the primary block available to use when verifying or decrypting the target block. This leads to a situation, identified



in BPSec Security Considerations, where a signature is verified to be valid but not from the expected Security Source.

### 6.3. Threat: Algorithm Vulnerabilities

Because this use of COSE leaves the specific algorithms chosen for BIB and BCB use up to the applications securing bundle data, it is important to use only COSE algorithms which are marked as recommended in the IANA registry [[IANA-COSE](#)].

## 7. IANA Considerations

Registration procedures referred to in this section are defined in [[RFC8126](#)].

### 7.1. BPSec Security Contexts

Within the "Bundle Protocol" registry [[IANA-BUNDLE](#)], the following entry has been added to the "BPSec Security Context Identifiers" sub-registry.

Value	Description	Reference
TBD-COSE	COSE	This specification.

Table 6

## 8. Acknowledgments

The interoperability minimum algorithms and parameters are based on the draft [[I-D.ietf-dtn-bpsec-interop-sc](#)].

## 9. References

### 9.1. Normative References

[[IANA-BUNDLE](#)] IANA, "Bundle Protocol", <<https://www.iana.org/assignments/bundle/>>.

[[IANA-CORE](#)] IANA, "Constrained RESTful Environments (CoRE) Parameters", <<https://www.iana.org/assignments/core-parameters/>>.

[[IANA-COSE](#)] IANA, "CBOR Object Signing and Encryption (COSE)", <<https://www.iana.org/assignments/cose/>>.

[[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>>.

- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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>>.
- [RFC8230] Jones, M., "Using RSA Algorithms with CBOR Object Signing and Encryption (COSE) Messages", RFC 8230, DOI 10.17487/RFC8230, September 2017, <<https://www.rfc-editor.org/info/rfc8230>>.
- [RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.
- [I-D.ietf-dtn-bpsec] Birrane, E. and K. McKeever, "Bundle Protocol Security Specification", Work in Progress, Internet-Draft, draft-ietf-dtn-bpsec-22, 10 March 2020, <<https://tools.ietf.org/html/draft-ietf-dtn-bpsec-22>>.

## 9.2. Informative References

- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <<https://www.rfc-editor.org/info/rfc7049>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP

205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

**[I-D.ietf-dtn-bpbis]**

Burleigh, S., Fall, K., and E. Birrane, "Bundle Protocol Version 7", Work in Progress, Internet-Draft, draft-ietf-dtn-bpbis-26, 28 July 2020, <<https://tools.ietf.org/html/draft-ietf-dtn-bpbis-26>>.

**[I-D.ietf-dtn-bpsec-interop-sc]**

Birrane, E., "BPsec Interoperability Security Contexts", Work in Progress, Internet-Draft, draft-ietf-dtn-bpsec-interop-sc-01, 4 February 2020, <<https://tools.ietf.org/html/draft-ietf-dtn-bpsec-interop-sc-01>>.

**[github-dtn-bpsec-cose]** Sipos, B., "DTN Bundle Protocol Security COSE Security Contexts", <<https://github.com/BSipos-RKF/dtn-bpsec-cose/>>.

## Appendix A. Examples

### A.1. Symmetric Key COSE\_Mac0

This is an example of a MAC with implied recipient (and its key material). The provided figures are extended diagnostic notation [RFC8610].

The 256-bit key used is shown below.

```
[
{
  / kty / 1: 4, / symmetric /
  / kid / 2: 'ExampleMAC',
  / k / -1: h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e3999db
          ae4ce45c'
}
]
```

Figure 1: Symmetric Key

```
[
  7, / BP version /
  0, / flags /
  0, / CRC type /
  [1, "///dst/svc"], / destination /
  [1, "///src/bp"], / source /
  [1, "///src/bp"], / report-to /
  [0, 40], / timestamp /
  1000000 / lifetime /
]
```

Figure 2: Primary block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  <<300>> / type-specific-data: age /
]
```

Figure 3: Target block CBOR diagnostic

The external\_aad is the encoded primary block. The payload is the encoded target block.

```
[
  "MAC0", / context /
  h'a10105', / protected /
  h'880700008201692f2f6473742f7376638201682f2f7372632f62708201682f2f7372
    632f6270820018281a000f4240', / external_aad /
  h'85070200004319012c' / payload /
]
```

Figure 4: MAC\_structure CBOR diagnostic

```

[
  [2], / targets /
  0, / security context TBD /
  0, / flags /
  [
    [ / target block #2 /
      [ / result /
        17, / COSE_Mac0 tag /
        [
          <<{ / protected /
            / alg / 1:5 / HMAC 256//256 /
          }>>,
          { / unprotected /
            / kid / 4:'ExampleMAC'
          },
          null, / payload /
          h'1349a33b41b020e46669b714b53a1b79db458fdef0f0b7a0daebde6baf27
            7472' / tag /
        ]
      ]
    ]
  ]
]

```

Figure 5: Abstract Security Block CBOR diagnostic

## A.2. RSA Keypair COSE\_Sign1

This is an example of a signature with implied recipient (and its key material). The provided figures are extended diagnostic notation [[RFC8610](#)].

The 512-bit private key used is below. It is not supposed to be a secure configuration, only intended to explain the procedure. This signature uses zero-length salt for deterministic output, which differs from the parameter specified by [[RFC8230](#)] and is not recommended for normal use.

```
[
  { / signing private key /
    / kty / 1: 3, / RSA /
    / kid / 2: 'ExampleRSA',
    / n / -1: b64'3bUZ1LR9oBiBpx6lGZuvtMBPTAS5qG0sF8A7Q0DUz13fs71PH0e9nD
    / e / -2: b64'AQAB',
    / d / -3: b64'yCQmj2foSFAXKuB1Nmre8RLyArP5Td08lSxJ0UWl1ixmFRoso_2jHI
    / p / -4: b64'7snebs70tMJ67A1qA4Yk5ujvjyaDEIsfch_fRwVIVik',
    / q / -5: b64'7bAM_t782esDusNKAzr5EQaa3wjTQ2CUXBKEFSLgc1E',
    / dP / -6: b64'Iiay7kwhCV0rMWl1uQ1NZ8z2vhV29z2-gJb4WvLxdok',
    / dQ / -7: b64'bC7WK2dJBNKv9uC0HlxIIItSzxtIYfjFGNYD8i7Wo5E',
    / qInv / -8: b64'6efvn6d0ADFQJxNLqjRJyE5E1m_dYQEvCI2mAqixshA'
  }
]
```

Figure 6: Private Keys

```
[
  7, / BP version /
  0, / flags /
  0, / CRC type /
  [1, "///dst/svc"], / destination /
  [1, "///src/bp"], / source /
  [1, "///src/bp"], / report-to /
  [0, 40], / timestamp /
  1000000 / lifetime /
]
```

Figure 7: Primary block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  <<300>> / type-specific-data: age /
]
```

Figure 8: Target block CBOR diagnostic

The external\_aad is the encoded primary block. The payload is the encoded target block.

```
[
  "Signature1", / context /
  h'a1013824', / protected /
  h'880700008201692f2f6473742f7376638201682f2f7372632f62708201682f2f7372
    632f6270820018281a000f4240', / external_aad /
  h'85070200004319012c' / payload /
]
```

Figure 9: Sig\_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  0, / flags /
  [
    [ / target block #2 /
      [ / result /
        18, / COSE_Sign1 tag /
        [
          <<{ / protected /
            / alg / 1:-37 / PS256 /
          }>>,
          { / unprotected /
            / kid / 4:'ExampleRSA'
          },
          null, / payload /
          h'53d983df0590f529456b661d36f217d722aa88497f04779385a9a786693d
            518778a23b912e02e272ea120adf0c1ddf2e08fb5efc54c1f6d36a95054b
            745fa47e' / signature /
        ]
      ]
    ]
  ]
]
```

Figure 10: Abstract Security Block CBOR diagnostic

### A.3. Symmetric Key COSE\_Encrypt0

This is an example of an encryption with implied recipient (and its direct content encryption key). The provided figures are extended diagnostic notation [[RFC8610](#)].

This example uses a single shared content encryption key, which is not recommended for normal use. The 256-bit key used is shown below. A random IV is generated for this operation and is indicated in a standard way in the unprotected header.

```
[
  {
    / kty / 1: 4, / symmetric /
    / kid / 2: 'ExampleCEK',
    / k / -1: h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e3999db
      ae4ce45c'
  }
]
```

Figure 11: Symmetric Keys

```
[
  7, / BP version /
  0, / flags /
  0, / CRC type /
  [1, "///dst/svc"], / destination /
  [1, "///src/bp"], / source /
  [1, "///src/bp"], / report-to /
  [0, 40], / timestamp /
  10000000 / lifetime /
]
```

Figure 12: Primary block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  <<300>> / type-specific-data: age /
]
```

Figure 13: Initial Target block CBOR diagnostic

The `external_aad` is a concatenation of the encoded primary block and the encoded augmented target block (its block data removed).

```
[
  "Encrypt0", / context /
  h'a10103', / protected /
  h'880700008201692f2f6473742f7376638201682f2f7372632f62708201682f2f7372
    632f6270820018281a000f4240850702000040' / external_aad /
]
```

Figure 14: Enc\_structure CBOR diagnostic



```
[
  [2], / targets /
  0, / security context TBD /
  0, / flags /
  [
    [ / target block #2 /
      [ / result /
        16, / COSE_Encrypt0 tag /
        [
          <<{ / protected /
            / alg / 1:3 / A256GCM /
          }>>,
          { / unprotected /
            / kid / 4:'ExampleCEK',
            / iv / 5: h'6f3093eba5d85143c3dc484a'
          },
          null / payload /
        ]
      ]
    ]
  ]
]
```

Figure 15: Abstract Security Block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  h'63bb1617fc5076cec266907a7143d28587f04e' / ciphertext /
]
```

Figure 16: Encrypted Target block CBOR diagnostic

#### A.4. Symmetric KEK COSE\_Encrypt

This is an example of an encryption with a random CEK and an explicit key-encryption key (KEK) identified by a Key ID. The provided figures are extended diagnostic notation [[RFC8610](#)].

The keys used are shown in [Figure 17](#). A random IV is generated for this operation and is indicated in a standard way in the unprotected header of [Figure 21](#).

```
[
  {
    / kty / 1: 4, / symmetric /
    / kid / 2: 'ExampleKEK',
    / k / -1: h'0e8a982b921d1086241798032fedc1f883eab72e4e43bb2d11cfae38
      ad7a972e'
  },
  {
    / kty / 1: 4, / symmetric /
    / kid / 2: 'ExampleCEK',
    / k / -1: h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e3999db
      ae4ce45c'
  }
]
```

Figure 17: Symmetric Keys

```
[
  7, / BP version /
  0, / flags /
  0, / CRC type /
  [1, "///dst/svc"], / destination /
  [1, "///src/bp"], / source /
  [1, "///src/bp"], / report-to /
  [0, 40], / timestamp /
  1000000 / lifetime /
]
```

Figure 18: Primary block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  <<300>> / type-specific-data: age /
]
```

Figure 19: Initial Target block CBOR diagnostic

The `external_aad` is a concatenation of the encoded primary block and the encoded augmented target block (its block data removed).

The CEK and content plaintext are the same here as in [Figure 14](#) but the context text is different.

```
[
  "Encrypt", / context /
  h'a10103', / protected /
  h'880700008201692f2f6473742f7376638201682f2f7372632f62708201682f2f7372
    632f6270820018281a000f4240850702000040' / external_aad /
]
```

Figure 20: Enc\_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  0, / flags /
  [
    [ / target block #2 /
      [ / result /
        96, / COSE_Encrypt tag /
        [
          <<{ / protected /
            / alg / 1:3 / A256GCM /
          }>>,
          { / unprotected /
            / iv / 5: h'6f3093eba5d85143c3dc484a'
          },
          null, / payload /
          [
            [ / recipient /
              h'', / protected /
              { / unprotected /
                / alg / 1:-5, / A256KW /
                / kid / 4:'ExampleKEK'
              },
              h'917f2045e1169502756252bf119a94cdac6a9d8944245b5a9a26d403
                a6331159e3d691a708e9984d', / key-wrapped /
            [] / no more layers /
          ]
        ]
      ]
    ]
  ]
]
```

Figure 21: Abstract Security Block CBOR diagnostic

Although the same CEK is used in this example as the Encrypt0 example, the block ciphertext is different than [Figure 16](#) because the Enc\_structure (used as AAD) is different.

```
[  
  7, / type code - bundle age /  
  2, / block num /  
  0, / flags /  
  0, / CRC type /  
  h'63bb160aa1804f936570b982bf7c396694e574' / ciphertext /  
]
```

Figure 22: Encrypted Target block CBOR diagnostic

#### **Author's Address**

Brian Sipos  
RKF Engineering Solutions, LLC  
7500 Old Georgetown Road  
Suite 1275  
Bethesda, MD 20814-6198  
United States of America

Email: [BSipos@rkf-eng.com](mailto:BSipos@rkf-eng.com)