DTN Bundle Protocol Security COSE Security Contexts
draft-bsipos-dtn-bpsec-cose-01

Abstract

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

Table of Contents

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.

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 Contexts

Rather than defining a single security context for both integrity and confidentiality blocks, this document specifies two separate security contexts which are analogous to the two BPSec block types. Each security context allows a specific set of BPSec Result IDs.

The existing COSE message-marking tags in Section 2 of [RFC8152] SHALL be used as BPSec Result ID values for all COSE security contexts (see Table 1 and Table 2). 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.

3.1. COSE Integrity Context

The COSE Integrity Context has a Security Context ID of TBD-CI.

The integrity context SHALL allow only the Result IDs from Table 1. Each integrity context result value SHALL consist of the COSE message indicated by Table 1 in its decoded form.

Result ID	Result Message
97	COSE_Mac
17	COSE_Mac0
98	COSE_Sign
18	COSE_Sign1

Table 1: 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 Context

The COSE Confidentiality Context has a Security Context ID of TBD-CC.

The confidentiality context SHALL allow only the Result IDs from
Table 2. Each confidentiality context result value SHALL consist of
the COSE message indicated by Table 2 in its decoded form.

Result ID	Result Message
96	COSE_Encrypt
16	COSE_Encrypt0

Table 2: 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.

When used in a BPSec result, each COSE message SHALL contain an
explicit algorithm identifier in the lower (content) layers. A BPSec
security operation always occurs within the context of the immutable
primary block and its parameters. When available and not implied by
the bundle source, a COSE message SHOULD contain a key identifier in
the highest 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.

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.

[4.1](#). 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 3. Implementations capable of
doing so SHOULD support the asymmetric keyed and key-encryption
algorithms of Table 3.

BPSec Block	COSE Layer	Name	Code	Implementation Requirements
Integrity	1	HMAC 256/256	5	Required
Integrity	1	ES256	-7	Recommended
Integrity	1	PS256	-37	Recommended
Confidentiality	1	A256GCM	3	Required
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 3: 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 4.1 recommends key identifiers when possible.
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-CI | COSE Integrity      | This specification.|
|        |                     |                    |
| TBD-CC | COSE Confidentiality| This specification.|
+--------+---------------------+--------------------+
```

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

[I-D.ietf-dtn-bpsec]
          Birrane, E. and K. McKeever, "Bundle Protocol Security
          Specification", draft-ietf-dtn-bpsec-22 (work in
          progress), March 2020.

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

[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>.

## 9.2.  Informative References

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

   [I-D.ietf-dtn-bpbis]
              Burleigh, S., Fall, K., and E. Birrane, "Bundle Protocol
              Version 7", draft-ietf-dtn-bpbis-25 (work in progress),
              May 2020.

   [I-D.ietf-dtn-bpsec-interop-sc]
              Birrane, E., "BPSec Interoperability Security Contexts",
              draft-ietf-dtn-bpsec-interop-sc-01 (work in progress),
              February 2020.

   [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>.

## Appendix A.  Examples

## A.1.  Symmetric Key COSE_Mac0

   This is an example of a MAC with implied recipient (and its key
   material).  The two provided figures are CBOR diagnostic notation
   [RFC7049] of the target block being signed and the Abstract Security
   Block (which will itself be enveloped within a BIB).

   The 256-bit key used is shown below.

```
  / signing key /
  h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e3999dbae4ce45c'
```

                              Symmetric Key

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

                  Figure 1: Primary block CBOR diagnostic

```
  [
    7, / type code - bundle age /
    2, / block num /
    0, / flags /
    0, / CRC type /
    h'19012c' / type-specific-data:
      300 \ age \
    /
  ]
```

                   Figure 2: 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'880700008201462f2f6473742f8201462f2f7372632f8201462f2f7372632f820018
    281a000f4240', / external_aad /
  h'85070200004319012c' / payload /
]
```

                   Figure 3: MAC_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  0, / flags /
  [
    [ / target block #2 /
      [ / result /
        17, / COSE_Mac0 tag /
        [
          h'a10105' / protected {
             \ alg \ 1:5 \ HMAC 256//256 \
           } / ,
          { / unprotected /
            / kid / 4:'mykey'
          },
          null, / payload /
          h'd98308918d36dc4190a93f84c8d857015b75b78edea3360282555257c3be
            f847' / tag /
        ]
      ]
    ]
  ]
]
```

                Figure 4: 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 two provided figures are CBOR diagnostic notation
   [[RFC7049]()] of the target block being signed and the Abstract Security
   Block (which will itself be enveloped within a BIB).

   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.

   -----BEGIN RSA PRIVATE KEY-----
   MIIBOwIBAAJBAN21GdS0faAYgacepRmbr7TAT0wEuahjrBfAO0Dg1M5d37O9Tx9H
   vZw2OEcLq2WTvf0Kja1JWpqdoJm17LghhPkCAwEAAQJBAMgkJo9n6EhQFyrgdTZq
   3vES8gKz+U3TvJUsSdFFpZYsZhUaLKP9oxyIxl3IvK5iS0oAsW0nqI7aMcBoPmxZ
   pQECIQDuyd5uzvS0wnrsDWoDhiTm6O+PJoMQix9yH99HBUhWKQIhAO2wDP7e/Nnr
   A7rDSgM6+REGmt8I00NglFwShBUi4HJRAiAiJrLuTCEJXSsxaXW5DU1nzPa+FXb3
   Pb6Alvha8vF2iQIgbC7WK2dJBNKv9uCOHlxIItSzxtIYfjFGNYYD8i7Wo5ECIQDp
   5++fp04AMVAnE0uqNEnITkTWb91hAS8IjaYCqLGyEA==
   -----END RSA PRIVATE KEY-----
```

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

                  Figure 5: Primary block CBOR diagnostic

```
  [
    7, / type code - bundle age /
    2, / block num /
    0, / flags /
    0, / CRC type /
    h'19012c' / type-specific-data:
      300 \ age \
    /
  ]
```

                  Figure 6: 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'880700008201462f2f6473742f8201462f2f7372632f8201462f2f7372632f820018
    281a000f4240', / external_aad /
  h'85070200004319012c' / payload /
]
```

                  Figure 7: Sig_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  0, / flags /
  [
    [ / target block #2 /
      [ / result /
        18, / COSE_Sign1 tag /
        [
          h'a1013824' / protected {
             \ alg \ 1:-37 \ PS256 \
           } / ,
          { / unprotected /
            / kid / 4:'mykey'
          },
          null, / payload /
          h'1a35746072396c74275fd7b443a0d7391a0f92012a53e0accc543daa51ae
            6faae551a4843a0bc7c3bf808e3638ddc381355b54cc60f4ca9dea15923b
            5986e758' / signature /
        ]
      ]
    ]
  ]
]
```

                   Figure 8: 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 CBOR
   diagnostic notation [RFC7049] of the target block being encrypted,
   the Abstract Security Block (which will itself be enveloped within a
   BCB), and the resulting target block.

   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.

   / content encryption key /
   h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e3999dbae4ce45c'

                             Symmetric Keys

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

                    Figure 9: Primary block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  h'19012c' / type-specific-data:
    300 \ age \
  /
]
```

                  Figure 10: 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'880700008201662f2f6473742f8201662f2f7372632f8201662f2f7372632f820018
    281a000f4240850702000040' / external_aad /
]
```

                     Figure 11: Enc_structure CBOR diagnostic

```
    [
      [2], / targets /
      0, / security context TBD /
      0, / flags /
      [
        [ / target block #2 /
          [ / result /
            16, / COSE_Encrypt0 tag /
            [
              h'a10103', / protected {
                 \ alg \ 1:3 \ A256GCM \
               } /
              { / unprotected /
                / kid / 4:'mykey',
                / iv / 5: h'6f3093eba5d85143c3dc484a'
              },
              null / payload /
            ]
          ]
        ]
      ]
    ]
```

                Figure 12: Abstract Security Block CBOR diagnostic

```
    [
      7, / type code - bundle age /
      2, / block num /
      0, / flags /
      0, / CRC type /
      h'63bb16685ef432a0e6f1d404da71959081a715' / ciphertext /
    ]
```

                Figure 13: 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