Workgroup: Delay-Tolerant Networking
Internet-Draft: draft-bsipos-dtn-bpsec-cose-03
Published: 14 November 2020
Intended Status: Standards Track
Expires: 18 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

Copyright Notice

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. The focus of the recommended algorithms is to allow BPSec to be used in a Public Key Infrastructure (PKI) as described in Section 1.2.

Examples of specific uses are provided in Appendix A to aid in implementation support of the interoperability algorithms.

1.1. Scope

This document describes a profile of COSE which is tailored for use in BPSec and a method of including full COSE messages within BPSec security blocks. This document does not address:

*Policies or mechanisms for issuing Public Key Infrastructure Using X.509 (PKIX) certificates; provisioning, deploying, or accessing certificates and private keys; deploying or accessing certificate revocation lists (CRLs); or configuring security parameters on an individual entity or across a network.

*Uses of COSE beyond the profile defined in this document.

*How those COSE algorithms are intended to be used within a larger security context. Many header parameters used by COSE (e.g., key identifiers) depend on the network environment and security policy related to that environment.

1.2. PKIX Environments and CA Policy

This specification gives requirements about how to use PKIX certificates issued by a Certificate Authority (CA), but does not define any mechanisms for how those certificates come to be.

To support the PKIX uses defined in this document, the CA(s) issuing certificates for BP nodes are aware of the end use of the certificate, have a mechanism for verifying ownership of a Node ID, and are issuing certificates directly for that Node ID. BPSec security acceptors authenticate the Node ID of security sources when verifying integrity using a public key provided by a PKIX certificate (see Section 4.3.1).

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 parameters and COSE messages which can be present.

The COSE security context SHALL have the Security Context ID specified in Section 7.1.

The COSE security context has parameters to carry public key-related information, with code points are defined in Section 3.1, and results to carry COSE messages, with code points defined in Section 3.2 and Section 3.3. 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 structure SHALL be directly included within the abstract security block (ASB) CBOR structure. There is no use of embedded encoded CBOR (e.g. CBOR encoded as a byte string) in this specification.

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 Security Parameters

Each COSE context parameter value SHALL consist of the COSE structure indicated by Table 1 in its decoded (CBOR item) form. Each security block MAY contain any number of each parameter type. See Section 4.3 for a definition of how the aggregate of all security parameters apply to each security result.

Implementations capable of handling asymmetric-keyed algorithms SHOULD support the public key handling parameters of Table 1. COSE security parameters SHALL NOT contain any private key material. The security parameters are all stored in the bundle as plaintext and are visible to any bundle handlers.

Parameter ID	Parameter Structure	Reference
1	COSE_Key	[RFC8152]
2	COSE_KeySet	[RFC8152]
3	COSE_X509 as x5chain	[I-D.ietf-cose-x509]
4	COSE_X509 as x5bag	[I-D.ietf-cose-x509]

Table 1: COSE Security Parameters

3.2. COSE Integrity

When used within a Block Integrity Block, COSE context SHALL allow all Parameter IDs defined in Table 1. 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.

3.3. COSE Confidentiality

When used within a Block Confidentiality Block, COSE context SHALL allow all Parameter IDs defined in Table 1. When used within a Block Confidentiality Block, COSE context SHALL allow only the Result IDs from Table 3. Each confidentiality result value SHALL consist of the COSE message indicated by Table 3 in its decoded form.

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

Table 3: 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 Messages

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 SHALL contain a key identifier in the highest (recipient) layer. See Section 4.3 for specifics about asymmetric key identifiers. When a key identifier is not available, BPSec acceptors SHALL use the Security Source (if available) and the Bundle Source to imply which keys can be used for security operations. Using implied keys has an interoperability risk, see Section 6.3 for details. 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.

4.2. 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 and key-encryption algorithms of Table 4. Implementations capable of doing so SHOULD support the asymmetric keyed and key-encryption algorithms of Table 4.

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	Required
Integrity or Confidentiality	2	ECDH-ES + A256KW	-31	Recommended
Integrity or Confidentiality	2	RSAES-OAEP w/ SHA-256	-41	Recommended

Table 4: 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 SHALL 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 SHALL include a public key identifier (see Section 4.3).

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 SHALL include a public key identifier (see Section 4.3). 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.

4.3. Asymmetric Key Types and Identifiers

This section applies when a BIB uses a public key for verification, or when a BCB uses a public key for encryption. When using asymmetric keyed algorithms, the security source SHALL include a public key identifier as a recipient header. The public key identifier SHALL be either a "kid" [RFC8152], an "x5t" [I-D.ietf-cose-x509], or an equivalent identifier.

When a BIB result contains a "kid" identifier, the security source SHOULD include an appropriate COSE public key in the security parameters. When BIB result contains a "x5t" identifier, the security source SHOULD include an appropriate PKIX certificate chain in the security parameters. For a BIB, if all potential security acceptors are known to possess related public key and/or certificate data then the public key parameters can be omitted. Risks of not including related data are described in Section 6.3 and Section 6.4.

When present, public keys and certificates SHOULD be included as ASB parameters rather than within ASB results. This provides size efficiency when multiple security results are present because they will all be from the same security source and likely share the same public key material. Security acceptors SHALL still process public keys or certificates present in a result as applying to that individual result.

Security acceptors SHALL aggregate all public keys from all parameters within a single BIB or BCB, independent of encoded type or order of parameters. Because each context contains a single set of security parameters which apply to all results in the same context, security acceptors SHALL treat all public keys as being related to the security source itself and potentially applying to every result.

4.3.1. PKIX Certificates

When PKIX certificates are present as parameters, security sources SHOULD include the entire certification chain to the root CA. When PKIX certificates are used by security acceptors and the end-entity certificate is not explicitly trusted (i.e. pinned), the security acceptor SHALL perform the certification path validation of [RFC5280] up to one or more trusted CA certificates. Leaving out part of the certification chain can cause the security acceptor to fail to validate a BIB if the left-out certificates are unknown to the acceptor (see Section 6.4).

When a PKIX certificate is referenced by a BIB result, security acceptors SHALL authenticate either the Security Source (if present) or the Bundle Source (as the implied security source) against any NODE-ID contained in the referenced certificate as defined in [I-D.ietf-dtn-tcpclv4]. If the Security Source authentication result is Failure or if the result is Absent and security policy requires an authenticated Node ID, the acceptor SHALL treat the security result as invalid.

All certificates used by COSE security SHALL include a key usage extension in accordance with [RFC5280]. The key usage extension is required to be supported by CAs conforming to the profile of [RFC5280]. A security acceptor SHALL limit the use of PKIX certificates based on the key usage extension.

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 additional authenticated data (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: BP Node Impersonation

When certificates are referenced by BIB results it is possible that the certificate does not contain a NODE-ID or does contain one but has a mismatch with the actual security source (see Section 1.2). Having a CA-validated certificate does not alone guarantee the identity of the security source from which the certificate is provided; additional validation procedures in Section 4.3.1 bind the Node ID based on the contents of the certificate.

6.3. Threat: Unidentifiable Key

The profile in Section 4.2 recommends key identifiers when possible and the parameters in section Section 3.1 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.

Because the key identifier headers are unprotected (see Section 4.3), there is still the possibility that an active attacker removes or alters key identifier(s) in the result. This can cause the security acceptor to not be able to properly verify a valid signature or not use the correct private key to decrypt valid ciphertext.

6.4. Threat: Non-Trusted Public Key

The profile in Section 4.2 allows the use of PKIX which typically involves end-entity certificates chained up to a trusted root CA. This allows a BIB to contain end-entity certificates not previously known to a security acceptor but still trust the certificate by verifying it up to a trusted CA. In an environment where security acceptors are known to already contain needed root and intermediate CAs there is no need to include those CAs in a proper chain within the security parameters, but this has a risk of an acceptor not actually having one of the needed CAs.

Because the security parameters are not included as AAD, there is still the possibility that an active attacker removes or alters certification chain data in the parameters. This can cause the security acceptor to be able to verify a valid signature but not trust the public key used to perform the verification.

6.5. Threat: Passive Leak of Key Material

It is important that the key requirements of Section 3.1 apply only to public keys and PKIX certificates. Including non-public key material in ASB parameters will expose that material in the bundle data and over the bundle convergence layer during transport.

6.6. 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 5

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-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>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[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>.

[RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <https://www.rfc-editor.org/info/rfc8551>.

[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>.

[I-D.ietf-dtn-tcpclv4]
Sipos, B., Demmer, M., Ott, J., and S. Perreault, "Delay-Tolerant Networking TCP Convergence Layer Protocol Version 4", Work in Progress, Internet-Draft, draft-ietf-dtn-tcpclv4-22, 26 October 2020, <https://tools.ietf.org/html/draft-ietf-dtn-tcpclv4-22>.

[I-D.ietf-cose-x509]
Schaad, J., "CBOR Object Signing and Encryption (COSE): Header parameters for carrying and referencing X.509 certificates", Work in Progress, Internet-Draft, draft-ietf-cose-x509-07, 17 September 2020, <https://tools.ietf.org/html/draft-ietf-cose-x509-07>.

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>.

[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

These examples are intended to have the correct structure of COSE security blocks but in some cases use simplified algorithm parameters or smaller key sizes than are required by the actual COSE profile defined in this documents. Each example indicates how it differs from the actual profile if there is a meaningful difference.

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 an explicit signer key ID and signer public key itself (as a COSE_Key). The provided figures are extended diagnostic notation [RFC8610].

The only differences between this example and a use of a PKIX public key certificate are: the parameters would have an x5chain parameter instead of a COSE_Key type, and the signature recipient would reference an "x5t" value instead of a "kid" value. Neither of these is a change to a protected header so, given the same private key, there would be no change to the signature itself.

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: b64'3bUZ1LR9oBiBpx6lGZuvtMBPTAS5qGOsF8A7QODUzl3fs71PH0e9nD
                  Y4RwurZZO9_QqNrUlamp2gmbXsuCGE-Q',
    / e / -2: b64'AQAB',
    / d / -3: b64'yCQmj2foSFAXKuB1Nmre8RLyArP5TdO8lSxJ0UWllixmFRoso_2jHI
                  jGXci8rmJLSgCxbSeojtoxwGg-bFmlAQ',
    / p / -4: b64'7snebs70tMJ67A1qA4Yk5ujvjyaDEIsfch_fRwVIVik',
    / q / -5: b64'7bAM_t782esDusNKAzr5EQaa3wjTQ2CUXBKEFSLgclE',
    / dP / -6: b64'Iiay7kwhCV0rMWl1uQ1NZ8z2vhV29z2-gJb4WvLxdok',
    / dQ / -7: b64'bC7WK2dJBNKv9uCOHlxIItSzxtIYfjFGNYYD8i7Wo5E',
    / qInv / -8: b64'6efvn6dOADFQJxNLqjRJyE5E1m_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 /
  1, / flags /
  [ / parameters /
    [
      101, / COSE key /
      { / public key /
        / kty / 1: 3, / RSA /
        / kid / 2: 'ExampleRSA',
        / n / -1: b64'3bUZ1LR9oBiBpx6lGZuvtMBPTAS5qGOsF8A7QODUzl3fs71PH0
                      e9nDY4RwurZZO9_QqNrUlamp2gmbXsuCGE-Q',
        / e / -2: b64'AQAB',
      }
    ]
  ],
  [
    [ / 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 /
  1000000 / 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