Workgroup: Delay-Tolerant Networking
Internet-Draft: draft-bsipos-dtn-bpsec-cose-04
Published: 22 December 2020
Intended Status: Standards Track
Expires: 25 June 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 (see Section 3.6.1) using a public key provided by a PKIX certificate (see Section 4.3.1) following the certificate profile of Section 5.

1.3. Use of CDDL

This document defines CBOR structure using the Concise Data Definition Language (CDDL) of [RFC8610]. The entire CDDL structure can be extracted from the XML version of this document using the XPath expression:

```
'//sourcecode[@type="cddl"]'
```

The following initial fragment defines the top-level symbols of this document's CDDL, including the ASB data structure with its parameter/result sockets.

```
start = AAD-value / ext-data-asb
```

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 8.1.

Both types of security block can use the same parameters, defined in Section 3.2, to carry public key-related information and each type of security block allows specific COSE message results, defined in Section 3.3.

3.1. Security Scope

The scope here refers to the set of information used by the security context to cryptographically bind with the plaintext data being integrity-protected or confididentiality-protected. This information is generically referred to as additional authenticated data (AAD), which is also the term used by COSE to describe the same data.

The sources for AAD within the COSE context are described below, controlled by the AAD Scope Flags parameter of Section 3.2.2, and implemented as defined in Section 3.5.1.

Bundle Primary Block The primary block identifies a bundle and, once created, the contents of this block are immutable. Changes to the primary block associated with the security target indicate that the target is no longer in its original bundle. Including this data as part of AAD ensures that security target appears in the same bundle that the security source intended.

Target Block Metadata When the target block is a canonical block (i.e., not the primary block) it contains its block-type-specific data, which is the subject of the security operation, but also metadata identifying the block. This metadata explicitly excludes the CRC type and value fields because the CRC is derived from the block-type-specific data. Including this data as part of AAD ensures that the target data appears in the same block that the security source intended.

Security Block Metadata The BPSec block containing the security result for which the AAD is assembled also has metadata identifying the block. Including this data as part of AAD ensures that the security result appears in the same block that the security source intended.

3.2. 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. When a parameter is not present, the security acceptor SHALL use the Default Value of Table 1.

Parameter ID	Parameter Structure	Reference	Default Value
1	COSE_Key	[RFC8152]	none
2	COSE_KeySet	[RFC8152]	none

Parameter ID	Parameter Structure	Reference	Default Value
3	COSE_X509 as x5chain	[I-D.ietf-cose-x509]	none
4	COSE_X509 as x5bag	[I-D.ietf-cose-x509]	none
5	AAD_Scope	Section 3.2.2 of this RFC	0x7 (all contexts)

Table 1: COSE Security Parameters

```
$bpsec-param-pair /= [1, COSE_Key]
$bpsec-param-pair /= [2, COSE_KeySet]
$bpsec-param-pair /= [3, x5chain: COSE_X509]
$bpsec-param-pair /= [4, x5bag: COSE_X509]
$bpsec-param-pair /= [5, AAD-scope]
```

Figure 1: COSE context parameters CDDL

3.2.1. Key Containers

Implementations capable of handling asymmetric-keyed algorithms SHOULD support the public key handling parameters of Table 1. See Section 4.3 for a definition of how the aggregate of all public keys and certificates in security parameters apply to each security result.

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.

3.2.2. AAD Scope

The AAD Scope parameter controls what data is included in the AAD for both integrity and confidentiality operations. The AAD Scope parameter SHALL be encoded as a unit value with bit flags defined in Table 2. The default value for this parameter has all flags set which has the AAD include all available context.

A CDDL representation of this definition is included in Figure 2 for reference.

Name	Code	Description
has-primary-ctx	0x01	If bit is set, indicates that the primary block is included in AAD scope.
has-target-ctx	0x02	If bit is set, indicates that the target block metadata is included in AAD scope.
has-security-ctx	0x04	If bit is set, indicates that the security block metadata is included in AAD scope.

Name	Code	Description
Reserved	others	

Table 2: AAD Scope Flags

```
AAD-scope = uint .bits AAD-scope-flags
AAD-scope-flags = &(
    has-primary-ctx: 0,
    has-target-ctx: 1,
    has-security-ctx: 2,
)
```

Figure 2: AAD Scope CDDL

3.3. Results

Although each COSE context result is a COSE message, the types of message allowed depend upon the security block type in which the result is present: only MAC or signature messages are allowed in a BIB and only encryption messages are allowed in a BCB.

The code points for Result ID values are 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).

These generic requirements are formalized in the CDDL fragment of Figure 3.

```
$bpsec-result-pair /= [16, COSE_Encrypt0]
$bpsec-result-pair /= [17, COSE_Mac0]
$bpsec-result-pair /= [18, COSE_Sign1]
$bpsec-result-pair /= [96, COSE_Encrypt]
$bpsec-result-pair /= [97, COSE_Mac]
$bpsec-result-pair /= [98, COSE_Sign]
```

Figure 3: COSE context results CDDL

3.3.1. Integrity Messages

When used within a Block Integrity Block, the COSE context SHALL allow only the Result IDs from Table 3. Each integrity result value

SHALL consist of the COSE message indicated by Table 3 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 3: 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].

The COSE "protected attributes from the application" used for a signature or MAC result SHALL be the encoded data defined in Section 3.5.1. The COSE payload used for a signature or MAC result SHALL be either the block-type-specific data of the target, if the target is not the primary block, or an empty byte string if the target is the primary block.

3.3.2. Confidentiality Messages

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. Security acceptors MUST NOT assume that the size of the plaintext is the same as the size of the ciphertext.

Each confidentiality result SHALL use the "detached" payload form with nil payload value. The confidentiality result for COSE_Encrypt and COSE_Encrypt0 messages are computed by the procedure in Section 5.3 of [RFC8152].

The COSE "protected attributes from the application" used for an encryption result SHALL be the encoded data defined in Section 3.5.1. The COSE payload used for an encryption result SHALL be the block-type-specific data of the target. Because confidentiality of the primary block is disallowed by BPSec, there is no logic here for handling a BCB with a target on the primary block.

3.4. Key Considerations

This specification does not impose any additional key requirements beyond those already specified for each COSE algorithim required in Section 4.

3.5. Canonicalization Algorithms

Generating or processing COSE messages for the COSE context follows the profile defined in Section 4 with the "protected attributes from the application" (i.e., the "external_aad" item) generated as defined in Section 3.5.1.

3.5.1. Generating AAD

The AAD used for both integrity and confidentiality messages SHALL be the determistically encoded form of a CBOR array containing the following:

1. The first item SHALL be either: the CBOR array (unencoded) form of the primary block of the bundle if the AAD Scope has the has-primary-ctx flag set, otherwise the null value.

2. The second item SHALL be either: a CBOR array containing the first three fields of the target block (i.e., the block type code, block number, and control flags) if the AAD Scope has the has-target-ctx flag set, otherwise the null value.

3. The third item SHALL be either: a CBOR array containing the first three fields of the security block containing the result (i.e., the block type code, block number, and control flags) if the AAD Scope has the has-security-ctx flag set, otherwise the null value.

A CDDL representation of this data is shown below in Figure 4.

```
AAD-value = bstr .cbor AAD-structure
AAD-structure = [
    primary-ctx:  null / primary-block,  ; if has-primary-ctx is set
    target-ctx:   null / block-metadata, ; if has-target-ctx is set
    security-ctx: null / block-metadata  ; if has-security-ctx is set
]
; The first three fields of BP "canonical-block-structure"
block-metadata = [
    block-type-code: uint,
    block-number: uint,
    block-control-flags,
]
```

Figure 4: COSE context AAD CDDL

3.5.2. Payload Data

When correlating between BPSec target block-type-specific-data and COSE plaintext or payload, any byte string SHALL be handled in its decoded (CBOR item) form. This means any CBOR header or tag in a source encoding are ignored for the purposes of security processing. This also means that if the source byte string was encoded in a non-conforming way, for example in indefinite-length form or with a non-minimum-size lengnth, the security processing always treats it in a determistically encoded CBOR form.

3.6. Processing

This section describes block-level requirements for handling COSE security data.

Security results generated for BIB or BCB results SHALL conform to the COSE profile of Section 4. Security acceptors SHOULD

3.6.1. Security Source Authentication

This section explains how the certificate profile of Section 5 is used by a security acceptor to both validate an end-entity certificate and to use that certificate to authenticate the security source for the COSE security context.

Because of the standard policy of using separate certificates for transport, signing, and encryption (see Section 5.1) a single Node ID is likely to be associated with mulitple certificates, and any or all of those certificates can be present as security parameters (see Section 3.2.1). When present, a security acceptor SHALL use an "x5t" identifier from a COSE recipient to identify an end-entity certificate to use for result processing. Security acceptors SHALL NOT assume that a validated certificate containing a NODE-ID matching a security source is enough to associate a certificate with a COSE message or recipient.

3.6.1.1. Certificate Path and Purpose Validation

For each end-entity certificate referenced by a COSE context result, the security acceptor SHALL perform the certification path validation of [RFC5280] up to one of the acceptor's trusted CA certificates. If enabled by local policy, the entity SHALL perform an OCSP check of each certificate providing OCSP authoritiy information in accordance with [RFC6960]. If certificate validation fails or if security policy disallows a certificate for any reason, the acceptor SHALL treat the associated security result as failed. Leaving out part of the certification chain can cause the entity to fail to validate a certificate if the left-out certificates are unknown to the entity (see Section 7.2).

For each end-entity certificate referenced by a COSE context result, the security acceptor SHALL apply security policy to the Key Usage extension (if present) and Extended Key Usage extension (if present) in accordance with Section 4.2.1.12 of [RFC5280] and the profile in Section 5.

3.6.1.2. Node ID Authentication

If required by security policy, for each end-entity certificate referenced by a COSE context result the security acceptor SHALL validate the certificate NODE-ID in accordance with Section 6 of [RFC6125] using the NODE-ID reference identifier from either the Security Source (if present) or the Bundle Source (as the implied security source). If the NODE-ID validation result is Failure or if the result is Absent and security policy requires an authenticated Node ID, the security acceptor SHALL treat the result as failed.

3.6.2. Policy Recommendations

A RECOMMENDED security policy is to enable the use of OCSP checking when internet connectivity is present. A RECOMMENDED security policy is that if an Extended Key Usage is present that it needs to contain "id-kp-bundleSecurity" to be usable as an end-entity certificate for with COSE security results. A RECOMMENDED security policy is to require a validated Node ID (of Section 3.6.1.2) and to ignore any other identifiers in the end-entity certificate.

This policy relies on and informs the certificate requirements in Section 4.3.1. This policy assumes that a DTN-aware CA (see Section 1.2) will only issue a certificate for a Node ID when it has verified that the private key holder actually controls the DTN node; this is needed to avoid the threat identified in Section 7.4. This policy requires that a certificate contain a NODE-ID and allows the certificate to also contain network-level identifiers. A tailored policy on a more controlled network could relax the requirement on Node ID validation and/or Extended Key Usage presence.

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 7.5 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 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
Confidentiality	2	A256KW	-5	Required
Confidentiality	2	ECDH-ES + A256KW	-31	Recommended
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 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. When a COSE_Sign is used with an 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. When a COSE_Sign is used with an 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 an RSA keypair, the signature uses a PSS salt in accordance with Section 2 of [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. When a COSE_Encrypt is used with an overall key-encryption key (KEK), the recipient layer SHALL include a key identifier for the KEK.

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. When a COSE_Encrypt is used with an EC keypair, the recipient layer SHALL include a public key identifier (see Section 4.3). When a COSE_Encrypt is used with an EC keypair, the security source SHALL generate an ephemeral EC keypair for each security operation. When processing a COSE_Encrypt with an EC keypair, the security acceptor SHALL process all KDF and HMAC context data from the recipient headers in accordance with Section 11.2 of [RFC8152] even though the source is not required to provide any of those parameters.

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. When a COSE_Encrypt is used with an RSA keypair, the recipient layer SHALL include a public key identifier (see Section 4.3).

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 7.5 and Section 7.6.

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 7.6).

The end entity certificate associated with a BPSec security source SHALL adhere to the profile of Section 5.

5. PKIX Certificate Profile

All end-entity certificates used for BPSec SHALL conform to [RFC5280], or any updates or successors to that profile.

This profile requires Version 3 certificates due to the extensions used by this profile. Security acceptors SHALL reject as invalid Version 1 and Version 2 end-entity certificates.

Security acceptors SHALL accept certificates that contain an empty Subject field or contain a Subject without a Common Name. Identity information in end-entity certificates is contained entirely in the subjectAltName extension as a NODE-ID, as defined in [I-D.ietf-dtn-tcpclv4].

All end-entity and CA certificates used for BPSec SHOULD contain both a Subject Key Identifier and an Authority Key Identifier extension in accordance with [RFC5280]. Security acceptors SHOULD NOT rely on either a Subject Key Identifier and an Authority Key Identifier being present in any received certificate. Including key identifiers simplifies the work of an entity needing to assemble a certification chain.

A BPSec end-entity certificate SHALL contain a NODE-ID which authenticates the Node ID of the security source. The identifier type NODE-ID is defined in [I-D.ietf-dtn-tcpclv4].

When allowed by CA policy, a BPSec end-entity certificate SHOULD contain a PKIX Extended Key Usage extension in accordance with Section 4.2.1.12 of [RFC5280]. When the PKIX Extended Key Usage extension is present, it SHALL contain a key purpose "id-kp-bundleSecurity" as defined in [I-D.ietf-dtn-tcpclv4]. The "id-kp-bundleSecurity" purpose MAY be combined with other purposes in the same certificate.

When allowed by CA policy, a BPSec end-entity certificate SHALL contain a PKIX Key Usage extension in accordance with Section 4.2.1.3 of [RFC5280]. The PKIX Key Usage bits which are consistent with COSE security are: digitalSignature, nonRepudiation, keyEncipherment, and keyAgreement. The specific algorithms used by COSE messages in security results determine which of those key uses are exercised. See Section 5.1 for discussion of key use policies across multiple certificates.

A BPSec end-entity certificate MAY contain an Online Certificate Status Protocol (OCSP) URI within an Authority Information Access extension in accordance with Section 4.2.2.1 of [RFC5280]. Security acceptors are not expected to have continuous internet connectivity sufficient to perform OCSP verification.

5.1. Multiple-Certificate Uses

A RECOMMENDED security policy is to limit asymmetric keys (and thus public key certificates) to single uses among the following:

Bundle transport: With key uses as defined in the convergence layer specification(s).

Block signing: With key use digitalSignature and/or nonRepudiation

Block encryption: With key use keyEncipherment and/or keyAgreement

This policy is the same one recommended by Section 6 of [RFC8551] for email security and by Section 5.2 of [NIST-SP800-57] more generally. Effectively this means that a BP node uses separate certificates for transport (e.g., as a TCPCL entity), BIB signing (as a security source), and BCB encryption (as a security acceptor).

6. 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.

7. 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.

7.1. 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.

7.2. Threat: Untrusted End-Entity Certificate

The profile in Section 3.6.1 uses end-entity certificates chained up to a trusted root CA. A security source can include a certificate set which does not contain the full chain, possibly excluding intermediate or root CAs. In an environment where security acceptors are known to already contain needed root and intermediate CAs there is no need to include those CAs, but this has a risk of an acceptor not actually having one of the needed CAs.

7.3. Threat: Certificate Validation Vulnerabilities

Even when a security acceptor is operating properly an attacker can attempt to exploit vulnerabilities within certificate check algorithms or configuration to authenticate using an invalid certificate. An invalid certificate exploit could lead to higher-level security issues and/or denial of service to the Node ID being impersonated.

There are many reasons, described in [RFC5280] and [RFC6125], why a certificate can fail to validate, including using the certificate outside of its valid time interval, using purposes for which it was not authorized, or using it after it has been revoked by its CA. Validating a certificate is a complex task and can require network connectivity outside of the primary BP convergence layer network path(s) if a mechanism such as OCSP [RFC6960] is used by the CA. The configuration and use of particular certificate validation methods are outside of the scope of this document.

7.4. 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.

7.5. Threat: Unidentifiable Key

The profile in Section 4.2 recommends key identifiers when possible and the parameters in section Section 3.2 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.

7.6. 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.

7.7. Threat: Passive Leak of Key Material

It is important that the key requirements of Section 3.2 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.

7.8. 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].

8. IANA Considerations

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

8.1. 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

9. Acknowledgments

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

10. References

10.1. 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>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://www.rfc-editor.org/info/rfc6125>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/info/rfc6960>.

[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-25, 1 December 2020, <https://tools.ietf.org/html/draft-ietf-dtn-bpsec-25>.

[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-tcpclv4-24, 7 December 2020, <https://tools.ietf.org/html/draft-ietf-dtn-tcpclv4-24>.

[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-x509-08, 14 December 2020, <https://tools.ietf.org/html/draft-ietf-cose-x509-08>.

10.2. Informative References

[NIST-SP800-57] National Institute of Standards and Technology, "Recommendation for Key Management - Part 1: General", NIST Special Publication 800-57 Revision 4, DOI 10.6028/NIST.SP.800-57pt1r5, May 2020, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf>.

[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-30, 10 December 2020, <https://tools.ietf.org/html/draft-ietf-dtn-bpbis-30>.

[I-D.ietf-dtn-bpsec-interop-sc]
Birrane, E., "BPSec Default Security Contexts", Work in Progress, Internet-Draft, draft-ietf-dtn-bpsec-interop-sc-02, 1 November 2020, <https://tools.ietf.org/html/draft-ietf-dtn-bpsec-interop-sc-02>.

[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.

All of these examples operate within the context of the bundle primary block of Figure 5 with a security target block of Figure 6. All example figures use the extended diagnostic notation [RFC8610].

```
[
  7, / BP version /
  0, / flags /
  0, / CRC type /
  [1, "//dst/svc"], / 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 /
  <<300>> / type-specific-data: age /
]
```

Figure 6: Target block CBOR diagnostic

All of the examples also operate within a security block containing the AAD Scope parameter with only "has-primary-ctx" and "has-target-ctx" flags set. This results in a consistent AAD-value as shown in Figure 7, which is used as the bytestring for COSE external_aad in all of the examples. Note that the AAD-value is itself a bytestring which happens to contain encoded CBOR.

```
<<[
  [ 7, 0, 0, [ 1, "//dst/svc" ], [ 1, "//src/" ], [ 1, "//src/" ],
    [ 0, 40 ], 1000000 ], / primary-ctx /
  [ 7, 2, 0 ], / target-ctx /
  null / security-ctx /
]>>
```

Figure 7: Example scope AAD-value CBOR diagnostic

The only differences between these examples which use EC or RSA keypairs 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 recipient would contain 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 or wrapped-key data.

A.1. Symmetric Key COSE_Mac0

This is an example of a MAC with recipient having a 256-bit symmetric key identified by a "kid".

```
[
  {
    / kty / 1: 4, / symmetric /
    / kid / 2: 'ExampleMAC',
    / k / -1: h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e39
99dbae4ce45c'
  }
]
```

Figure 8: Symmetric Key

The external_aad is the encoded data from Figure 7. The payload is the encoded target block-type-specific data from Figure 6.

```
[
  "MAC0", / context /
  h'a10105', / protected /
  h'83880700008201692f2f6473742f7376638201662f2f7372632f8201662f2f73
72632f820018281a000f424083070200f6', / external_aad /
  h'19012c' / payload /
]
```

Figure 9: MAC_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  1, / flags: params-present /
  [ / parameters /
    [
      5, / AAD-scope /
      0x03 / has-primary-ctx | has-target-ctx /
    ]
  ],
  [
    [ / target block #2 /
      [ / result /
        17, / COSE_Mac0 tag /
        [
          <<{ / protected /
             / alg / 1:5 / HMAC 256//256 /
          }>>,
          { / unprotected /
            / kid / 4:'ExampleMAC'
          },
          null, / payload /
          h'190264a1e6a9734990e552660df3c4641efb88fd6439aba866577c7b
6d174b87' / tag /
        ]
      ]
    ]
  ]
]
```

Figure 10: Abstract Security Block CBOR diagnostic

A.2. EC Keypair COSE_Sign1

This is an example of a signature with a recipient having a P-256 curve EC keypair identified by a "kid". The associated public key is included as a security parameter.

```
[
  { / signing private key /
    / kty / 1: 2, / EC2 /
    / kid / 2: 'ExampleEC2',
    / crv / -1: 1, / P-256 /
    / x / -2: h'44c1fa63b84f172b50541339c50beb0e630241ecb4eebbddb8b5
e4fe0a1787a8',
    / y / -3: h'059451c7630d95d0b550acbd02e979b3f4f74e645b74715fafbc
1639960a0c7a',
    / d / -4: h'dd6e7d8c4c0e0c0bd3ae1b4a2fa86b9a09b7efee4a233772cf51
89786ea63842'
  }
]
```

Figure 11: Example Keys

The external_aad is the encoded data from Figure 7. The payload is the encoded target block-type-specific data from Figure 6.

```
[
  "Signature1", / context /
  h'a10126', / protected /
  h'83880700008201692f2f6473742f7376638201662f2f7372632f8201662f2f73
72632f820018281a000f424083070200f6', / external_aad /
  h'19012c' / payload /
]
```

Figure 12: Sig_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  1, / flags: params-present /
  [ / parameters /
    [
      1, / COSE key /
      { / public key /
        / kty / 1: 2, / EC2 /
        / kid / 2: 'ExampleEC2',
        / crv / -1: 1, / P-256 /
        / x / -2: h'44c1fa63b84f172b50541339c50beb0e630241ecb4eebbdd
b8b5e4fe0a1787a8',
        / y / -3: h'059451c7630d95d0b550acbd02e979b3f4f74e645b74715f
afbc1639960a0c7a'
      }
    ],
    [
      5, / AAD-scope /
      0x03 / has-primary-ctx | has-target-ctx /
    ]
  ],
  [
    [ / target block #2 /
      [ / result /
        18, / COSE_Sign1 tag /
        [
          <<{ / protected /
             / alg / 1:-7 / ES256 /
          }>>,
          { / unprotected /
            / kid / 4:'ExampleEC2'
          },
          null, / payload /
          h'eb085c162bac4ec45c974766b897ee227b189fa257a8fb195c830f04
f6d6a90318b3e915938e4d32c1baace8aa0bf983f52efcbf1b127b296e72673f3d73
3023' / signature /
        ]
      ]
    ]
  ]
]
```

Figure 13: Abstract Security Block CBOR diagnostic

A.3. RSA Keypair COSE_Sign1

This is an example of a signature with a recipient having a 1024-bit RSA keypair identified by a "kid". The associated public key is included as a security parameter.

This key strength is not supposed to be a secure configuration, only intended to explain the procedure. This signature uses a random salt, so the full signature output is not deterministic.

```
[
  { / signing private key /
    / kty / 1: 3, / RSA /
    / kid / 2: 'ExampleRSA',
    / n / -1: h'b0b5fd85f52c91844007443c9f9371980025f76d51fc9c676812
31da610cb291ba637ce813bffdb2e9c653258607389ec97dad3db295fded67744ed6
20707db36804e74e56a494030a73608fc8d92f2f0578d2d85cc201ef0ff22d7835d2
d147d3b90a6884276235a01c2be99dfc597f79554362fc1eb03639cac5ccaddb29
25',
    / e / -2: h'010001',
    / d / -3: h'9b5d26ad6445ef1aab80b809e4f329684e9912d556c4166f041d
1b1fb93c04b4037ffd0dbe6f8a8a86e70bab6e0f6344983a9ada27ed9ff7de816fde
eb5e7be48e607ce5fda4581ca6338a9e019fb3689b28934192b6a190cdda910abb5a
86a2f7b6f9cd5011049d8de52ddfef73aa06df401c55623ec196720f54920deb4f
01',
    / p / -4: h'db22d94e7784a27b568cbf985307ea8d6430ff6b88c18a7086fd
4f57a326572f2250c39e48a6f8e2201661c2dfe12c7386835b649714d050aa36123e
c3d00e75',
    / q / -5: h'ce7016adc5f326b7520397c5978ee2f50e69279983d54c5d76f0
5bcd61de0879d7056c923540dff9cbae95dcc0e5e86b52b3c902dc9669c8021c6955
7effb9f1',
    / dP / -6: h'6a6fcaccea106a3b2e16bf18e57b7ad9a2488a4758ed68a8af6
86a194f0d585b7477760c738d6665aee0302bcf4237ad0530d83b4b86b887f5a4bdc
7eea427e1',
    / dQ / -7: h'28a4cae245b1dcb285142e027a1768b9c4af915b59285a93a04
22c60e05edd9e57663afd023d169bd0ad3bd62da8563d231840802ebbf271ad70b89
05ba3af91',
    / qInv / -8: h'07b5a61733896270a6bd2bb1654194c54e2bc0e061b543a4e
d9fa73c4bc79c87148aa92a451c4ab8262b6377a9c7b97f869160ca6f5d853ee4b65
f4f92865ca3'
  }
]
```

Figure 14: Example Keys

The external_aad is the encoded data from Figure 7. The payload is the encoded target block-type-specific data from Figure 6.

```
[
  "Signature1", / context /
  h'a1013824', / protected /
  h'83880700008201692f2f6473742f7376638201662f2f7372632f8201662f2f73
72632f820018281a000f424083070200f6', / external_aad /
  h'19012c' / payload /
]
```

Figure 15: Sig_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  1, / flags: params-present /
  [ / parameters /
    [
      1, / COSE key /
      { / public key /
        / kty / 1: 3, / RSA /
        / kid / 2: 'ExampleRSA',
        / n / -1: h'b0b5fd85f52c91844007443c9f9371980025f76d51fc9c67
681231da610cb291ba637ce813bffdb2e9c653258607389ec97dad3db295fded6774
4ed620707db36804e74e56a494030a73608fc8d92f2f0578d2d85cc201ef0ff22d78
35d2d147d3b90a6884276235a01c2be99dfc597f79554362fc1eb03639cac5ccaddb
2925',
        / e / -2: h'010001'
      }
    ],
    [
      5, / AAD-scope /
      0x03 / has-primary-ctx | has-target-ctx /
    ]
  ],
  [
    [ / target block #2 /
      [ / result /
        18, / COSE_Sign1 tag /
        [
          <<{ / protected /
             / alg / 1:-37 / PS256 /
          }>>,
          { / unprotected /
            / kid / 4:'ExampleRSA'
          },
          null, / payload /
          h'2229cec7cd4e77e55b7ef39e0305931527e3075e7cad4969ecf1bdc5
cb8662435128718c7ba465d2251a770a6c48ddc62f515fca43482ae137fffa67c86b
c60b3b838875621b276235bdc4269f45fd0c08fdd607650d03ae75b86364f7f5f2cc
442d60e72bff7939478deba7e3492ea96f8ac1f953583df897138f66c16bc2
07' / signature /
        ]
      ]
    ]
  ]
]
```

Figure 16: Abstract Security 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 "kid". The keys used are shown in Figure 17.

```
[
  {
    / kty / 1: 4, / symmetric /
    / kid / 2: 'ExampleKEK',
    / k / -1: h'0e8a982b921d1086241798032fedc1f883eab72e4e43bb2d11cf
ae38ad7a972e'
  },
  {
    / kty / 1: 4, / symmetric /
    / kid / 2: 'ExampleCEK',
    / k / -1: h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e39
99dbae4ce45c'
  }
]
```

Figure 17: Example Keys

The external_aad is the encoded data from Figure 7. The payload is the encoded target block-type-specific data from Figure 6.

```
[
  "Encrypt", / context /
  h'a10103', / protected /
  h'8388070000820169 2f2f6473742f7376638201662f2f7372632f8201662f2f73
72632f820018281a000f424083070200f6' / external_aad /
]
```

Figure 18: Enc_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  1, / flags: params-present /
  [ / parameters /
    [
      5, / AAD-scope /
      0x03 / has-primary-ctx | has-target-ctx /
    ]
  ],
  [
    [ / 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'917f2045e1169502756252bf119a94cdac6a9d8944245b5a9a26
d403a6331159e3d691a708e9984d' / key-wrapped /
            ]
          ]
        ]
      ]
    ]
  ]
]
```

Figure 19: Abstract Security Block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  h'63bb162d8ee2e8175cfc340b6df978864907a2' / ciphertext /
]
```

Figure 20: Encrypted Target block CBOR diagnostic

A.5. EC Keypair COSE_Encrypt

This is an example of an encryption with an P-256 curve ephemeral sender keypair and a static recipient keypair identified by a "kid". The keys used are shown in Figure 21.

```
[
  { / sender ephemeral private key /
    / kty / 1: 2, / EC2 /
    / crv / -1: 1, / P-256 /
    / x / -2: h'fedaba748882050d1bef8ba992911898f554450952070aeb4788
ca57d1df6bcc',
    / y / -3: h'ceaa8e7ff4751a4f81c70e98f1713378b0bd82a1414a2f493c1c
9c0670f28d62',
    / d / -4: h'a2e4ed4f2e21842999b0e9ebdaad7465efd5c29bd5761f5c2088
0f9d9c3b122a'
  },
  { / recipient private key /
    / kty / 1: 2, / EC2 /
    / kid / 2: 'ExampleEC2',
    / crv / -1: 1, / P-256 /
    / x / -2: h'44c1fa63b84f172b50541339c50beb0e630241ecb4eebbddb8b5
e4fe0a1787a8',
    / y / -3: h'059451c7630d95d0b550acbd02e979b3f4f74e645b74715fafbc
1639960a0c7a',
    / d / -4: h'dd6e7d8c4c0e0c0bd3ae1b4a2fa86b9a09b7efee4a233772cf51
89786ea63842'
  },
  {
    / kty / 1: 4, / symmetric /
    / kid / 2: 'ExampleCEK',
    / k / -1: h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e39
99dbae4ce45c'
  }
]
```

Figure 21: Example Keys

The external_aad is the encoded data from Figure 7. The payload is the encoded target block-type-specific data from Figure 6.

```
[
  "Encrypt", / context /
  h'a10103', / protected /
  h'83880700008201692f2f6473742f7376638201662f2f7372632f8201662f2f73
72632f820018281a000f424083070200f6' / external_aad /
]
```

Figure 22: Enc_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  1, / flags: params-present /
  [ / parameters /
    [
      1, / COSE key /
      { / public key /
        / kty / 1: 2, / EC2 /
        / kid / 2: 'ExampleEC2',
        / crv / -1: 1, / P-256 /
        / x / -2: h'44c1fa63b84f172b50541339c50beb0e630241ecb4eebbdd
b8b5e4fe0a1787a8',
        / y / -3: h'059451c7630d95d0b550acbd02e979b3f4f74e645b74715f
afbc1639960a0c7a'
      }
    ],
    [
      5, / AAD-scope /
      0x03 / has-primary-ctx | has-target-ctx /
    ]
  ],
  [
    [ / 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:-31, / ECDH-ES + A256KW /
                / kid / 4:'ExampleEC2',
                / ephemeral key / -1:{
                  1:2,
                  -1:1,
                  -2:h'fedaba748882050d1bef8ba992911898f554450952070
aeb4788ca57d1df6bcc',
                  -3:h'ceaa8e7ff4751a4f81c70e98f1713378b0bd82a1414a2
f493c1c9c0670f28d62'
                },
                / PartyU nonce / -22:h'e6bd83a5a06841c2ea1dd4eebaaa
```

```
f252'
              },
              h'e20b6fd9b46cdaae9e67ccf4893706802a7acb0c3b3a792b3fcb
a110f2f27d7972934f4e6497ac89' / key-wrapped /
            ]
          ]
        ]
      ]
    ]
  ]
]
```

Figure 23: Abstract Security Block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  h'63bb162d8ee2e8175cfc340b6df978864907a2' / ciphertext /
]
```

Figure 24: Encrypted Target block CBOR diagnostic

A.6. RSA Keypair COSE_Encrypt

This is an example of an encrypion with a recipient having a 1024-bit RSA keypair identified by a "kid". The associated public key is included as a security parameter.

This key strength is not supposed to be a secure configuration, only intended to explain the procedure. This padding scheme uses a random salt, so the full layer-2 ciphertext output is not deterministic.

```
[
  { / recipient private key /
    / kty / 1: 3, / RSA /
    / kid / 2: 'ExampleRSA',
    / n / -1: h'b0b5fd85f52c91844007443c9f9371980025f76d51fc9c676812
31da610cb291ba637ce813bffdb2e9c653258607389ec97dad3db295fded67744ed6
20707db36804e74e56a494030a73608fc8d92f2f0578d2d85cc201ef0ff22d7835d2
d147d3b90a6884276235a01c2be99dfc597f79554362fc1eb03639cac5ccaddb29
25',
    / e / -2: h'010001',
    / d / -3: h'9b5d26ad6445ef1aab80b809e4f329684e9912d556c4166f041d
1b1fb93c04b4037ffd0dbe6f8a8a86e70bab6e0f6344983a9ada27ed9ff7de816fde
eb5e7be48e607ce5fda4581ca6338a9e019fb3689b28934192b6a190cdda910abb5a
86a2f7b6f9cd5011049d8de52ddfef73aa06df401c55623ec196720f54920deb4f
01',
    / p / -4: h'db22d94e7784a27b568cbf985307ea8d6430ff6b88c18a7086fd
4f57a326572f2250c39e48a6f8e2201661c2dfe12c7386835b649714d050aa36123e
c3d00e75',
    / q / -5: h'ce7016adc5f326b7520397c5978ee2f50e69279983d54c5d76f0
5bcd61de0879d7056c923540dff9cbae95dcc0e5e86b52b3c902dc9669c8021c6955
7effb9f1',
    / dP / -6: h'6a6fcaccea106a3b2e16bf18e57b7ad9a2488a4758ed68a8af6
86a194f0d585b7477760c738d6665aee0302bcf4237ad0530d83b4b86b887f5a4bdc
7eea427e1',
    / dQ / -7: h'28a4cae245b1dcb285142e027a1768b9c4af915b59285a93a04
22c60e05edd9e57663afd023d169bd0ad3bd62da8563d231840802ebbf271ad70b89
05ba3af91',
    / qInv / -8: h'07b5a61733896270a6bd2bb1654194c54e2bc0e061b543a4e
d9fa73c4bc79c87148aa92a451c4ab8262b6377a9c7b97f869160ca6f5d853ee4b65
f4f92865ca3'
  },
  {
    / kty / 1: 4, / symmetric /
    / kid / 2: 'ExampleCEK',
    / k / -1: h'13bf9cead057c0aca2c9e52471ca4b19ddfaf4c0784e3f3e8e39
99dbae4ce45c'
  }
]
```

Figure 25: Example Keys

The external_aad is the encoded data from Figure 7. The payload is the encoded target block-type-specific data from Figure 6.

```
[
  "Encrypt", / context /
  h'a10103', / protected /
  h'83880700008201692f2f6473742f7376638201662f2f7372632f8201662f2f73
72632f820018281a000f424083070200f6' / external_aad /
]
```

Figure 26: Enc_structure CBOR diagnostic

```
[
  [2], / targets /
  0, / security context TBD /
  1, / flags: params-present /
  [ / parameters /
    [
      1, / COSE key /
      { / public key /
        / kty / 1: 3, / RSA /
        / kid / 2: 'ExampleRSA',
        / n / -1: h'b0b5fd85f52c91844007443c9f9371980025f76d51fc9c67
681231da610cb291ba637ce813bffdb2e9c653258607389ec97dad3db295fded6774
4ed620707db36804e74e56a494030a73608fc8d92f2f0578d2d85cc201ef0ff22d78
35d2d147d3b90a6884276235a01c2be99dfc597f79554362fc1eb03639cac5ccaddb
2925',
        / e / -2: h'010001'
      }
    ],
    [
      5, / AAD-scope /
      0x03 / has-primary-ctx | has-target-ctx /
    ]
  ],
  [
    [ / 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:-41, / RSAES-OAEP w SHA-256 /
                / kid / 4:'ExampleRSA'
              },
              h'69e76a39b908090f55b1048c95bd1683d9ce702fd9ed4a149650
72fd411936ade41a36a9f62921635d14406eb0e1fa3c02ca0d957a4a44006aa3c03e
326867964b166f8731ebcb20d413fd8f26c57c337689dc42235bfd2b928619b0d4f2
7ec118c608ad9d18c881bc5124833483ded5f5fb079805f3e299fa45f756ecc4c3
e6' / key-wrapped /
            ]
          ]
```

```
          ]
        ]
      ]
    ]
  ]
```

Figure 27: Abstract Security Block CBOR diagnostic

```
[
  7, / type code - bundle age /
  2, / block num /
  0, / flags /
  0, / CRC type /
  h'63bb162d8ee2e8175cfc340b6df978864907a2' / ciphertext /
]
```

Figure 28: 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