                     Bundle Protocol Security Specification
                          draft-ietf-dtn-bpsec-03

Abstract

   This document defines a security protocol providing end to end data
   integrity and confidentiality services for the Bundle Protocol.

Status of This Memo

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

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

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

   This Internet-Draft will expire on May 3, 2017.

Copyright Notice

Table of Contents

1. Introduction

   This document defines security features for the Bundle Protocol
   [BPBIS] intended for use in delay-tolerant networks, in order to
   provide Delay-Tolerant Networking (DTN) security services.

1.1. Motivation

   The Bundle Protocol is used in DTNs that overlay multiple networks,
   some of which may be challenged by limitations such as intermittent
   and possibly unpredictable loss of connectivity, long or variable
   delay, asymmetric data rates, and high error rates. The purpose of
   the Bundle Protocol is to support interoperability across such
   stressed networks.

   The stressed environment of the underlying networks over which the
   Bundle Protocol operates makes it important for the DTN to be
   protected from unauthorized use, and this stressed environment poses
   unique challenges for the mechanisms needed to secure the Bundle
   Protocol. Furthermore, DTNs may be deployed in environments where a
   portion of the network might become compromised, posing the usual
   security challenges related to confidentiality and integrity.

1.2. Supported Security Services

   This specification supports end-to-end integrity and confidentiality
   services associated with BP bundles.

   Integrity services ensure data within a bundle are not changed. Data
   changes may be caused by processing errors, environmental conditions,
   or intentional manipulation. An integrity service is one that
   provides sufficient confidence to a data receiver that data has not
   changed since its value was last asserted.

   Confidentiality services ensure that the values of some data within a
   bundle can only be determined by authorized receivers of the data.
   When a bundle traverses a DTN, many nodes in the network other than
   the destination node MAY see the contents of a bundle. A
   confidentiality service allows a destination node to generate data
   values from otherwise encrypted contents of a bundle.

   NOTE: Hop-by-hop authentication is NOT a supported security service
   in this specification, for three reasons.

1. The term "hop-by-hop" is ambiguous in a BP overlay, as nodes that
    are adjacent in the overlay may not be adjacent in physical
    connectivity. This condition is difficult or impossible to
    predict in the overlay and therefore makes the concept of hop-by-
    hop authentication difficult or impossible to enforce at the
    overlay.

2. Networks in which BPSec may be deployed may have a mixture of
    security-aware and not-security-aware nodes. Hop-by-hop
    authentication cannot be deployed in a network if adjacent nodes
    in the network have different security capabilities.

3. Hop-by-hop authentication can be viewed as a special case of data
    integrity. As such, it is possible to develop policy that
    provides a version of authentication using the integrity
    mechanisms defined in this specification.

1.3. Specification Scope

   This document describes the Bundle Protocol Security Specification
   (BPSec), which provides security services for blocks within a bundle.
   This includes the data specification for individual BP extension
   blocks and the processing instructions for those blocks.

   BPSec applies, by definition, only to those nodes that implement it,
   known as "security-aware" nodes. There MAY be other nodes in the DTN
   that do not implement BPSec. All nodes can interoperate with the
   exception that BPSec security operations can only happen at BPSec
   security-aware nodes.

   This specification does not address individual cipher suite
   implementations. The definition and enumeration of cipher suites
   should be undertaken in separate specification documents.

   This specification does not address the implementation of security
   policy and does not provide a security policy for the BPSec.
   Security policies are typically based on the nature and capabilities
   of individual networks and network operational concepts. However,
   this specification does recommend policy considerations when building
   a security policy.

   This specification does not address how to combine the BPSec security
   blocks with other protocols, other BP extension blocks, or other best
   practices to achieve security in any particular network
   implementation.

1.4. Related Documents

   This document is best read and understood within the context of the
   following other DTN documents:

   "Delay-Tolerant Networking Architecture" [RFC4838] defines the
   architecture for delay-tolerant networks, but does not discuss
   security at any length.

   The DTN Bundle Protocol [BPBIS] defines the format and processing of
   the blocks used to implement the Bundle Protocol, excluding the
   security-specific blocks defined here.

   The Bundle Security Protocol [RFC6257] and Streamlind Bundle Security
   Protocol [SBSP] introduce the concepts of security blocks for
   security services. BPSec is based off of these documents.

1.5. Terminology

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in
   [RFC2119].

   This section defines those terms whose definition is important to the
   understanding of concepts within this specification.

   o Source - the bundle node from which a bundle originates.

   o Destination - the bundle node to which a bundle is ultimately
      destined.

   o Forwarder - the bundle node that forwarded the bundle on its most
      recent hop.

   o Intermediate Receiver, Waypoint, or "Next Hop" - the neighboring
      bundle node to which a forwarder forwards a bundle.

   o Path - the ordered sequence of nodes through which a bundle passes
      on its way from source to destination. The path is not
      necessarily known by the bundle, or any bundle-aware nodes.

   The application of these terms applied to a sample network topology
   is shown in Figure 1. This figure shows four bundle nodes (BN1, BN2,
   BN3, BN4) residing above some transport layer(s). Three distinct
   transport and network protocols (T1/N1, T2/N2, and T3/N3) are also
   shown.

```
+---------v-|   +->>>>>>>>>>v-+    +->>>>>>>>>>v-+   +-^---------+
| BN1     v |   | ^   BN2   v |    | ^   BN3   v |   | ^   BN4   |
+---------v-+   +-^---------v-+    +-^---------v-+   +-^---------+
| T1      v |   + ^   T1/T2  v |   + ^   T2/T3  v |  | ^   T3    |
+---------v-+   +-^---------v-+    +-^---------v +   +-^---------+
| N1      v |   | ^   N1/N2  v |   | ^   N2/N3  v |  | ^   N3    |
+---------v-+   +-^---------v +    +-^---------v-+   +-^---------+
|       >>>>>>>>^          >>>>>>>>>>^           >>>>>>>>^       |
+----------+   +-----------+    +------------+   +-----------+
|              |               |                |              |
|<-- An Internet --->|        |<--- An Internet  --->|
|              |               |                |              |
```
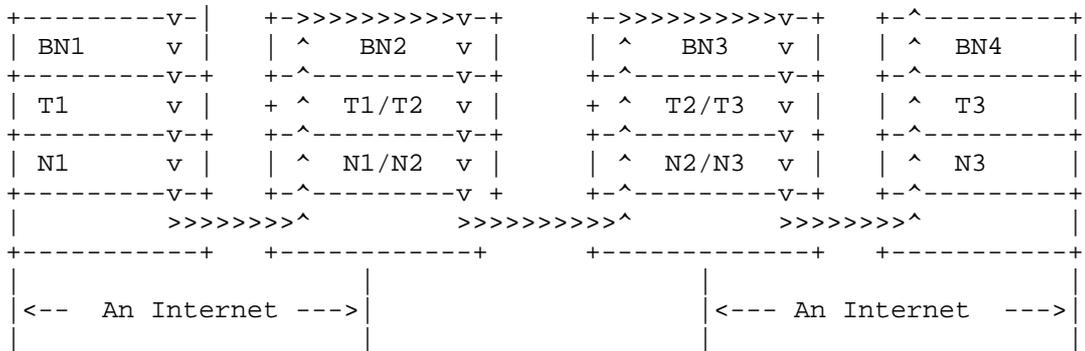
                Figure 1: Bundle Nodes Sitting Above the Transport Layer.

Consider the case where BN1 originates a bundle that it forwards to
BN2. BN2 forwards the bundle to BN3, and BN3 forwards the bundle to
BN4. BN1 is the source of the bundle and BN4 is the destination of
the bundle. BN1 is the first forwarder, and BN2 is the first
intermediate receiver; BN2 then becomes the forwarder, and BN3 the
intermediate receiver; BN3 then becomes the last forwarder, and BN4
the last intermediate receiver, as well as the destination.

If node BN2 originates a bundle (for example, a bundle status report
or a custodial signal), which is then forwarded on to BN3, and then
to BN4, then BN2 is the source of the bundle (as well as being the
first forwarder of the bundle) and BN4 is the destination of the
bundle (as well as being the final intermediate receiver).

The following security-specific terminology is also defined to
clarify security operations in this specifiation.

o Security Service - the security features supported by this
   specification: integrity and confidentiality.

o Security Source - a bundle node that adds a security block to a
   bundle.

o Security Target - the block within a bundle that receives a
   security-service as part of a security-operation.

o Security Block - a BPSec extension block in a bundle.

o Security Operation - the application of a security service to a
   security target, notated as OP(security service, security target).
   For example, OP(confidentiality, payload). Every security

operation in a bundle MUST be unique, meaning that a security
service can only be applied to a security target once in a bundle.
A security operation is implemented by a security block.

2. Key Properties

The application of security services in a DTN is a complex endeavor
that must consider physical properties of the network, policies at
each node, and various application security requirements. Rather
than enumerate all potential security implementations in all
potential DTN topologies, this specification defines a set of key
properties of a security system. The security primitives outlined in
this document MUST enable the realization of these properties in a
DTN deploying the Bundle Protocol.

2.1. Block-Level Granularity

Blocks within a bundle represent different types of information. The
primary block contains identification and routing information. The
payload block carries application data. Extension blocks carry a
variety of data that may augment or annotate the payload, or
otherwise provide information necessary for the proper processing of
a bundle along a path. Therefore, applying a single level and type
of security across an entire bundle fails to recognize that blocks in
a bundle may represent different types of information with different
security needs.

Security services within this specification MUST provide block level
granularity where applicable such that different blocks within a
bundle may have different security services applied to them.

For example, within a bundle, a payload might be encrypted to protect
its contents, whereas an extension block containing summary
information related to the payload might be integrity signed but
otherwise unencrypted to provide certain nodes access to payload-
related data without providing access to the payload.

Each security block in a bundle will be associated with a specific
security operation.

2.2. Multiple Security Sources

A bundle MAY have multiple security blocks and these blocks MAY have
different security sources.

The Bundle Protocol allows extension blocks to be added to a bundle
at any time during its existence in the DTN. When a waypoint node
adds a new extension block to a bundle, that extension block may have

security services applied to it by that waypoint. Similarly, a
waypoint node may add a security service to an existing extension
block, consistent with its security policy. For example, a node
representing a boundary between a trusted part of the network and an
untrusted part of the network may wish to apply payload encryption
for bundles leaving the trusted portion of the network.

In each case, a node other than the bundle originator may add a
security service to the bundle and, as such, the source for the
security service will be different than the source of the bundle
itself. Security services MUST track their orginating node so as to
properly apply policy and key selection associated with processing
the security service at the bundle destination.

Referring to Figure 1, if the bundle that originates at BN1 is given
security blocks by BN1, then BN1 is the security source for those
blocks as well as being the source of the bundle. If the bundle that
originates at BN1 is then given a security block by BN2, then BN2 is
the security source for that block even though BN1 remains the bundle
source.

2.3. Mixed Security Policy

Different nodes in a DTN may have different security related
capabilities. Some nodes may not be security aware and will not
understand any security related extension blocks. Other nodes may
have security policies that require evaluation of security services
at places other than the bundle destination (such as verifying
integrity signatures at certain waypoint nodes). Other nodes may
ignore any security processing if they are not the destination of the
bundle. The security services described in this specification must
allow each of these scenarios.

Extension blocks representing security services MUST have their block
processing flags set such that the block will be treated
appropriately by non-security-aware nodes.

Extension blocks providing integrity services within a bundle MUST
support options to allow waypoint nodes to evaluate these signatures
if such nodes have the proper configuraton to do so.

2.4. User-Selected Ciphersuites

The security services defined in this specification rely on a variety
of cipher suites providing integrity signatures, ciphertext, and
other information necessary to populate security blocks. Users may
wish to select different cipher suites to implement different
security services. For example, some users may wish to use a SHA-256

based hash for integrity whereas other users may require a SHA-384
hash instead. The security services defined in this specification
MUST provide a mechanism for identifying what cipher suite has been
used to populate a security block.

2.5. Deterministic Processing

In all cases, the processing order of security services within a
bundle must avoid ambiguity when evaluating security at the bundle
destination. This specification MUST provide determinism in the
application and evaluation of security services, even when doing so
results in a loss of flexibility.

3. Security Block Definitions

There are two types of security blocks that may be included in a
bundle. These are the Block Integrity Block (BIB) and the Block
Confidentiality Block (BCB).

   The BIB is used to ensure the integrity of its security target(s).
   The integrity information in the BIB MAY (when possible) be
   verified by any node in between the BIB security source and the
   bundle destination. BIBs MAY be added to, and removed from,
   bundles as a matter of security policy.

   The BCB indicates that the security target(s) has been encrypted,
   in whole or in part, at the BCB security source in order to
   protect its content while in transit. The BCB may be decrypted by
   appropriate nodes in the network, up to and including the bundle
   destination, as a matter of security policy.

A security operation MUST NOT be applied more than once in a bundle.
For example, the two security operations: OP(integrity, payload) and
OP(integrity, payload) are considered redundant and MUST NOT appear
together in a bundle. However, the two security operations
OP(integrity, payload) and OP(integrity, extension_block_1) MAY both
be present in the bundle. Also, the two security operations
OP(integrity, extension_block_1) and OP(integrity, extension_block_2)
are unique and may both appear in the same bundle.

If the same security service is to be applied to multiple security
targets, and cipher suite parameters for each security service are
identical, then the set of security operations can be represented as
a single security block with multiple security targets. In such a
case, all security operations represented in the security block MUST
be applied/evaluated together.

3.1. Block Identification

   This specification requires that every target block of a security
   operation be uniquely identifiable. The definition of the extension
   block header from [BPBIS] provides such a mechanism in the "Block
   Number" field, which provides a unique identifier for a block within
   a bundle. Within this specification, a security target will be
   identified by its unique Block Number.

   A security block MAY apply to multiple security targets if and only
   if all cipher suite parameters, security source, and key information
   are common for the security operation. In such a case, the security
   block MUST contain security results for each covered security target.
   The use of multiple security targets in a security block provides an
   efficiency mechanism so that identical ciphersuite information does
   not need to be repeated across multiple security blocks.

3.2. Block Representation

   Each security block uses the Canonical Bundle Block Format as defined
   in [BPBIS]. That is, each security block is comprised of the
   following elements:

   o Block Type Code

   o Block Number

   o Block Processing Control Flags

   o CRC Type and CRC Field

   o Block Data Length

   o Block Type Specific Data Fields

   The structure of the BIB and BCB Block Type Specific Data fields are
   identifcal and illustrated in Figure 2. In this figure, field names
   prefaced with an '*' are optional and their inclusion in the block is
   indicated by the Cipher Suite Flags field.

```
+=================================================
|    Field Name      |      Field Data Type      |
+=================================================
| # Security Targets | Unsigned Integer          |
+--------------------+---------------------------+
| Security Targets   | Array (Unsigned Integer)  |
+--------------------+---------------------------+
| Cipher Suite ID    | Unsigned Integer          |
+--------------------+---------------------------+
| Cipher Suite Flags | Unsigned Integer          |
+--------------------+---------------------------+
| Security Source    | URI - OPTIONAL            |
+--------------------+---------------------------+
| Cipher Parameters  | Byte Array - OPTIONAL     |
+--------------------+---------------------------+
| Security Result    | Byte Array                |
+--------------------+---------------------------+
```

Figure 2: BIB and BCB Block Structure

Where the block fields are identified as follows.

o # Security Targets - The number of security targets for this
   security block. This value MUST be at least 1.

o Security Targets - This array contains the unique identifier of
   the blocks targetted by this security operation. Each security
   target MUST represent a block present in the bundle. A security
   target MUST NOT be repeated in this array.

o Cipher suite ID - Identifies the cipher suite used to implement
   the security service represented by this block and applied to each
   security target.

o Cipher suite flags - Identifies which optional security block
   fields are present in the block. The structure of the Cipher
   Suite Flags field is shown in Figure 3. The presence of an
   optional field is indicated by setting the value of the
   corresponding flag to one. A value of zero indicates the
   corresponding optional field is not present. The BPSEC Cipher
   Suite Flags are defined as follows.

```
           Bit   Bit   Bit   Bit   Bit   Bit   Bit   Bit
            7     6     5     4     3     2     1     0
          +-------------------------------+-----+-----+
          |      reserved                 | src |parm |
          +-------------------------------+-----+-----+
            MSB                                   LSB
```
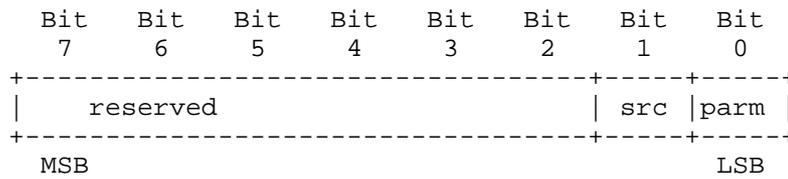
Figure 3: Cipher Suite Flags

Where:

* bits 7-2 are reserved for future use.

* src - bit 1 indicates whether the Security Source is present in the block.

* parm - bit 0 indicates whether or not the Cipher Suite Parameters field is present in the block.

o (OPTIONAL) Security Source (URI) - This identifies the node that inserted the security service in the bundle. If the security source is not present then the source MAY be inferred from the bundle source, the previous hop, or some other node as defined by security policy.

o (OPTIONAL) Parameters (Byte Array) - Compound field of the following two items.

* Length (Unsigned Integer) - specifies the length of the next field, which captures the parameters data.

* Data (Byte Array) - A byte array encoding one or more cipher suite parameters, with each parameter represented as a Type-Length-Value (TLV) triplet, defined as follows.

+ Type (Byte) - The parameter type.

+ Length (Unsigned Integer) - The length of the parameter.

+ Value (Byte Array) - The parameter value.

See Section 3.6 for a list of parameter types that MUST be supported by BPSEC implementations. BPSEC cipher suite specifications MAY define their own parameters to be represented in this byte array.

o Security Result (Byte Array) - A security result is the output of an appropriate cipher suite specific calculation (e.g., a

signature, Message Authentication Code (MAC), or cipher-text block key). There MUST exist one security result for each security target in the security block. A security result is a multi-field component, described as follows.

* Total Length (Unsigned Integer) - specifies the length, in bytes, of the remaining security result information.

* Results (Byte Array) - This field captures each of the security results, catenated together, one for each security target covered by the security block. Each result is captured by the four-tuple of (Target, Type, Len, Value). The meaning of each is given below.

  + Target (Optional) (Unsigned Integer) - If the security block has multiple security targets, the target field is the Block Number of the security target to which this result field applies. If the security block only has a single security target, this field is omitted.

  + Type (Unsigned Integer) - The type of security result field.

  + Length (Unsigned Integer) - The length of the result field.

  + Value (Byte Array) - The results of the cipher suite specific calculation.

3.3. Block Integrity Block

   A BIB is an ASB with the following characteristics:

   The Block Type Code value MUST be 0x02.

   The Block Processing Control flags value can be set to whatever values are required by local policy. Cipher suite designers should carefully consider the effect of setting flags that either discard the block or delete the bundle in the event that this block cannot be processed.

   A security target for a BIB MUST NOT reference a security block defined in this specification (e.g., a BIB or a BCB).

   The cipher suite ID MUST be documented as an end-to-end authentication-cipher suite or as an end-to-end error-detection-cipher suite.

   An EID-reference to the security source MAY be present. If this field is not present, then the security source of the block SHOULD

be inferred according to security policy and MAY default to the
bundle source. The security source may also be specified as part
of key information described in Section 3.6.

The security result captures the result of applying the cipher
suite calculation (e.g., the MAC or signature) to the relevant
parts of the security target, as specified in the cipher suite
definition. This field MUST be present.

The cipher suite MAY process less than the entire security target.
If the cipher suite processes less than the complete, original
security target, the cipher suite parameters MUST specify which
bytes of the security target are protected.

Notes:

o Since OP(integrity, target) is allowed only once in a bundle per
   target, it is RECOMMENDED that users wishing to support multiple
   integrity signatures for the same target define a multi-signature
   cipher suite.

o For some cipher suites, (e.g., those using asymmetric keying to
   produce signatures or those using symmetric keying with a group
   key), the security information MAY be checked at any hop on the
   way to the destination that has access to the required keying
   information, in accordance with Section 3.5.

o The use of a generally available key is RECOMMENDED if custodial
   transfer is employed and all nodes SHOULD verify the bundle before
   accepting custody.

3.4. Block Confidentiality Block

A BCB is an ASB with the following characteristics:

The Block Type Code value MUST be 0x03.

The Block Processing Control flags value can be set to whatever
values are required by local policy, except that this block MUST
have the "replicate in every fragment" flag set if the target of
the BCB is the Payload Block. Having that BCB in each fragment
indicates to a receiving node that the payload portion of each
fragment represents cipher-text. Cipher suite designers should
carefully consider the effect of setting flags that either discard
the block or delete the bundle in the event that this block cannot
be processed.

A security target for a BCB MAY reference the payload block, a
non-security extension block, or a BIB block. A security target
in a BCB MUST NOT be another BCB.

The cipher suite ID MUST be documented as a confidentiality cipher
suite.

Any additional bytes generated as a result of encryption and/or
authentication processing of the security target SHOULD be placed
in an "integrity check value" field (see Section 3.6) or other
such appropriate area in the security result of the BCB.

An EID-reference to the security source MAY be present. If this
field is not present, then the security source of the block SHOULD
be inferred according to security policy and MAY default to the
bundle source. The security source may also be specified as part
of key information described in Section 3.6.

The security result MUST be present in the BCB. This compound
field normally contains fields such as an encrypted bundle
encryption key and/or authentication tag.

The BCB modifies the contents of its security target. When a BCB is
applied, the security target body data are encrypted "in-place".
Following encryption, the security target body data contains cipher-
text, not plain-text. Other security target block fields (such as
type, processing control flags, and length) remain unmodified.

Fragmentation, reassembly, and custody transfer are adversely
affected by a change in size of the payload due to ambiguity about
what byte range of the block is actually in any particular fragment.
Therefore, when the security target of a BCB is the bundle payload,
the BCB MUST NOT alter the size of the payload block body data.
Cipher suites SHOULD place any block expansion, such as
authentication tags (integrity check values) and any padding
generated by a block-mode cipher, into an integrity check value item
in the security result field (see Section 3.6) of the BCB. This "in-
place" encryption allows fragmentation, reassembly, and custody
transfer to operate without knowledge of whether or not encryption
has occurred.

Notes:

o The cipher suite MAY process less than the entire original
   security target body data. If the cipher suite processes less
   than the complete, original security target body data, the BCB for
   that security target MUST specify, as part of the cipher suite
   parameters, which bytes of the body data are protected.

   o The BCB's "discard" flag may be set independently from its
      security target's "discard" flag. Whether or not the BCB's
      "discard" flag is set is an implementation/policy decision for the
      encrypting node. (The "discard" flag is more properly called the
      "Discard if block cannot be processed" flag.)

   o A BCB MAY include information as part of additional authenticated
      data to address parts of the target block, such as EID references,
      that are not converted to cipher-text.

3.5. Block Interactions

   The security block types defined in this specification are designed
   to be as independent as possible. However, there are some cases
   where security blocks may share a security target creating processing
   dependencies.

   If confidentiality is being applied to a target that already has
   integrity applied to it, then an undesirable condition occurs where a
   security aware intermediate node would be unable to check the
   integrity result of a block because the block contents have been
   encrypted after the integrity signature was generated. To address
   this concern, the following processing rules MUST be followed.

   o If confidentiality is to be applied to a target, it MUST also be
      applied to any integrity operation already defined for that
      target. This means that if a BCB is added to encrypt a block,
      another BCB MUST also be added to encrypt a BIB also targeting
      that block.

   o An integrity operation MUST NOT be applied to a security target if
      a BCB in the bundle shares the same security target. This
      prevents ambiguity in the order of evaluation when receiving a BIB
      and a BCB for a given security target.

   o An integrity value MUST NOT be evaluated if the BIB providing the
      integrity value is the security target of an existing BCB block in
      the bundle. In such a case, the BIB data contains cipher-text as
      it has been encrypted.

   o An integrity value MUST NOT be evaluated if the security target of
      the BIB is also the security target of a BCB in the bundle. In
      such a case, the security target data contains cipher-text as it
      has been encrypted.

   o As mentioned in Section 3.3, a BIB MUST NOT have a BCB as its
      security target. BCBs may embed integrity results as part of
      cipher suite parameters.

These restrictions on block interactions impose a necessary ordering when applying security operations within a bundle. Specifically, for a given security target, BIBs MUST be added before BCBs. This ordering MUST be preserved in cases where the current BPA is adding all of the security blocks for the bundle or whether the BPA is a waypoint adding new security blocks to a bundle that already contains security blocks.

3.6. Parameters and Result Fields

Various cipher suites include several items in the cipher suite parameters and/or security result fields. Which items MAY appear is defined by the particular cipher suite description. A cipher suite MAY support several instances of the same type within a single block.

Each item is represented as a type-length-value. Type is a single byte indicating the item. Length is the count of data bytes to follow, and is an Unsigned Integer. Value is the data content of the item.

Item types, name, and descriptions are defined as follows.

Cipher suite parameters and result fields.

Type	Name	Description	Field
0	Reserved		
1	Initialization Vector (IV)	A random value, typically eight to sixteen bytes.	Cipher Suite Parameters
2	Reserved		
3	Key Information	Material encoded or protected by the key management system and used to transport an ephemeral key protected by a long-term key.	Cipher Suite Parameters
4	Content Range	Pair of Unsigned Integers (offset,length) specifying the range of payload bytes to which an operation applies. The offset MUST be the offset within the	Cipher Suite Parameters

		original bundle, even if the current bundle is a fragment.	
5	Integrity Signatures	Result of BIB digest or other signing operation.	Security Results
6	Unassigned		
7	Salt	An IV-like value used by certain confidentiality suites.	Cipher Suite Parameters
8	BCB Integrity Check Value (ICV) / Authentication Tag	Output from certain confidentiality cipher suite operations to be used at the destination to verify that the protected data has not been modified. This value MAY contain padding if required by the cipher suite.	Security Results
9-255	Reserved		

Table 1

3.7. BSP Block Example

   An example of BPSec blocks applied to a bundle is illustrated in
   Figure 4. In this figure the first column represents blocks within a
   bundle and the second column represents a unique identifier for each
   block, suitable for use as the security target of a BPSec security
   block. Since the mechanism and format of a security target is not
   specified in this document, the terminology B1...Bn is used to
   identify blocks in the bundle for the purposes of illustration.

```
           Block in Bundle          ID
   +===============================+====+
   |          Primary Block        | B1 |
   +-------------------------------+----+
   |              BIB              | B2 |
   |   OP(integrity, target=B1)    |    |
   +-------------------------------+----+
   |              BCB              | B3 |
   |   OP(confidentiality, target=B4) |    |
   +-------------------------------+----+
   |         Extension Block       | B4 |
   +-------------------------------+----+
   |              BIB              | B5 |
   |   OP(integrity, target=B6)    |    |
   +-------------------------------+----+
   |         Extension Block       | B6 |
   +-------------------------------+----+
   |              BCB              | B7 |
   |   OP(confidentiality,target=B8,B9) |    |
   +-------------------------------+----+
   |    BIB   (encrypted by B7)    | B8 |
   |   OP(integrity, target=B9)    |    |
   +-------------------------------+----|
   |          Payload Block        | B9 |
   +-------------------------------+----+
```

Figure 4: Sample Use of BSP Blocks

In this example a bundle has four non-security-related blocks: the
primary block (B1), three extension blocks (B4,B6), and a payload
block (B9). The following security applications are applied to this
bundle.

o An integrity signature applied to the canonicalized primary block.
   This is accomplished by a single BIB (B2).

o Confidentiality for the first extension block (B4). This is
   accomplished by a BCB block (B3).

o Integrity for the second extension block (B6). This is
   accomplished by a BIB block (B5). NOTE: If the extension block B6
   contains a representation of the serialized bundle (such as a hash
   over all blocks in the bundle at the time of its last
   transmission) then the BIB block is also providing an
   authentication service from the prior BPSEC-BPA to this BPSEC-BPA.

o An integrity signature on the payload (B10). This is accomplished
   by a BIB block (B8).

   o Confidentiality for the payload block and it's integrity
      signature. This is accomplished by a BCB block, B7, encrypting B8
      and B9.

4. Canonical Forms

   By definition, an integrity service determines whether any aspect of
   a block was changed from the moment the security service was applied
   at the security source until the point of current evaluation. To
   successfully verify the integrity of a block, the data passed to the
   verifying cipher suite MUST be the same bits, in the same order, as
   those passed to the signature-generating cipher suite at the security
   source.

   However, [BPBIS] does not specify a single on-the-wire encoding of
   bundles. In cases where a security source generates a different
   encoding than that used at a receiving node, care MUST be taken to
   ensure that the inputs to cipher suites at the receiving node is a
   bitwise match to inputs provided at the security source.

   This section provides guidance on how to create a canonical form for
   each type of block in a bundle. This form MUST be used when
   generating inputs to cipher suites for use by BPSec blocks.

   This specification does not define any security operation over the
   entire bundle and, therefore, provides no canonical form for a
   serialized bundle.

4.1. Technical Notes

   The following technical considerations hold for all canonicalizations
   in this section.

   o Any numeric fields defined as variable-length MUST be expanded to
      their "unpacked" form. For example, a 32-bit integer value MUST
      be unpacked to a four-byte representation.

   o Each block encoding MUST follow the CBOR encodings provided in
      [BPBISCBOR].

   o Canonical forms are not transmitted, they are used to generate
      input to a cipher suite for secuity processing at a security-aware
      node.

   o Reserved flags MUST NOT be included in any canonicalization as it
      is not known if those flags will chaneg in transit.

   o These canonicalization algorithms assume that endpoint IDs
      themselves are immutable and they are unsuitable for use in
      environments where that assumption might be violated.

   o Cipher suites MAY define their own canonicalization algorithms and
      require the use of those algorithms over the ones provided in this
      specification. In the event of conflicting canonicalization
      algorithms, cipher suite algorithms take precedence over this
      specification.

4.2. Primary Block Canonicalization

   The primary block canonical form is the same as the CBOR encoding of
   the block, with certain modifications to account for allowed block
   changes as the bundle traverses the DTN. The fields that compromise
   the primary block, and any special considerations for their
   representation in a canonical form, are as follows.

   o The Version field is included, without modification.

   o The Bundle Processing Flags field is used, with modification.
      Certain bundle processing flags MAY change as a bundle transits
      the DTN without indicating an integrity error. These flags, which
      are identified below, MUST NOT be represented in the canonicalized
      form of the bundle processing flags and, instead, be represented
      by the bit 0.

      * Reserved flags.

      * Bundle is a Fragment flag.

   o The CRC Type, Destination EID, Source Node ID, Report-To EID,
      Creation Timestamp, and Lifetime fields are included, without
      modification.

   o The fragment ID field MAY change if the bundle is fragmented in
      transit and, as such, this field MUST NOT be included in the
      canonicalization.

   o The CRC field MAY change at each hop - for example, if a bundle
      becomes fragmented, each fragment will have a different CRC value
      from the original signed primary block. As such, this field MUST
      NOT be included in the canonicalization.

4.3. Non-Primary-Block Canonicalization

   All non-primary blocks (NPBs) in [BPBIS] share the same block
   structure and should be canonicalized in the same way.

   Canonicalization for NPBs is dependent on whether the security
   operation being performed is integrity or confidentiality. Integrity
   operations consider every field in the block, whereas confidentiality
   operations only consider the block-type-specific data. Since
   confidentiality is applied to hide information (replacing plaintext
   with ciphertext) it provides no benefit to include in the
   confidentiality calculation information that MUST remain readable,
   such as block fields other than the block-type-specific data.

   The fields that comprise a NPB, and any special considerations for
   their representation in a canonical form, are as follows.

   o The Block Type Code field is included, without modification, for
      integrity operations and omitted for confidentiality operations.

   o The Block Number field is included, without modification, for
      integrity operations and omitted for confidentiality operations.

   o The Block Processing Control Flags field is included, without
      modification, for integrity operations and omitted for
      confidentiality operations, with the exception of reserved flags
      which are treated as 0 in both cases.

   o The CRC type and CRC fields are included, without modification,
      for integrity operations and omitted for confidentiality
      operations.

   o The Block Type Specific Data field is included, without
      modification, for both integrity and confidentiality operations,
      with the exception that in some cases only a portion of the
      payload data is to be processed. In such a case, only those bytes
      are included in the canonical form and additional cipher suite
      parameters are required to specify which part of the field is
      included.

5. Security Processing

   This section describes the security aspects of bundle processing.

5.1. Bundles Received from Other Nodes

   Security blocks MUST be processed in a specific order when received
   by a security-aware node. The processing order is as follows.

   o All BCB blocks in the bundle MUST be evaluated prior to evaluating
      any BIBs in the bundle. When BIBs and BCBs share a security
      target, BCBs MUST be evaluated first and BIBs second.

5.1.1. Receiving BCB Blocks

   If a received bundle contains a BCB, the receiving node MUST
   determine whether it has the responsibility of decrypting the BCB
   security target and removing the BCB prior to delivering data to an
   application at the node or forwarding the bundle.

   If the receiving node is the destination of the bundle, the node MUST
   decrypt any BCBs remaining in the bundle. If the receiving node is
   not the destination of the bundle, the node MAY decrypt the BCB if
   directed to do so as a matter of security policy.

   If the relevant parts of an encrypted payload block cannot be
   decrypted (i.e., the decryption key cannot be deduced or decryption
   fails), then the bundle MUST be discarded and processed no further.
   If an encrypted security target other than the payload block cannot
   be decrypted then the associated security target and all security
   blocks associated with that target MUST be discarded and processed no
   further. In both cases, requested status reports (see [BPBIS]) MAY
   be generated to reflect bundle or block deletion.

   When a BCB is decrypted, the recovered plain-text MUST replace the
   cipher-text in the security target body data

   If a BCB contains multiple security targets, all security targets
   MUST be processed if the BCB is processed by the Node. The effect of
   this is to be the same as if each security target had been
   represented by an individual BCB with a single security target.

5.1.2. Receiving BIB Blocks

   If a received bundle contains a BIB, the receiving node MUST
   determine whether it has the responsibility of verifying the BIB
   security target and whether to remove the BIB prior to delivering
   data to an application at the node or forwarding the bundle.

   A BIB MUST NOT be processed if the security target of the BIB is also
   the security target of a BCB in the bundle. Given the order of
   operations mandated by this specification, when both a BIB and a BCB

share a security target, it means that the security target MUST have
been encrypted after it was integrity signed and, therefore, the BIB
cannot be verified until the security target has been decrypted by
processing the BCB.

If the security policy of a security-aware node specifies that a
bundle should have applied integrity to a specific security target
and no such BIB is present in the bundle, then the node MUST process
this security target in accordance with the security policy. This
MAY involve removing the security target from the bundle. If the
removed security target is the payload or primary block, the bundle
MAY be discarded. This action may occur at any node that has the
ability to verify an integrity signature, not just the bundle
destination.

If the bundle has a BIB and the receiving node is the destination for
the bundle, the node MUST verify the security target in accordance
with the cipher suite specification. If a BIB check fails, the
security target has failed to authenticate and the security target
SHALL be processed according to the security policy. A bundle status
report indicating the failure MAY be generated. Otherwise, if the
BIB verifies, the security target is ready to be processed for
delivery.

If the bundle has a BIB and the receiving node is not the bundle
destination, the receiving node MAY attempt to verify the value in
the security result field. If the check fails, the node SHALL
process the security target in accordance to local security policy.
It is RECOMMENDED that if a payload integrity check fails at a
waypoint that it is processed in the same way as if the check fails
at the destination.

If a BIB contains multiple security targets, all security targets
MUST be processed if the BIB is processed by the Node. The effect of
this is to be the same as if each security target had been
represented by an individual BIB with a single security target.

5.2. Bundle Fragmentation and Reassembly

If it is necessary for a node to fragment a bundle and security
services have been applied to that bundle, the fragmentation rules
described in [BPBIS] MUST be followed. As defined there and repeated
here for completeness, only the payload may be fragmented; security
blocks, like all extension blocks, can never be fragmented.

Due to the complexity of bundle fragmentation, including the
possibility of fragmenting bundle fragments, integrity and
confidentiality operations are not to be applied to a bundle

representing a fragment (i.e., a bundle whose "bundle is a Fragment" flag is set in the Bundle Processing Control Flags field). Specifically, a BCB or BIB MUST NOT be added to a bundle fragment, even if the security target of the security block is not the payload. When integrity and confidentiality must be applied to a fragment, we RECOMMEND that encapsulation be used instead.

6. Key Management

    Key management in delay-tolerant networks is recognized as a difficult topic and is one that this specification does not attempt to solve.

7. Policy Considerations

    When implementing BPSec, several policy decisions must be considered. This section describes key policies that affect the generation, forwarding, and receipt of bundles that are secured using this specification.

    o If a bundle is received that contains more than one security operation, in violation of BPSec, then the BPA must determine how to handle this bundle. The bundle may be discarded, the block affected by the security operation may be discarded, or one security operation may be favored over another.

    o BPAs in the network MUST understand what security operations they should apply to bundles. This decision may be based on the source of the bundle, the destination of the bundle, or some other information related to the bundle.

    o If an intermediate receiver has been configured to add a security operation to a bundle, and the received bundle already has the security operation applied, then the receiver MUST understand what to do. The receiver may discard the bundle, discard the security target and associated BPSec blocks, replace the security operation, or some other action.

    o It is recommended that security operations only be applied to the payload block, the primary block, and any block-types specifically identified in the security policy. If a BPA were to apply security operations such as integrity or confidentiality to every block in the bundle, regardless of the block type, there could be downstream errors processing blocks whose contents must be inspected at every hop in the network path.

   o Adding a BIB to a security target that has already been encrypted
      by a BCB is not allowed. Therefore, we recommend three methods to
      add an integrity signature to an encrypted security target.

      1. At the time of encryption, an integrity signature may be
          generated and added to the BCB for the security target as
          additional information in the security result field.

      2. The encrypted block may be replicated as a new block and
          integrity signed.

      3. An encapsulation scheme may be applied to encapsulate the
          security target (or the entire bundle) such that the
          encapsulating structure is, itself, no longer the security
          target of a BCB and may therefore be the security target of a
          BIB.

8. Security Considerations

   Given the nature of delay-tolerant networking applications, it is
   expected that bundles may traverse a variety of environments and
   devices which each pose unique security risks and requirements on the
   implementation of security within BPSEC. For these reasons, it is
   important to introduce key threat models and describe the roles and
   responsibilities of the BPSEC protocol in protecting the
   confidentiality and integrity of the data against those threats
   throughout the DTN. This section provides additional discussion on
   security threats that BPSEC will face and describe in additional
   detail how BPSEC security mechanisms operate to mitigate these
   threats.

   It should be noted that BPSEC addresses only the security of data
   traveling over the DTN, not the underlying DTN itself. Additionally,
   BPSEC addresses neither the fitness of externally-defined
   cryptographic methods nor the security of their implementation. It
   is the responsibility of the BPSEC implementer that appropriate
   algorithms and methods are chosen. Furthermore, the BPSEC protocol
   does not address threats which share computing resources with the DTN
   and/or BPSEC software implementations. These threats may be
   malicious software or compromised libraries which intend to intercept
   data or recover cryptographic material. Here, it is the
   responsibility of the BPSEC implementer to ensure that any
   cryptographic material, including shared secret or private keys, is
   protected against access within both memory and storage devices.

   The threat model described here is assumed to have a set of
   capabilities identical to those described by the Internet Threat
   Model in [RFC3552], but the BPSEC threat model is scoped to

illustrate threats specific to BPSEC operating within DTN
environments and therefore focuses on man-in-the-middle (MITM)
attackers.

8.1. Attacker Capabilities and Objectives

BPSEC was designed to protect against MITM threats which may have
access to a bundle during transit from its source, Alice, to its
destination, Bob. A MITM node, Mallory, is a non-cooperative node
operating on the DTN between Alice and Bob that has the ability to
receive bundles, examine bundles, modify bundles, forward bundles,
and generate bundles at will in order to compromise the
confidentiality or integrity of data within the DTN. For the
purposes of this section, any MITM node is assumed to effectively be
security-aware even if it does not implement the BPSec protocol.
There are three classes of MITM nodes which are differentiated based
on their access to cryptographic material:

o Unprivileged Node: Mallory has not been provisioned within the
   secure environment and only has access to cryptographic material
   which has been publicly-shared.

o Legitimate Node: Mallory is within the secure environment and
   therefore has access to cryptographic material which has been
   provisioned to Mallory (i.e., K_M) as well as material which has
   been publicly-shared.

o Privileged Node: Mallory is a privileged node within the secure
   environment and therefore has access to cryptographic material
   which has been provisioned to Mallory, Alice and/or Bob (i.e.
   K_M, K_A, and/or K_B) as well as material which has been publicly-
   shared.

If Mallory is operating as a privileged node, this is tantamount to
compromise; BPSec does not provide mechanisms to detect or remove
Mallory from the DTN or BPSec secure environment. It is up to the
BPSec implementer or the underlying cryptographic mechanisms to
provide appropriate capabilities if they are needed. It should also
be noted that if the implementation of BPSec uses a single set of
shared cryptographic material for all nodes, a legitimate node is
equivalent to a privileged node because K_M == K_A == K_B.

A special case of the legitimate node is when Mallory is either Alice
or Bob (i.e., K_M == K_A or K_M == K_B). In this case, Mallory is
able to impersonate traffic as either Alice or Bob, which means that
traffic to and from that node can be decrypted and encrypted,
respectively. Additionally, messages may be signed as originating
from one of the endpoints.

8.2. Attacker Behaviors and BPSec Mitigations

8.2.1. Eavesdropping Attacks

   Once Mallory has received a bundle, she is able to examine the
   contents of that bundle and attempt to recover any protected data or
   cryptographic keying material from the blocks contained within. The
   protection mechanism that BPSec provides against this action is the
   BCB, which encrypts the contents of its security target, providing
   confidentiality of the data. Of course, it should be assumed that
   Mallory is able to attempt offline recovery of encrypted data, so the
   cryptographic mechanisms selected to protect the data should provide
   a suitable level of protection.

   When evaluating the risk of eavesdropping attacks, it is important to
   consider the lifetime of bundles on a DTN. Depending on the network,
   bundles may persist for days or even years. If a bundle does persist
   on the network for years and the cipher suite used for a BCB provides
   inadequate protection, Mallory may be able to recover the protected
   data before that bundle reaches its intended destination.

8.2.2. Modification Attacks

   As a node participating in the DTN between Alice and Bob, Mallory
   will also be able to modify the received bundle, including non-BPSec
   data such as the primary block, payload blocks, or block processing
   control flags as defined in [BPBIS]. Mallory will be able to
   undertake activities which include modification of data within the
   blocks, replacement of blocks, addition of blocks, or removal of
   blocks. Within BPSec, both the BIB and BCB provide integrity
   protection mechanisms to detect or prevent data manipulation attempts
   by Mallory.

   The BIB provides that protection to another block which is its
   security target. The cryptographic mechansims used to generate the
   BIB should be strong against collision attacks and Mallory should not
   have access to the cryptographic material used by the originating
   node to generate the BIB (e.g., K_A). If both of these conditions
   are true, Mallory will be unable to modify the security target or the
   BIB and lead Bob to validate the security target as originating from
   Alice.

   Since BPSec security operations are implemented by placing blocks in
   a bundle, there is no in-band mechanism for detecting or correcting
   certain cases where Mallory removes blocks from a bundle. If Mallory
   removes a BCB block, but keeps the security target, the security
   target remains encrypted and there is a possibility that there may no
   longer be sufficient information to decrypt the block at its

destination. If Mallory removes both a BCB (or BIB) and its security
target there is no evidence left in the bundle of the security
operation. Similarly, if Mallory removes the BIB but not the
security target there is no evidence left in the bundle of the
security operation. In each of these cases, the implementation of
BPSec MUST be combined with policy configuration at endpoints in the
network which describe the expected and required security operations
that must be applied on transmission and are expected to be present
on receipt. This or other similar out-of-band information is
required to correct for removal of security information in the
bundle.

A limitation of the BIB may exist within the implementation of BIB
validation at the destination node. If Mallory is a legitimate node
within the DTN, the BIB generated by Alice with K_A can be replaced
with a new BIB generated with K_M and forwarded to Bob. If Bob is
only validating that the BIB was generated by a legitimate user, Bob
will acknowledge the message as originating from Mallory instead of
Alice. In order to provide verifiable integrity checks, both a BIB
and BCB should be used. Alice creates a BIB with the protected data
block as the security target and then creates a BCB with both the BIB
and protected data block as its security targets. In this
configuration, since Mallory is only a legitimate node and does not
have access to Alice's key K_A, Mallory is unable to decrypt the BCB
and replace the BIB.

8.2.3. Topology Attacks

If Mallory is in a MITM position within the DTN, she is able to
influence how any bundles that come to her may pass through the
network. Upon receiving and processing a bundle that must be routed
elsewhere in the network, Mallory has three options as to how to
proceed: not forward the bundle, forward the bundle as intended, or
forward the bundle to one or more specific nodes within the network.

Attacks that involve re-routing the packets throughout the network
are essentially a special case of the modification attacks described
in this section where the attacker is modifying fields within the
primary block of the bundle. Given that BPSec cannot encrypt the
contents of the primary block, alternate methods must be used to
prevent this situation. These methods MAY include requiring BIBs for
primary blocks, using encapsulation, or otherwise strategically
manipulating primary block data. The specifics of any such
mitigation technique are specific to the implementation of the
deploying network and outside of the scope of this document.

Furthermore, routing rules and policies may be useful in enforcing
particular traffic flows to prevent topology attacks. While these

rules and policies may utilize some features provided by BPSec, their
definition is beyond the scope of this specification.

8.2.4. Message Injection

Mallory is also able to generate new bundles and transmit them into
the DTN at will. These bundles may either be copies or slight
modifications of previously-observed bundles (i.e., a replay attack)
or entirely new bundles generated based on the Bundle Protocol,
BPSec, or other bundle-related protocols. With these attacks
Mallory's objectives may vary, but may be targeting either the bundle
protocol or application-layer protocols conveyed by the bundle
protocol.

BPSec relies on cipher suite capabilities to prevent replay or forged
message attacks. A BCB used with appropriate cryptographic
mechanisms (e.g., a counter-based cipher mode) may provide replay
protection under certain circumstances. Alternatively, application
data itself may be augmented to include mechanisms to assert data
uniqueness and then protected with a BIB, a BCB, or both along with
other block data. In such a case, the receiving node would be able
to validate the uniqueness of the data.

9. Ciphersuite Authorship Considerations

Cipher suite developers or implementers should consider the diverse
performance and conditions of networks on which the Bundle Protocol
(and therefore BPSec) will operate. Specifically, the delay and
capacity of delay-tolerant networks can vary substantially. Cipher
suite developers should consider these conditions to better describe
the conditions when those suites will operate or exhibit
vulnerability, and selection of these suites for implementation
should be made with consideration to the reality. There are key
differences that may limit the opportunity to leverage existing
cipher suites and technologies that have been developed for use in
traditional, more reliable networks:

o Data Lifetime: Depending on the application environment, bundles
   may persist on the network for extended periods of time, perhaps
   even years. Cryptographic algorithms should be selected to ensure
   protection of data against attacks for a length of time reasonable
   for the application.

o One-Way Traffic: Depending on the application environment, it is
   possible that only a one-way connection may exist between two
   endpoints, or if a two-way connection does exist, the round-trip
   time may be extremely large. This may limit the utility of

session key generation mechanisms, such as Diffie-Hellman, as a
two-way handshake may not be feasible or reliable.

o Opportunistic Access: Depending on the application environment, a
   given endpoint may not be guaranteed to be accessible within a
   certain amount of time. This may make asymmetric cryptographic
   architectures which rely on a key distribution center or other
   trust center impractical under certain conditions.

10. Defining Other Security Blocks

   Other security blocks (OSBs) may be defined and used in addition to
   the security blocks identified in this specification. Both the usage
   of BIB, BCB, and any future OSBs MAY co-exist within a bundle and MAY
   be considered in conformance with BPSec if each of the following
   requirements are met by any future identified security blocks.

   o Other security blocks (OSBs) MUST NOT reuse any enumerations
      identified in this specification, to include the block type codes
      for BIB and BCB.

   o An OSB definition MUST state whether it can be the target of a BIB
      or a BCB. The definition MUST also state whether the OSB can
      target a BIB or a BCB.

   o An OSB definition MUST provide a deterinistic processing order in
      the event that a bundle is received containing BIBs, BCBs, and
      OSBs. This processing order MUST NOT alter the BIB and BCB
      processing orders identified in this specification.

   o An OSB definition MUST provide a canonicalization algorithm if the
      default non-primary-block canonicalization algorithm cannot be
      used to generate a deterministic input for a cipher suite. This
      requirement MAY be waived if the OSB is defined so as to never be
      the security target of a BIB or a BCB.

   o An OSB definition MAY NOT require any behavior of a BPSEC-BPA that
      is in conflict with the behavior identified in this specification.
      In particular, the security processing requirements imposed by
      this specification MUST be consistent across all BPSEC-BPAs in a
      network.

   o The behavior of an OSB when dealing with fragmentation MUST be
      specified and MUST NOT lead to ambiguous processing states. In
      particular, an OSB definition should address how to receive and
      process an OSB in a bundle fragment that may or may not also
      contain its security target. An OSB definition should also

address whether an OSB may be added to a bundle marked as a fragment.

Additionally, policy considerations for the management, monitoring, and configuration associated with blocks SHOULD be included in any OSB definition.

NOTE: The burden of showing compliance with processing rules is placed upon the standards defining new security blocks and the identification of such blocks shall not, alone, require maintenance of this specification.

11. Conformance

All implementations are strongly RECOMMENDED to provide some method of hop-by-hop verification by generating a hash to some canonical form of the bundle and placing an integrity signature on that form using a BIB.

12. IANA Considerations

This protocol has fields that have been registered by IANA.

12.1. Bundle Block Types

This specification allocates three block types from the existing "Bundle Block Types" registry defined in [RFC6255] .

Additional Entries for the Bundle Block-Type Codes Registry:

Value	Description	Reference
2	Block Integrity Block	This document
3	Block Confidentiality Block	This document

Table 2

12.2. Cipher Suite Flags

This protocol has a cipher suite flags field and certain flags are defined. An IANA registry has been set up as follows.

The registration policy for this registry is: Specification Required

The Value range is: Variable Length

Cipher Suite Flag Registry:

+------------------------+-----------------------+-------------+
| Bit Position (right to | Description | Reference |
| left) | | |
+------------------------+-----------------------+-------------+
0	Block contains result	This
		document
1	Block Contains	This
	parameters	document
2	Source EID ref present	This
		document
>3	Reserved	This
		document
+------------------------+-----------------------+-------------+

Table 3

12.3. Parameters and Results

   This protocol has fields for cipher suite parameters and results.
   The field is a type-length-value triple and a registry is required
   for the "type" sub-field. The values for "type" apply to both the
   cipher suite parameters and the cipher suite results fields. Certain
   values are defined. An IANA registry has been set up as follows.

   The registration policy for this registry is: Specification Required

   The Value range is: 8-bit unsigned integer.

Cipher Suite Parameters and Results Type Registry:

```
+---------+-------------------------------------------+-------------+
|  Value  |                Description                |  Reference  |
+---------+-------------------------------------------+-------------+
|    0    |                 reserved                  | Section 3.6 |
|    1    |         initialization vector (IV)        | Section 3.6 |
|    2    |                 reserved                  | Section 3.6 |
|    3    |              key information              | Section 3.6 |
|    4    | content-range (pair of Unsigned Integers) | Section 3.6 |
|    5    |            integrity signature            | Section 3.6 |
|    6    |                unassigned                 | Section 3.6 |
|    7    |                   salt                    | Section 3.6 |
|    8    |        BCB integrity check value (ICV)    | Section 3.6 |
|  9-191  |                 reserved                  | Section 3.6 |
| 192-250 |                private use                | Section 3.6 |
| 251-255 |                 reserved                  | Section 3.6 |
+---------+-------------------------------------------+-------------+
```

Table 4

13. References

13.1. Normative References

   [BPBIS]   Burleigh, S., Fall, K., and E. Birrane, "Bundle Protocol",
              draft-ietf-dtn-bpbis-04 (work in progress), July 2016.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC
              Text on Security Considerations", BCP 72, RFC 3552,
              DOI 10.17487/RFC3552, July 2003,
              <http://www.rfc-editor.org/info/rfc3552>.

   [RFC6255] Blanchet, M., "Delay-Tolerant Networking Bundle Protocol
              IANA Registries", RFC 6255, May 2011.

13.2. Informative References

   [BPBISCBOR]
              Burleigh, S., "Bundle Protocol CBOR Representation
              Specification", draft-burleigh-dtn-rs-cbor-01 (work in
              progress), April 2016.

   [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst,
              R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant
              Networking Architecture", RFC 4838, April 2007.

   [RFC6257] Symington, S., Farrell, S., Weiss, H., and P. Lovell,
              "Bundle Security Protocol Specification", RFC 6257, May
              2011.

   [SBSP]   Birrane, E., "Streamlined Bundle Security Protocol",
              draft-birrane-dtn-sbsp-01 (work in progress), October
              2015.

Appendix A. Acknowledgements

   The following participants contributed technical material, use cases,
   and useful thoughts on the overall approach to this security
   specification: Scott Burleigh of the Jet Propulsion Laboratory, Amy
   Alford and Angela Hennessy of the Laboratory for Telecommunications
   Sciences, and Angela Dalton and Cherita Corbett of the Johns Hopkins
   University Applied Physics Laboratory.

Authors' Addresses

   Edward J. Birrane, III
   The Johns Hopkins University Applied Physics Laboratory
   11100 Johns Hopkins Rd.
   Laurel, MD 20723
   US

   Phone: +1 443 778 7423
   Email: Edward.Birrane@jhuapl.edu

   Kenneth McKeever
   The Johns Hopkins University Applied Physics Laboratory
   11100 Johns Hopkins Rd.
   Laurel, MD 20723
   US

   Phone: +1 443 778 2237
   Email: Ken.McKeever@jhuapl.edu