Bundle Protocol Security Specification
draft-ietf-dtn-bpsec-17

Abstract

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

   This document is an update of the protocol described in RFC 6257,
   reflecting lessons learned. For this reason it obsoletes RFC 6257,
   an IRTF-stream document.

Status of This Memo

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

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

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

   This Internet-Draft will expire on July 25, 2020.

Copyright Notice

Table of Contents

1. Introduction

   This document defines security features for the Bundle Protocol (BP)
   [I-D.ietf-dtn-bpbis] and is intended for use in Delay Tolerant
   Networks (DTNs) to provide end-to-end security services.

   The Bundle Protocol specification [I-D.ietf-dtn-bpbis] defines DTN as
   referring to "a networking architecture providing communications in
   and/or through highly stressed environments" where "BP may be viewed
   as sitting at the application layer of some number of constituent
   networks, forming a store-carry-forward overlay network". The term
   "stressed" environment refers to multiple challenging conditions
   including intermittent connectivity, large and/or variable delays,
   asymmetric data rates, and high bit error rates.

   The BP might be deployed such that portions of the network cannot be
   trusted, posing the usual security challenges related to
   confidentiality and integrity. However, the stressed nature of the
   BP operating environment imposes unique conditions where usual
   transport security mechanisms may not be sufficient. For example,
   the store-carry-forward nature of the network may require protecting
   data at rest, preventing unauthorized consumption of critical
   resources such as storage space, and operating without regular
   contact with a centralized security oracle (such as a certificate
   authority).

   An end-to-end security service is needed that operates in all of the
   environments where the BP operates.

   This document is an update of the protocol described in RFC 6257,
   reflecting lessons learned. For this reason it obsoletes RFC 6257,
   an IRTF-stream document.

   Note to the RFC editor: Please delete below paragraph prior to RFC
   publication.

   The Internet Research Task Force will be requested to mark RFC6257 as
   obsolete.

1.1. Supported Security Services

   BPSec provides end-to-end integrity and confidentiality services for
   BP bundles, as defined in this section.

   Integrity services ensure that changes to target data within a bundle
   can be discovered. Data changes may be caused by processing errors,
   environmental conditions, or intentional manipulation. In the
   context of BPSec, integrity services apply to plain-text in the
   bundle.

   Confidentiality services ensure that target data is unintelligible to
   nodes in the DTN, except for authorized nodes possessing special
   information. This generally means producing cipher-text from plain-
   text and generating authentication information for that cipher-text.
   Confidentiality, in this context, applies to the contents of target
   data and does not extend to hiding the fact that confidentiality
   exists in the 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
       detect and therefore hop-by-hop authentication is difficult or
       impossible to enforce.

   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 is a special case of data integrity and
       can be achieved with the integrity mechanisms defined in this
       specification. Therefore, a separate authentication service is
       not necessary.

1.2. Specification Scope

   This document defines the security services provided by the BPSec.
   This includes the data specification for representing these services
   as BP extension blocks, and the rules for adding, removing, and
   processing these blocks at various points during the bundle's
   traversal of the DTN.

   BPSec applies only to those nodes that implement it, known as
   "security-aware" nodes. There might be other nodes in the DTN that

do not implement BPSec. While all nodes in a BP overlay can exchange bundles, BPSec security operations can only happen at BPSec security-aware nodes.

BPSec addresses only the security of data traveling over the DTN, not the underlying DTN itself. Furthermore, while the BPSec protocol can provide security-at-rest in a store-carry-forward network, it 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.

This specification addresses neither the fitness of externally-defined cryptographic methods nor the security of their implementation. Different networking conditions and operational considerations require varying strengths of security mechanism such that mandating a cipher suite in this specification may result in too much security for some networks and too little security in others. It is expected that separate documents will be standardized to define security contexts and cipher suites compatible with BPSec, to include those that should be used to assess interoperability and those fit for operational use in various network scenarios. A sample security context has been defined ([I-D.ietf-dtn-bpsec-interop-sc]) to support interoperability testing and serve as an exemplar for how security contexts should be defined for this specification.

This specification does not address the implementation of security policy and does not provide a security policy for the BPSec. Similar to cipher suites, security policies are based on the nature and capabilities of individual networks and network operational concepts. This specification does provide policy considerations when building a security policy.

With the exception of the Bundle Protocol, 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.

1.3. 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 DTNs and identifies certain security assumptions made by existing Internet protocols that are not valid in a DTN.

The Bundle Protocol [I-D.ietf-dtn-bpbis] defines the format and processing of bundles, defines the extension block format used to represent BPSec security blocks, and defines the canonicalization algorithms used by this specification.

The Concise Binary Object Representation (CBOR) format [RFC7049] defines a data format that allows for small code size, fairly small message size, and extensibility without version negotiation. The block-specific data associated with BPSec security blocks are encoded in this data format.

The Bundle Security Protocol [RFC6257] and Streamlined Bundle Security Protocol [I-D.birrane-dtn-sbsp] documents introduced the concepts of using BP extension blocks for security services in a DTN. The BPSec is a continuation and refinement of these documents.

1.4. 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 terminology either unique to the BPSec or otherwise necessary for understanding the concepts defined in this specification.

o   Bundle Destination - the node which receives a bundle and delivers the payload of the bundle to an application. Also, the Node ID of the Bundle Protocol Agent (BPA) receiving the bundle. The bundle destination acts as the security acceptor for every security target in every security block in every bundle it receives.

o   Bundle Source - the node which originates a bundle. Also, the Node ID of the BPA originating the bundle.

o   Cipher Suite - a set of one or more algorithms providing integrity and confidentiality services. Cipher suites may define necessary parameters but do not provide values for those parameters.

o   Forwarder - any node that transmits a bundle in the DTN. Also, the Node ID of the BPA that sent the bundle on its most recent hop.

o   Intermediate Receiver, Waypoint, or Next Hop - any node that
    receives a bundle from a Forwarder that is not the Bundle
    Destination. Also, the Node ID of the BPA at any such node.

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 in advance by the bundle or any BPAs in the DTN.

o   Security Acceptor - a bundle node that processes and dispositions
    one or more security blocks in a bundle. Also, the Node ID of
    that node.

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

o   Security Context - the set of assumptions, algorithms,
    configurations and policies used to implement security services.

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.

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

o   Security Source - a bundle node that adds a security block to a
    bundle. Also, the Node ID of that node.

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

2. Design Decisions

   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 application security requirements. This section
   identifies those desirable properties that guide design decisions for
   this specification and are necessary for understanding the format and
   behavior of the BPSec protocol.

2.1. Block-Level Granularity

   Security services within this specification must allow different
   blocks within a bundle to have different security services applied to
   them.

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 represent different types of information with different
security needs.

For example, a payload block might be encrypted to protect its
contents and an extension block containing summary information
related to the payload might be integrity signed but unencrypted to
provide waypoints access to payload-related data without providing
access to the payload.

2.2. Multiple Security Sources

A bundle can have multiple security blocks and these blocks can have
different security sources. BPSec implementations MUST NOT assume
that all blocks in a bundle have the same security operations and/or
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 adds a
new extension block to a bundle, that extension block MAY have
security services applied to it by that waypoint. Similarly, a
waypoint MAY add a security service to an existing extension block,
consistent with its security policy.

When a waypoint adds a security service to the bundle, the waypoint
is the security source for that service. The security block(s) which
represent that service in the bundle may need to record this security
source as the bundle destination might need this information for
processing.

For example, a bundle source may choose to apply an integrity service
to its plain-text payload. Later a waypoint node, representing a
gateway to an insecure portion of the DTN, may receive the bundle and
choose to apply a confidentiality service. In this case, the
integrity security source is the bundle source and the
confidentiality security source is the waypoint node.

2.3. Mixed Security Policy

The security policy enforced by nodes in the DTN may differ.

Some waypoints might not be security aware and will not be able to
process security blocks. Therefore, security blocks must have their
processing flags set such that the block will be treated
appropriately by non-security-aware waypoints.

Some waypoints will have security policies that require evaluating
security services even if they are not the bundle destination or the
final intended acceptor of the service. For example, a waypoint
could choose to verify an integrity service even though the waypoint
is not the bundle destination and the integrity service will be
needed by other nodes along the bundle's path.

Some waypoints will determine, through policy, that they are the
intended recipient of the security service and terminate the security
service in the bundle. For example, a gateway node could determine
that, even though it is not the destination of the bundle, it should
verify and remove a particular integrity service or attempt to
decrypt a confidentiality service, before forwarding the bundle along
its path.

Some waypoints could understand security blocks but refuse to process
them unless they are the bundle destination.

2.4. User-Defined Security Contexts

A security context is the union of security algorithms (cipher
suites), policies associated with the use of those algorithms, and
configuration values. Different contexts may specify different
algorithms, different polices, or different configuration values used
in the implementation of their security services. BPSec must provide
a mechanism for users to define their own security contexts.

For example, some users might prefer a SHA2 hash function for
integrity whereas other users might prefer a SHA3 hash function. The
security services defined in this specification must provide a
mechanism for determining what cipher suite, policy, and
configuration has been used to populate a security block.

2.5. Deterministic Processing

Whenever a node determines that it must process more than one
security block in a received bundle (either because the policy at a
waypoint states that it should process security blocks or because the
node is the bundle destination) the order in which security blocks
are processed must be deterministic. All nodes must impose this same
deterministic processing order for all security blocks. This
specification provides determinism in the application and evaluation

of security services, even when doing so results in a loss of
flexibility.

3. Security Blocks

3.1. Block Definitions

This specification defines two types of security block: the Block
Integrity Block (BIB) and the Block Confidentiality Block (BCB).

The BIB is used to ensure the integrity of its plain-text security
target(s). The integrity information in the BIB MAY be verified
by any node along the bundle path from the BIB security source to
the bundle destination. Security-aware waypoints add or remove
BIBs from bundles in accordance with their security policy. BIBs
are never used to sign the cipher-text provided by a BCB.

The BCB indicates that the security target(s) have been encrypted
at the BCB security source in order to protect their content while
in transit. The BCB is decrypted by security-aware nodes in the
network, up to and including the bundle destination, as a matter
of security policy. BCBs additionally provide authentication
mechanisms for the cipher-text they generate.

3.2. Uniqueness

Security operations in a bundle MUST be unique; the same security
service MUST NOT be applied to a security target more than once in a
bundle. Since a security operation is represented as a security
block, this limits what security blocks may be added to a bundle: if
adding a security block to a bundle would cause some other security
block to no longer represent a unique security operation then the new
block MUST NOT be added. It is important to note that any cipher-
text integrity mechanism supplied by the BCB is considered part of
the confidentiality service and, therefore, unique from the plain-
text integrity service provided by the BIB.

If multiple security blocks representing the same security operation
were allowed in a bundle at the same time, there would exist
ambiguity regarding block processing order and the property of
deterministic processing of blocks would be lost.

Using the notation OP(service, target), several examples illustrate
this uniqueness requirement.

o   Signing the payload twice: The two operations OP(integrity,
    payload) and OP(integrity, payload) are redundant and MUST NOT
    both be present in the same bundle at the same time.

o Signing different blocks: The two operations OP(integrity,
   payload) and OP(integrity, extension_block_1) are not redundant
   and both may be present in the same bundle at the same time.
   Similarly, the two operations OP(integrity, extension_block_1) and
   OP(integrity,extension_block_2) are also not redundant and may
   both be present in the bundle at the same time.

o Different Services on same block: The two operations OP(integrity,
   payload) and OP(confidentiality, payload) are not inherently
   redundant and may both be present in the bundle at the same time,
   pursuant to other processing rules in this specification.

3.3. Target Multiplicity

   Under special circumstances, a single security block MAY represent
   multiple security operations as a way of reducing the overall number
   of security blocks present in a bundle. In these circumstances,
   reducing the number of security blocks in the bundle reduces the
   amount of redundant information in the bundle.

   A set of security operations can be represented by a single security
   block when all of the following conditions are true.

   o The security operations apply the same security service. For
      example, they are all integrity operations or all confidentiality
      operations.

   o The security context parameters for the security operations are
      identical.

   o The security source for the security operations is the same.
      Meaning the set of operations are being added by the same node.

   o No security operations have the same security target, as that
      would violate the need for security operations to be unique.

   o None of the security operations conflict with security operations
      already present in the bundle.

   When representing multiple security operations in a single security
   block, the information that is common across all operations is
   represented once in the security block, and the information which is
   different (e.g., the security targets) are represented individually.

   It is RECOMMENDED that if a node processes any security operation in
   a security block that it process all security operations in the
   security block. This allows security sources to assert that the set
   of security operations in a security block are expected to be

processed by the same security acceptor. However, the determination
of whether a node actually is a security acceptor or not is a matter
of the policy of the node itself. In cases where a receiving node
determines that it is the security acceptor of only a subset of the
security operations in a security block, the node may choose to only
process that subset of security operations.

3.4. Target Identification

A security target is a block in the bundle to which a security
service applies. This target must be uniquely and unambiguously
identifiable when processing a security block. The definition of the
extension block header from [I-D.ietf-dtn-bpbis] provides a "Block
Number" field suitable for this purpose. Therefore, a security
target in a security block MUST be represented as the Block Number of
the target block.

3.5. Block Representation

Each security block uses the Canonical Bundle Block Format as defined
in [I-D.ietf-dtn-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 (if present)

o Block Data Length

o Block Type Specific Data Fields

Security-specific information for a security block is captured in the
"Block Type Specific Data Fields".

3.6. Abstract Security Block

The structure of the security-specific portions of a security block
is identical for both the BIB and BCB Block Types. Therefore, this
section defines an Abstract Security Block (ASB) data structure and
discusses the definition, processing, and other constraints for using
this structure. An ASB is never directly instantiated within a
bundle, it is only a mechanism for discussing the common aspects of
BIB and BCB security blocks.

The fields of the ASB SHALL be as follows, listed in the order in
which they must appear.

Security Targets:
    This field identifies the block(s) targeted by the security
    operation(s) represented by this security block. Each target
    block is represented by its unique Block Number. This field
    SHALL be represented by a CBOR array of data items. Each
    target within this CBOR array SHALL be represented by a CBOR
    unsigned integer. This array MUST have at least 1 entry and
    each entry MUST represent the Block Number of a block that
    exists in the bundle. There MUST NOT be duplicate entries in
    this array.

Security Context Id:
    This field identifies the security context used to implement
    the security service represented by this block and applied to
    each security target. This field SHALL be represented by a
    CBOR unsigned integer.

Security Context Flags:
    This field identifies which optional fields are present in the
    security block. This field SHALL be represented as a CBOR
    unsigned integer whose contents shall be interpreted as a bit
    field. Each bit in this bit field indicates the presence (bit
    set to 1) or absence (bit set to 0) of optional data in the
    security block. The association of bits to security block data
    is defined as follows.

    Bit 1 (the least-significant bit, 0x01): Security Context
           Parameters Present Flag.

    Bit 2 (0x02): Security Source Present Flag.

    Bit >2 Reserved

    Implementations MUST set reserved bits to 0 when writing this
    field and MUST ignore the values of reserved bits when reading
    this field. For unreserved bits, a value of 1 indicates that
    the associated security block field MUST be included in the
    security block. A value of 0 indicates that the associated
    security block field MUST NOT be in the security block.

Security Source (Optional):
    This field identifies the Endpoint that inserted the security
    block in the bundle. If the security source field is not
    present then the source MUST be inferred from other
    information, such as the bundle source, previous hop, or other

values defined by security policy. This field SHALL be
represented by a CBOR array in accordance with
[I-D.ietf-dtn-bpbis] rules for representing Endpoint
Identifiers (EIDs).

Security Context Parameters (Optional):
    This field captures one or more security context parameters
    that should be provided to security-aware nodes when processing
    the security service described by this security block. This
    field SHALL be represented by a CBOR array. Each entry in this
    array is a single security context parameter. A single
    parameter SHALL also be represented as a CBOR array comprising
    a 2-tuple of the id and value of the parameter, as follows.

    * Parameter Id. This field identifies which parameter is
       being specified. This field SHALL be represented as a CBOR
       unsigned integer. Parameter Ids are selected as described
       in Section 3.10.

    * Parameter Value. This field captures the value associated
       with this parameter. This field SHALL be represented by the
       applicable CBOR representation of the parameter, in
       accordance with Section 3.10.

    The logical layout of the parameters array is illustrated in
    Figure 1.

```
+----------------+----------------+     +----------------+
|  Parameter 1   |  Parameter 2   | ... |  Parameter N   |
+------+---------+------+---------+     +------+---------+
| Id   | Value   | Id   | Value   |     | Id   | Value   |
+------+---------+------+---------+     +------+---------+
```

                Figure 1: Security Context Parameters

Security Results:
    This field captures the results of applying a security service
    to the security targets of the security block. This field
    SHALL be represented as a CBOR array of target results. Each
    entry in this array represents the set of security results for
    a specific security target. The target results MUST be ordered
    identically to the Security Targets field of the security
    block. This means that the first set of target results in this
    array corresponds to the first entry in the Security Targets
    field of the security block, and so on. There MUST be one
    entry in this array for each entry in the Security Targets
    field of the security block.

The set of security results for a target is also represented as a CBOR array of individual results. An individual result is represented as a 2-tuple of a result id and a result value, defined as follows.

*   Result Id. This field identifies which security result is being specified. Some security results capture the primary output of a cipher suite. Other security results contain additional annotative information from cipher suite processing. This field SHALL be represented as a CBOR unsigned integer. Security result Ids will be as specified in Section 3.10.

*   Result Value. This field captures the value associated with the result. This field SHALL be represented by the applicable CBOR representation of the result value, in accordance with Section 3.10.

The logical layout of the security results array is illustrated in Figure 2. In this figure there are N security targets for this security block. The first security target contains M results and the Nth security target contains K results.

```
+-----------------------------+       +-----------------------------+
|           Target 1          |       |           Target N          |
+-----------+----+------------+       +-----------------------------+
| Result 1  |    | Result M   | ...   | Result 1  |    | Result K   |
+----+------+ .. +----+-------+       +----+------+ .. +----+-------+
| Id | Value |   | Id | Value |       | Id | Value |   | Id | Value |
+----+------+    +----+-------+       +----+------+    +----+-------+
```
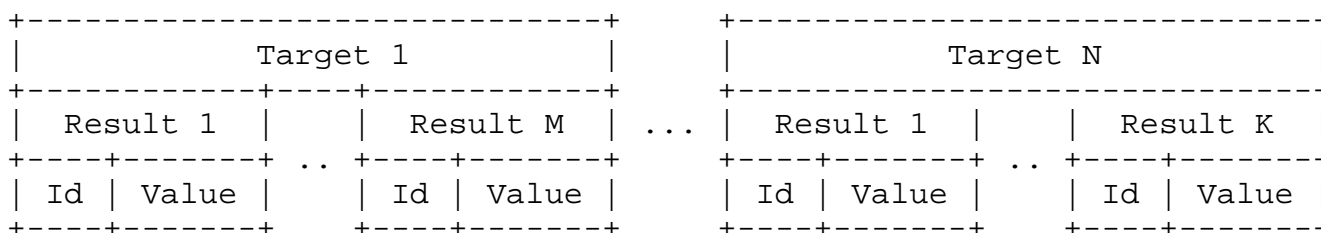
Figure 2: Security Results

3.7. Block Integrity Block

A BIB is a bundle extension block with the following characteristics.

o   The Block Type Code value is as specified in Section 11.1.

o   The Block Type Specific Data Fields follow the structure of the ASB.

o   A security target listed in the Security Targets field MUST NOT reference a security block defined in this specification (e.g., a BIB or a BCB).

o   The Security Context Id MUST utilize an end-to-end authentication cipher or an end-to-end error detection cipher.

o The EID of 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 be specified as part of
   security context information described in Section 3.10.

Notes:

o It is RECOMMENDED that designers 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.

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
   security context.

o For some security contexts, (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 bundle destination that has access to the required
   keying information, in accordance with Section 3.9.

3.8. Block Confidentiality Block

   A BCB is a bundle extension block with the following characteristics.

   The Block Type Code value is as specified in Section 11.1.

   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.

   The Block Type Specific Data Fields follow the structure of the
   ASB.

   A security target listed in the Security Targets field can
   reference the payload block, a non-security extension block, or a
   BIB. A BCB MUST NOT include another BCB as a security target. A
   BCB MUST NOT target the primary block.

   The Security Context Id MUST utilize a confidentiality cipher that
   provides authenticated encryption with associated data (AEAD).

Additional information created by a cipher suite (such as
additional authenticated data) can be placed either in a security
result field or in the generated cipher-text. The determination
of where to place these data is a function of the cipher suite and
security context used.

The EID of 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 be specified as part of
security context information described in Section 3.10.

The BCB modifies the contents of its security target(s). When a BCB
is applied, the security target body data are encrypted "in-place".
Following encryption, the security target Block Type Specific Data
field contains cipher-text, not plain-text. Other block fields
remain unmodified, with the exception of the Block Data Length field,
which MUST be updated to reflect the new length of the Block Type
Specific Data field.

Notes:

o It is RECOMMENDED that designers 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.

o The BCB block processing control flags can be set independently
   from the processing control flags of the security target(s). The
   setting of such flags SHOULD be an implementation/policy decision
   for the encrypting node.

3.9. 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 a security target of a BCB is also a security target of a BIB, an
undesirable condition occurs where a security aware waypoint would be
unable to validate the BIB because one of its security target's
contents have been encrypted by a BCB. To address this situation the
following processing rules MUST be followed.

o When adding a BCB to a bundle, if some (or all) of the security
   targets of the BCB also match all of the security targets of an
   existing BIB, then the existing BIB MUST also be encrypted. This
   can be accomplished by either adding a new BCB that targets the

existing BIB, or by adding the BIB to the list of security targets for the BCB. Deciding which way to represent this situation is a matter of security policy.

o When adding a BCB to a bundle, if some (or all) of the security targets of the BCB match some (but not all) of the security targets of a BIB then that BIB MUST be altered in the following way. Any security results in the BIB associated with the BCB security targets MUST be removed from the BIB and placed in a new BIB. This newly created BIB MUST then be encrypted. The encryption of the new BIB can be accomplished by either adding a new BCB that targets the new BIB, or by adding the new BIB to the list of security targets for the BCB. Deciding which way to represent this situation is a matter of security policy.

o A BIB MUST NOT be added for a security target that is already the security target of a BCB. In this instance, the BCB is already providing authentication and integrity of the security target and the BIB would be redundant, insecure, and cause ambiguity in block processing order.

o A BIB integrity value MUST NOT be evaluated if the BIB is the security target of an existing BCB. In this case, the BIB data is encrypted.

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

o As mentioned in Section 3.7, a BIB MUST NOT have a BCB as its security target.

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.

Since any cipher suite used with a BCB MUST be an AEAD cipher suite, it is inefficient and insecure for a single security source to add both a BIB and a BCB for the same security target. In cases where a security source wishes to calculate both a plain-text integrity mechanism and encrypt a security target, a BCB with a security context that generates such signatures as additional security results MUST be used instead.

3.10. Parameter and Result Identification

   Each security context MUST define its own context parameters and
   results. Each defined parameter and result is represented as the
   tuple of an identifier and a value. Identifiers are always
   represented as a CBOR unsigned integer. The CBOR encoding of values
   is as defined by the security context specification.

   Identifiers MUST be unique for a given security context but do not
   need to be unique amongst all security contexts.

3.11. BSP Block Examples

   This section provides two examples of BPSec blocks applied to a
   bundle. In the first example, a single node adds several security
   operations to a bundle. In the second example, a waypoint node
   received the bundle created in the first example and adds additional
   security operations. In both examples, the first column represents
   blocks within a bundle and the second column represents the Block
   Number for the block, using the terminology B1...Bn for the purpose
   of illustration.

3.11.1. Example 1: Constructing a Bundle with Security

   In this example a bundle has four non-security-related blocks: the
   primary block (B1), two extension blocks (B4,B5), and a payload block
   (B6). The bundle source wishes to provide an integrity signature of
   the plain-text associated with the primary block, the second
   extension block, and the payload. The bundle source also wishes to
   provide confidentiality for the first extension block. The resultant
   bundle is illustrated in Figure 3 and the security actions are
   described below.

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

                 Figure 3: Security at Bundle Creation

   The following security actions were applied to this bundle at its
   time of creation.

   o An integrity signature applied to the canonicalized primary block
      (B1), the second extension block (B5) and the payload block (B6).
      This is accomplished by a single BIB (B2) with multiple targets.
      A single BIB is used in this case because all three targets share
      a security source, security context, and security context
      parameters. Had this not been the case, multiple BIBs could have
      been added instead.

   o Confidentiality for the first extension block (B4). This is
      accomplished by a BCB (B3). Once applied, the contents of
      extension block B4 are encrypted. The BCB MUST hold an
      authentication signature for the cipher-text either in the cipher-
      text that now populates the first extension block or as a security
      result in the BCB itself, depending on which security context is
      used to form the BCB. A plain-text integrity signature may also
      exist as a security result in the BCB if one is provided by the
      selected confidentiality security context.

3.11.2. Example 2: Adding More Security At A New Node

   Consider that the bundle as it is illustrated in Figure 3 is now
   received by a waypoint node that wishes to encrypt the second
   extension block and the bundle payload. The waypoint security policy
   is to allow existing BIBs for these blocks to persist, as they may be
   required as part of the security policy at the bundle destination.

The resultant bundle is illustrated in Figure 4 and the security
actions are described below. Note that block IDs provided here are
ordered solely for the purpose of this example and not meant to
impose an ordering for block creation. The ordering of blocks added
to a bundle MUST always be in compliance with [I-D.ietf-dtn-bpbis].

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

                Figure 4: Security At Bundle Forwarding

The following security actions were applied to this bundle prior to
its forwarding from the waypoint node.

o Since the waypoint node wishes to encrypt blocks B5 and B6, it
   MUST also encrypt the BIBs providing plain-text integrity over
   those blocks. However, BIB B2 could not be encrypted in its
   entirety because it also held a signature for the primary block
   (B1). Therefore, a new BIB (B7) is created and security results
   associated with B5 and B6 are moved out of BIB B2 and into BIB B7.

o Now that there is no longer confusion of which plain-text
   integrity signatures must be encrypted, a BCB is added to the
   bundle with the security targets being the second extension block
   (B5) and the payload (B6) as well as the newly created BIB holding
   their plain-text integrity signatures (B7). A single new BCB is
   used in this case because all three targets share a security
   source, security context, and security context parameters. Had

this not been the case, multiple BCBs could have been added
instead.

4. Canonical Forms

Security services require consistency and determinism in how
information is presented to cipher suites at the security source and
at a receiving node. For example, integrity services require that
the same target information (e.g., the same bits in the same order)
is provided to the cipher suite when generating an original signature
and when generating a comparison signature. Canonicalization
algorithms are used to construct a stable, end-to-end bit
representation of a target block.

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

The canonicalization of the primary block is as specified in
[I-D.ietf-dtn-bpbis].

All non-primary blocks share the same block structure and are
canonicalized as specified in [I-D.ietf-dtn-bpbis] with the following
exceptions.

o   If the service being applied is a confidentiality service, then
    the Block Type Code, Block Number, Block Processing Control Flags,
    CRC Type and CRC Field (if present), and Block Data Length fields
    MUST NOT be included in the canonicalization. Confidentiality
    services are used solely to convert the Block Type Specific Data
    Fields from plain-text to cipher-text.

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

These canonicalization algorithms assume that Endpoint IDs do not
change from the time at which a security source adds a security block
to a bundle and the time at which a node processes that security
block.

Cipher suites and security contexts 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, those algorithms take
precedence over this specification.

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 When BIBs and BCBs share a security target, BCBs MUST be evaluated
      first and BIBs second.

5.1.1. Receiving BCBs

   If a received bundle contains a BCB, the receiving node MUST
   determine whether it is the security acceptor for any of the security
   operations in the BCB. If so, the node MUST process those operations
   and remove any operation-specific information from the BCB prior to
   delivering data to an application at the node or forwarding the
   bundle. If processing a security operation fails, the target SHALL
   be processed according to the security policy. A bundle status
   report indicating the failure MAY be generated. When all security
   operations for a BCB have been removed from the BCB, the BCB MUST be
   removed from 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 MUST process the BCB if
   directed to do so as a matter of security policy.

   If the security policy of a security-aware node specifies that a
   bundle should have applied confidentiality to a specific security
   target and no such BCB 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 block, the bundle MUST be
   discarded.

   If an encrypted payload block cannot be decrypted (i.e., the cipher-
   text cannot be authenticated), 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 [I-D.ietf-dtn-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 Block Type Specific Data Fields. If the Block Data Length field was modified at the time of encryption it MUST be updated to reflect the decrypted block length.

If a BCB contains multiple security operations, each operation processed by the node MUST be be treated as if the security operation has been represented by a single BCB with a single security operation for the purposes of report generation and policy processing.

5.1.2. Receiving BIBs

If a received bundle contains a BIB, the receiving node MUST determine whether it is the security acceptor for any of the security operations in the BIB. If so, the node MUST process those operations and remove any operation-specific information from the BIB prior to delivering data to an application at the node or forwarding the bundle. If processing a security operation fails, the target SHALL be processed according to the security policy. A bundle status report indicating the failure MAY be generated. When all security operations for a BIB have been removed from the BIB, the BIB MUST be removed from 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 can occur at any node that has the ability to verify an integrity signature, not just the bundle destination.

If a receiving node is not the security acceptor of a security operation in a BIB it MAY attempt to verify the security operation anyway to prevent forwarding corrupt data. If the verification 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 bundle destination. If the check passes, the

node MUST NOT remove the security operation from the BIB prior to
forwarding.

If a BIB contains multiple security operations, each operation
processed by the node MUST be be treated as if the security operation
has been represented by a single BIB with a single security operation
for the purposes of report generation and policy processing.

5.2. Bundle Fragmentation and Reassembly

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

Due to the complexity of payload block fragmentation, including the
possibility of fragmenting payload block fragments, integrity and
confidentiality operations are not to be applied to a bundle
representing a fragment. Specifically, a BCB or BIB MUST NOT be
added to a bundle if the "Bundle is a Fragment" flag is set in the
Bundle Processing Control Flags field.

Security processing in the presence of payload block fragmentation
may be handled by other mechanisms outside of the BPSec protocol or
by applying BPSec blocks in coordination with an encapsulation
mechanism.

6. Key Management

There exist a myriad of ways to establish, communicate, and otherwise
manage key information in a DTN. Certain DTN deployments might
follow established protocols for key management whereas other DTN
deployments might require new and novel approaches. BPSec assumes
that key management is handled as a separate part of network
management and this specification neither defines nor requires a
specific key management strategy.

7. Security 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. No single set of policy decisions is envisioned to
work for all secure DTN deployments.

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 a waypoint 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
   blocks that absolutely need them. If a BPA were to apply security
   operations such as integrity or confidentiality to every block in
   the bundle, regardless of need, there could be downstream errors
   processing blocks whose contents must be inspected or changed at
   every hop along the path.

o It is recommended that BCBs be allowed to alter the size of
   extension blocks and the payload block. However, care must be
   taken to ensure that changing the size of the payload block while
   the bundle is in transit do not negatively affect bundle
   processing (e.g., calculating storage needs, scheduling
   transmission times, caching block byte offsets).

o Adding a BIB to a security target that has already been encrypted
   by a BCB is not allowed. If this condition is likely to be
   encountered, there are (at least) three possible policies that
   could handle this situation.

   1. At the time of encryption, a plain-text 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.

o It is recommended that security policy address whether cipher
   suites whose cipher-text is larger (or smaller) than the initial
   plain-text are permitted and, if so, for what types of blocks.
   Changing the size of a block may cause processing difficulties for
   networks that calculate block offsets into bundles or predict
   transmission times or storage availability as a function of bundle
   size. In other cases, changing the size of a payload as part of
   encryption has no significant impact.

8. Security Considerations

   Given the nature of DTN 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. This section provides additional
   discussion on security threats that BPSec will face and describes how
   BPSec security mechanisms operate to mitigate these threats.

   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. In doing so, it is assumed that the DTN (or significant
   portions of the DTN) are completely under the control of an attacker.

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. Long-lived bundles imply

that the data exists in the network for a longer period of time and, thus, there may be more opportunities to capture those bundles. Additionally, bundles that are long-lived imply that the information stored within them may remain relevant and sensitive for long enough that, once captured, there is sufficient time to crack encryption associated with the bundle. 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 either before that bundle reaches its intended destination or before the information in the bundle is no longer considered sensitive.

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 [I-D.ietf-dtn-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 mechanisms 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, 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 and the BCB should require an IND-CCA2
encryption scheme. Such an encryption scheme will guard against
signature substitution attempts by Mallory. In this case, 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.

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. Security Context Considerations

9.1. Identification and Configuration

Security blocks must uniquely define the security context for their
services. This context MUST be uniquely identifiable and MAY use
parameters for customization. Where policy and configuration
decisions can be captured as parameters, the security context
identifier may identify a cipher suite. In cases where the same
cipher suites are used with differing predetermined configurations
and policies, users can define multiple security contexts that use
the same cipher suite.

Network operators must determine the number, type, and configuration
of security contexts in a system. Networks with rapidly changing
configurations may define relatively few security contexts with each
context customized with multiple parameters. For networks with more
stability, or an increased need for confidentiality, a larger number
of contexts can be defined with each context supporting few, if any,
parameters.

Security Context Examples

+---------+-----------+--+
| Context | Parameters | Definition |
| Id | | |
+---------+-----------+--+
1	Key, IV	AES-GCM-256 cipher suite with provided
		ephemeral key and initialization vector.
2	IV	AES-GCM-256 cipher suite with
		predetermined key and predetermined key
		rotation policy.
3	Nil	AES-GCM-256 cipher suite with all info
		predetermined.
+---------+-----------+--+

Table 1

9.2. Authorship

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. Developers should
consider these conditions to better describe the conditions when
those contexts will operate or exhibit vulnerability, and selection
of these contexts for implementation should be made with
consideration for this reality. There are key differences that may
limit the opportunity for a security context 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.

When developing new security contexts for use with BPSec, the following information SHOULD be considered for inclusion in these specifications.

o Security Context Parameters. Security contexts MUST define their parameter Ids, the data types of those parameters, and their CBOR encoding.

o Security Results. Security contexts MUST define their security result Ids, the data types of those results, and their CBOR encoding.

o New Canonicalizations. Security contexts may define new canonicalization algorithms as necessary.

o Cipher-Text Size. Security contexts MUST state whether their associated cipher suites generate cipher-text (to include any authentication information) that is of a different size than the input plain-text.

   If a security context does not wish to alter the size of the plain-text, it should consider defining the following policy.

   * Place overflow bytes, authentication signatures, and any additional authenticated data in security result fields rather than in the cipher-text itself.

   * Pad the cipher-text in cases where the cipher-text is smaller than the plain-text.

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 can co-exist within a bundle and can 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 deterministic 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 can 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 MUST 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. IANA Considerations

   This specification includes fields requiring registries managed by
   IANA.

11.1. Bundle Block Types

   This specification allocates two block types from the existing
   "Bundle Block Types" registry defined in [I-D.ietf-dtn-bpbis].

Additional Entries for the Bundle Block-Type Codes Registry:

```
+-------+----------------------------+---------------+
| Value |         Description        |   Reference   |
+-------+----------------------------+---------------+
|  TBA  |    Block Integrity Block   | This document |
|  TBA  | Block Confidentiality Block| This document |
+-------+----------------------------+---------------+
```

Table 2

The Bundle Block Types namespace notes whether a block type is meant
for use in BP version 6, BP version 7, or both. The two block types
defined in this specification are meant for use with BP version 7.

11.2. Security Context Identifiers

BPSec has a Security Context Identifier field for which IANA is
requested to create and maintain a new registry named "BPSec Security
Context Identifiers". Initial values for this registry are given
below.

The registration policy for this registry is: Specification Required.

The value range is: unsigned 16-bit integer.

BPSec Security Context Identifier Registry

```
+-------+-------------+---------------+
| Value | Description |   Reference   |
+-------+-------------+---------------+
|   0   |   Reserved  | This document |
+-------+-------------+---------------+
```

Table 3

12. References

12.1. Normative References

[I-D.ietf-dtn-bpbis]
          Burleigh, S., Fall, K., and E. Birrane, "Bundle Protocol
          Version 7", draft-ietf-dtn-bpbis-18 (work in progress),
          January 2020.

   [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>.

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

   [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object
              Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049,
              October 2013, <https://www.rfc-editor.org/info/rfc7049>.

12.2. Informative References

   [I-D.birrane-dtn-sbsp]
              Birrane, E., Pierce-Mayer, J., and D. Iannicca,
              "Streamlined Bundle Security Protocol Specification",
              draft-birrane-dtn-sbsp-01 (work in progress), October
              2015.

   [I-D.ietf-dtn-bpsec-interop-sc]
              Birrane, E., "BPSec Interoperability Security Contexts",
              draft-ietf-dtn-bpsec-interop-sc-00 (work in progress),
              March 2019.

   [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst,
              R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant
              Networking Architecture", RFC 4838, DOI 10.17487/RFC4838,
              April 2007, <https://www.rfc-editor.org/info/rfc4838>.

   [RFC6257] Symington, S., Farrell, S., Weiss, H., and P. Lovell,
              "Bundle Security Protocol Specification", RFC 6257,
              DOI 10.17487/RFC6257, May 2011,
              <https://www.rfc-editor.org/info/rfc6257>.

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