

Delay-Tolerant Networking  
Internet-Draft  
Intended status: Experimental  
Expires: July 1, 2016

E. Birrane  
Johns Hopkins Applied Physics Laboratory  
December 29, 2015

**DTN Security Best Practices**  
**draft-birrane-dtn-sec-practices-00**

Abstract

This document describes best practices associated with achieving a variety of security use cases with the set of DTN-related standards.

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 July 1, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">1.1.</a>	<a href="#">Overview</a>	<a href="#">2</a>
<a href="#">1.2.</a>	<a href="#">Motivation for Application-Layer Security</a>	<a href="#">2</a>
<a href="#">1.3.</a>	<a href="#">Scope of Security Information</a>	<a href="#">3</a>
<a href="#">1.4.</a>	<a href="#">Classes of Security Information</a>	<a href="#">4</a>
<a href="#">1.5.</a>	<a href="#">Scope</a>	<a href="#">5</a>
<a href="#">2.</a>	<a href="#">Protocol Overview</a>	<a href="#">6</a>
<a href="#">2.1.</a>	<a href="#">Overview</a>	<a href="#">6</a>
<a href="#">2.2.</a>	<a href="#">Bundle Protocol Review</a>	<a href="#">6</a>
<a href="#">2.3.</a>	<a href="#">Bundle Protocol Security (BPSEC)</a>	<a href="#">6</a>
<a href="#">2.4.</a>	<a href="#">Bundle-in-Bundle Encapsulation</a>	<a href="#">7</a>
<a href="#">3.</a>	<a href="#">Policy Considerations</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">Best Practices</a>	<a href="#">9</a>
<a href="#">4.1.</a>	<a href="#">Overview</a>	<a href="#">9</a>
<a href="#">4.2.</a>	<a href="#">Bundle Source End-to-End Block Security</a>	<a href="#">9</a>
<a href="#">4.3.</a>	<a href="#">Waypoint Block Security</a>	<a href="#">10</a>
<a href="#">4.4.</a>	<a href="#">Security Destinations</a>	<a href="#">10</a>
<a href="#">4.5.</a>	<a href="#">Cascading Operations</a>	<a href="#">11</a>
<a href="#">4.6.</a>	<a href="#">Hop by Hop Authentication</a>	<a href="#">13</a>
<a href="#">4.7.</a>	<a href="#">Path Verification</a>	<a href="#">14</a>
<a href="#">4.8.</a>	<a href="#">Parallel Authenticators/Decrypters</a>	<a href="#">15</a>
<a href="#">4.9.</a>	<a href="#">Primary Block Integrity</a>	<a href="#">16</a>
<a href="#">4.10.</a>	<a href="#">Primary Block Privacy</a>	<a href="#">16</a>
<a href="#">5.</a>	<a href="#">IANA Considerations</a>	<a href="#">17</a>
<a href="#">6.</a>	<a href="#">Informative References</a>	<a href="#">17</a>
	<a href="#">Author's Address</a>	<a href="#">17</a>

## [1.](#) Introduction

### [1.1.](#) Overview

This document outlines the motivation for an end-to-end, application layer security capability as the collaborative effect of individual capabilities. Such a mix-and-match model of applying services allows for the more effective securing of a diverse set of disparate challenged internetworking scenarios.

In the context of this document, security refers to providing for the end-to-end integrity and confidentiality of application data.

### [1.2.](#) Motivation for Application-Layer Security

Path diversity in a packetized, wireless internetwork increases resiliency to loss of individual links. However, packetization and multi-path, multi-hop communication severs the relationship between user data and a communications link; it is no longer sufficient to



tightly control a single communication link to provide security for data exchange. The packets that comprise user data, by definition, may traverse multiple links as they traverse the network and accumulate at some user destination.

Securing link layers is not a sufficient mechanism for securing end-to-end data for two reasons, as follows.

**Impractical Coordination of Multiple Links:**

Every link and enclave participating in the message path must coordinate to ensure that a particular data exchange retains all necessary security services. This is intractable when thousands of packets representing a single set of user data flow over multiple links and through multiple enclaves.

**Shared Security Access Over Shared Links:**

Different users of an internetwork may require different security considerations. Since the concept of resource sharing drives the adoption of internetworking, multiple missions will want to use individual links to amortize the cost of resilient communications. If security is restricted to links only, then every user sharing the link must use the same security services of the link. In such a scenario, there is no mechanism to finely tune per-user security settings.

### **1.3. Scope of Security Information**

At least three scopes of security exist in a packetized internetwork: Link, Enclave, and End-to-End. These are based, loosely and conceptually, on the Unix file permission concept of "User", "Group", and "Other".



Layer	Responsibilities
Link	<ul style="list-style-type: none"> <li>- point-to-point data exchange protected from data corruption.</li> <li>- link-specific security mechanisms at both the physical and data layers.</li> <li>- Ensures transmissions over the link are authenticated and preserve the integrity and confidentiality of the message.</li> </ul>
Enclave	<ul style="list-style-type: none"> <li>- Bound administrative and/or technical domains.</li> <li>- Abstract link details when links within one enclave behave differently than links in another.</li> </ul>
End-to-End	<ul style="list-style-type: none"> <li>- Ensure that application data is secured regardless of links or enclaves.</li> <li>- Remove assumptions based on a particular path of a packet in the network or other underlying security mechanisms.</li> </ul>

#### Logical Security Scopes

#### **1.4. Classes of Security Information**

Three types of security-related information are considered by this document: Security-Related Protocols, Node Security Policy, and Cipher Suite Support.

##### Security-Related Protocols:

Protocols identify the data models, model encodings, and control information associated with the communication and application of the data model across a network.

##### Node Security Policy:

Security policy describes how individual nodes within a network populate the data models associated with the protocols providing data security. It is possible for multiple nodes in an internetwork to implement identical protocols but to use



different features of those protocols based on local or group policy. This policy may be derived from data directly or indirectly related to security and, therefore, policy drivers must be considered separately from security protocols.

#### Cipher Suite Support:

Separate from protocol features and the policy that determines what features to apply when, cipher suites generate the data that is carried by security protocols. Since multiple cipher suites can be used to generate the data used to populate the data model of security-related protocols, cipher suite support must be considered separately from protocols.

### **1.5. Scope**

This document addresses how to achieve a series of application-layer, end-to-end security functions via combinations of protocols, policies, and cipher suites. This document does not provide the full specification for any single protocol, policy, or cipher suite.

Specifically, this specification provides ways to achieve the following kinds of behavior in an internetwork supporting certain protocols and implementing certain policies and cipher suites.

#### Decoupled Routing and Security.

Original transmitters and forwarders of a bundle may wish to apply security settings based on some envisioned end point for a security service. However, it is unlikely in a general internetworking deployment that a node will know the exact path taken by a bundle through an internetwork. This is particularly the case when the internetwork spans multiple enclaves with different administrative policies. Therefore, security services must be independent of individual message paths.

#### Make Common Cases Simple and Efficient.

There exists a common set of security services that are applied to bundles, namely the end-to-end integrity and confidentiality of message payloads. While there exist many exotic permutations of security services for various internetworking use cases, this simple and common case must remain effective and efficient so as to not penalize simpler networks to accommodate complex networks, whenever possible.

#### Provide Security Services Equally.

Messages exchanged within a DTN may have multiple security services applied to different parts of them. For example, security services applied to message headers separately from secondary headers or payloads. To the extent possible, the





implementation of security functions should be agnostic to the type of data being secured.

## **2. Protocol Overview**

### **2.1. Overview**

This section provides a brief overview of the protocols considered by this best practice document. This section covers only those significant functional aspects necessary to inform the discussion of how to combine functions for security services. Protocols covered by this document include the Bundle Protocol (BP) [[RFC5050](#)], Bundle Protocol Security (BPsec) [[BPsec](#)], and Bundle-in-Bundle Encapsulation (BIBE) [[BIBE](#)].

### **2.2. Bundle Protocol Review**

The Bundle Protocol (BP) is a packetized, overlay, store-and-forward protocol proposed for the exchange of data in a variety of challenged internetworking scenarios. A BP protocol data unit (PDU) is characterized as a series of variable-length blocks, with two special blocks required in the bundle and all other blocks optional. The two required blocks are the primary block (which acts as a message header) and the payload block, which is a standard payload area. Additional blocks, called extension blocks and conceptually similar to secondary headers, may also be added to the bundle. Extension blocks may be added at any time, and by any node, as the bundle traverses the internetwork. Bundles are addressed using End Point Identifiers (EIDs), which identify an overlay destination (endpoint) in the internetwork. The mapping between EIDs and nodes in the network is many-to-many, so a node may be associated with several EIDs, and one EID may be associated with multiple nodes to form a multi-cast address.

### **2.3. Bundle Protocol Security (BPSEC)**

The security standard currently proposed for BP and DTN is the Bundle Protocol Security (BPSEC). BPsec defines security services captured in extension blocks that may be applied to discrete portions of a bundle. BPsec, as a protocol, operates at the "Group" and "World" layer, without reliance on link security mechanisms. Policy decisions on how BPsec services should or should not be applied to a bundle may or may not choose to consider link layer mechanisms.

BPsec provides two extension blocks that capture integrity and confidentiality services for other blocks within a bundle, as follows.



**Block Integrity Blocks (BIB):**

BIBs provide an integrity signature over some other block in the bundle, such that a change to the contents of the protected "target" block would be detected by comparing the signature captured in the BIB with a signature directly computed from the contents of the target block.

**Block Confidentiality Blocks (BCB):**

BCBs provide a confidentiality mechanism over some other block in the bundle. A BCB captures annotative information as to how a protected, "target" block has been encrypted and the content of the target block is re-written with ciphertext.

Note, BPsec does not specify the cipher suites used to populate the BIB and BCB blocks. The selection of cipher suites and keys to generate necessary data is a matter of policy.

**2.4. Bundle-in-Bundle Encapsulation**

Typically, the payload of a bundle contains some user or application data (or a fragmented portion of such data). The BIBE protocol provides a mechanism by which one bundle can be set as the payload of another bundle. This introduces the terminology of encapsulated and encapsulating bundles, as follows.

**Encapsulated Bundle:**

An encapsulated bundle is a bundle that is serialized, in whole, as the payload of some other bundle. Once encapsulated, the bundle is indistinguishable from a block of application payload on the wire and is not treated as a bundle until it is extracted at the destination of its encapsulating bundle. At the encapsulating bundle destination, the encapsulated bundle is extracted and passed to the destination as if it had been delivered there directly.

**Encapsulating Bundle:**

An encapsulating bundle is a bundle which has, as its payload, an encapsulated bundle. Any extension blocks or policy decisions made regarding this encapsulating bundle are separate from the encapsulated bundle. The encapsulated bundle is treated solely as a payload until the encapsulating bundle reaches its destination, at which point the encapsulating bundle is discarded and the encapsulated bundle is reconstituted and given to the node for processing.

The BIBE mechanism is used to create tunnels with the BP specification and is a useful way to maintain a separation between security and routing while allowing some way to introduce required



security waypoints in a path. Namely, while it is not possible, in the general case, to tell a single bundle to traverse multiple specific nodes from end to end, it is possible to establish multiple tunnels for the bundle to pass through using the BIBE mechanism.

### **3. Policy Considerations**

Policies and configurations must be documented separately from both implementing protocols and best practices. Since the primary value of sharing policy and configuration information is to ensure the interoperability of multiple security services this information should be standardized whenever possible. Security policy documents should identify what security services are required in given network deployments and what actions should be taken when messages do not adhere to these expectations.

The following policy scenarios are strongly recommended for consideration in any such documentation relating to standards for DTN security.

#### **Less Security Than Required:**

When the network requires a certain level of security, such as encrypted payloads or authenticated message exchange and a message is received without this information, the network must handle this in a uniform way. Most policies require not forwarding the message, but the level of logging, error messaging, and updates to local configurations should be discussed as a matter of policy.

#### **More Security Than Required:**

Similarly, when messages are received that contain authentication, integrity, or confidentiality when they should not, a decision must be made as to whether these services will be honored by the network.

#### **Security Evaluation In Transit:**

Some security services may be evaluated at a node, even when the node is not the bundle destination or a security destination. For example, a node may choose to validate an integrity signature of a bundle block. If an integrity check fails to validate, the intermediate node may choose to ignore the error, remove the offending block, or remove the entire bundle.

#### **Fragmentation:**

Policy must determine how security blocks are distributed amongst the new bundle fragments, so as to allow received fragments to be validated at downstream nodes



#### Block and Bundle Severability:

Distinct from fragmentation, nodes must decide whether a security error associated with a block implies a larger security error associated with the bundle. If blocks and bundles are considered severable, then an offending block may be omitted from the bundle. Otherwise, a bundle should be discarded whenever any of its constituent blocks are discarded.

## **4. Best Practices**

### **4.1. Overview**

Complex security activities are achieved through the combination of multiple discrete protocols rather than the creation of tightly-coupled, highly-purposed protocols. Given a set of loosely coupled, highly cohesive protocols, a set of best-practices can be provided to implement security operations.

### **4.2. Bundle Source End-to-End Block Security**

#### **4.2.1. Need**

This is the common case for block-level security services. In this case, a bundle source wishes to apply integrity and/or confidentiality to one or more blocks in a bundle and for these services to persist until the bundle reaches its destination.

#### **4.2.2. Recommended Practice**

This is the common case supported directly by BPsec without modification. In this case, each block being protected will have an additional security block (BIB or BCB) added to the bundle. The BIB and BCB blocks will contain all necessary security information based on cipher suites which must be selected in accordance with some policy at the node. Once the BIB and BCB blocks are added to the bundle, the bundle may be sent through the network and no additional operations are necessary until the bundle reaches the destination, at which point BIBs and BCBs are verified and processed in accordance with the BPsec specification.

#### **4.2.3. Additional Policy Considerations**

##### Transmit Rules:

The originating node must determine, by policy and configuration, what services are necessary based on the destination of the bundle.

##### Key and Cipher Suite Selection:





The originating node must determine which keys are used to configure which cipher suites will populate the necessary blocks.

### **4.3. Waypoint Block Security**

#### **4.3.1. Need**

Certain network configurations may require that security services be added to a bundle by a waypoint node rather than the originator of the bundle. A motivating example of this need is a network where a bundle requires only integrity services within an enclave but requires confidentiality before the bundle leaves the enclave to route over a more public network. In such cases, a gateway node at the border of the enclave and the public network may add confidentiality services to any bundle that does not already have such services before allowing the bundle to leave the enclave.

#### **4.3.2. Recommended Practice**

This is a minor extension to the case where the bundle source adds a security block. In this case, BPsec blocks (BIB and BCB) are also added to the bundle, but the security source of these blocks is listed as the waypoint node adding the block, rather than the bundle source node. The processing and behavior of the block is, otherwise, unchanged.

#### **4.3.3. Additional Policy Considerations**

The policy considerations for a waypoint adding a security service are the same as when a bundle source adds a security service.

### **4.4. Security Destinations**

#### **4.4.1. Need**

There may be times when a bundle is requested to go through a specific waypoint node en-route to its destination. From a security standpoint, this is typically done to ensure that some security result is achieved, for example ensuring that a bundle goes through a specific gateway for appending extra security services.

#### **4.4.2. Recommended Practice**

The current recommended practice for general networks to accomplish security-related destinations is to use the BIBE to wrap a bundle into an encapsulating bundle, and then use the security destination as the destination of the encapsulating bundle. In this way, the



routing mechanism used in the network is not coupled to the security system, and the encapsulated bundle is not burdened with tracking multiple intermediate destinations.

This is the equivalent of creating a tunnel between the current node and the security destination. If, at the security destination, a subsequent security destination is necessary, the process may be repeated.

#### **4.4.3. Additional Policy Considerations**

##### **Security Destination Identification**

To require security destinations, there must be some mechanism by which security destinations are identified and some other mechanism to associate bundles with those security destinations.

##### **Extension Block Handling**

Care must be taken to process, correctly, extension blocks both in the encapsulated bundle and the encapsulating bundle. There may exist extension blocks in the encapsulated bundle that wish to be processed at every hop taken by the bundle, even while it is encapsulated. In such situations it might be possible to carry these blocks in the encapsulating bundle and merge them back into the encapsulated bundle at the security destination. Note, however, there is no standard for this.

#### **4.5. Cascading Operations**

##### **4.5.1. Need**

Cascading operations are security services that are applied to the same data multiple times, such as is the case when performing super-encryption. The BPSec standard does not allow the same security service to be applied to the same target data multiple times (for example, a payload cannot be encrypted twice with two BCBs).

There are several reasons for wanting to replace or modify security services found in a bundle. Policy may require a stronger security service before a bundle is allowed to leave an enclave.

Alternatively, portions of the network may be configured with different cipher suite support rendering in-situ integrity checking impossible unless a new integrity signature in a supported cipher suite is added. At times, encrypting parts of an existing BCB or BIB to hide cipher suite details may be required.



#### **4.5.2. Recommended Practice**

When cascading operations are to be applied from the current node to the bundle destination each of these practices can be applied directly to the bundle. In cases where cascading operations are only to be applied from the current node to someplace other than the bundle destination then, first, Security Destination best practices must be applied.

There are three recommended ways to satisfy this need in BP networks: Bundle encapsulation, Block encapsulation, and custom blocks.

##### **Bundle Encapsulation:**

There have been many proposals relating to how to stack security operations amongst blocks in a bundle. However, each of these results in complex situations regarding the order in which operations are applied and how to preserve meaning in the presence of fragmentation.

This approach maintains the BPsec restriction of one security service per target in a single bundle and use BIBE and encapsulation. In such a scheme, the existing bundle becomes the encapsulated bundle. The encapsulating bundle then applies whatever additional security services are necessary to its payload, thereby applying them to the encapsulated bundle.

##### **Block Encapsulation**

There is, currently, no standard for block encapsulation. However, the target block and its associated security blocks may, themselves, be packed into a single new block within the bundle and new security services may be added to that encapsulating block.

##### **Custom Security**

A set of custom security blocks can be defined by a particular network that operate orthogonally from the BPsec security blocks. This provides fine-grained control over security in specific network deployments. This method is only practical in closed, highly controlled networks where custom block definition and processing is both technically feasible and economical.

#### **4.5.3. Additional Policy Considerations**

##### **Security Service Identification:**

Nodes in the network must be able to identify appropriate security services and cipher suites to some understood destination.



## **4.6. Hop by Hop Authentication**

### **4.6.1. Need**

Hop by hop authentication ensures that a received bundle's last hop (i.e. most recent forwarder) matches what the bundle claims it's last hop to be. Checking each hop along a path in the network is one way to establish a chain of trust. More importantly, verifying the appropriateness of the node who sent a received bundle is a way of protecting networks against certain types of attacks. Specifically, the internals of a network can be protected against resource-consuming attacks if a gateway node can detect inappropriate traffic and prevent its ingest into the rest of the network.

### **4.6.2. Recommended Practice**

There are three recommended ways to satisfy this need in BP networks: Reliable link layers, integrity on ephemeral blocks, and canonicalized whole-bundle signatures.

#### Reliable Link Layers:

By definition, link-layer security secures the transmission between two points (a link) in a network. Wherever hop-by-hop authentication is required, the network might simply require the use of a secure point-to-point link layer. In such a case, there is no need for an application-layer mechanism for hop-by-hop security.

#### Ephemeral Block Integrity:

Assuming that secure link layers are not guaranteed to be available, a second practice is to insert a short-lived (ephemeral) block into a bundle just prior to transmission that identifies the transmitter of the bundle, sign that block with a BPSec BIB, and then transmit the bundle.

Such an ephemeral block is defined in the BP specification as a Prior Hop Notification (PHN) Block and can be used for this purpose. Alternatively, a user may define their own block type which can hold any information they wish, to include a signature calculated over the entire bundle rather than an assertion of the previous bundle transmitter.

When the bundle is received at the next hop, this ephemeral block can be verified to ensure that the node that signed the block is the same as the node referenced in the block. At this point, the ephemeral block and its associated BIB are no longer necessary and can be either removed from the bundle or kept for historical accounting with the bundle.





#### Canonicalized Whole-Bundle Signature:

The signing of an ephemeral block such as the PHN does not provide a guarantee that the contents of the bundle remained unchanged across the hop. In the extreme case where the entire contents of the bundle must be authenticated at every hop in the bundle, a canonicalized form of the bundle must be generated and signed by the transmitting node and then checked at the next hop of the bundle.

The most straightforward way to achieve this is to use the BIBE to encapsulate the entire bundle as the payload of an encapsulating block, place a BIB on the encapsulating block payload, and then make the next hop the destination of the encapsulating block.

#### [4.6.3.](#) Additional Policy Considerations

##### Applicability:

By global policy or by next-hop, a transmitting node must have some way of determining that hop-by-hop authentication is necessary and that either a secure link layer, an ephemeral block, or some other method is needed to protect the transmission.

##### Link Layer Identification:

When using secure link layers, the BPA must have some mechanism of determining if the link layer selected for transmission has an appropriate security model.

#### [4.7.](#) Path Verification

##### [4.7.1.](#) Need

A common request in a secured internetwork is to provide a signed listing of each node traversed by a bundle on its way from sender to receiver.

##### [4.7.2.](#) Recommended Practice

There are three recommended practices for accomplishing this task: Per-Hop Extension Blocks, a Signed Log, and an Encrypted Log.

##### Per-Hop Extension Blocks:

A new extension block can be added to the bundle at each node, and that new block can be integrity signed by BPsec. In networks using the Previous Hop Notification (PHN) block, the PHNs can be signed and kept for each hop.



**Signed Log:**

A single extension block can be defined to act as a log book of visited nodes, such that each node visited by the bundle adds a new, signed data entry into the log.

**Encrypted Log:**

This approach works similarly to a Signed Log, except that the extension block is encrypted with a BCB and only those nodes in the network with the appropriate keys can decrypt and modify the log.

**4.7.3. Additional Policy Considerations**

None.

**4.8. Parallel Authenticators/Decrypters****4.8.1. Need**

Security in the context of multicasting presents challenging operational concepts for how to validate a received bundle that carries multiple integrity signatures. In this case, a bundle should validate a security service if any one of multiple security data items is verified.

**4.8.2. Recommended Practice**

It is recommended that a multi-case cipher suite specification be defined and used to generate multiple signatures (for integrity) or multiple ciphertexts (for confidentiality). This approach allows BPsec to operate without modification, as the cipher suite implementation both generates and verifies security results.

Multiple signatures would be stored directly in the BIB as part of cipher suite data. Such cipher suites could verify a signature if any 1 signature matched, if N of M signatures matched, or if all signatures match, based on policy.

Multiple encryptors could work by encrypted the plaintext multiple times to generate multiple ciphertexts which, in total, would replace the plain text in a specific block in the bundle. Additionally, the bundle source or other identifying information could be encrypted once per key and stored as additional authenticated data. On decryption, a node could determine the appropriate ciphertext to use by decrypting the bundle source from the additional authenticated data and then decrypting the ciphertext associated with that key.



#### **4.8.3. Additional Policy Considerations**

Key Lists:

Each node that encrypts, decrypts, or authenticates based on a multi-cast cipher suite would need to keep a list of each key used.

### **4.9. Primary Block Integrity**

#### **4.9.1. Need**

It may be necessary to ensure that the primary block in a bundle has not changed since the bundle was first transmitted.

#### **4.9.2. Recommended Practice**

The BPsec allows the BIB to target the primary block, just as it can target any other block in a bundle. As such, this capability can be accomplished by inserting a BIB in the bundle whose target is the primary block.

#### **4.9.3. Additional Policy Considerations**

None.

### **4.10. Primary Block Privacy**

#### **4.10.1. Need**

It may be necessary in certain cases to hide the contents of a primary block for portions of a bundle journey.

#### **4.10.2. Recommended Practice**

There are two recommended practices for accomplishing this task: Encapsulation and Custom Extension Blocks.

Encapsulation:

The most straightforward way to hide portions of the primary block is to use BIBE to encapsulate the entire bundle. Then, the encapsulating block can have whatever primary block information is necessary to get the bundle through the portions of the network where the original primary block should be hidden. In this case, the encapsulated bundle should be encrypted with a BCB from the encapsulating block.

Custom Extension Block



A custom extension block may be defined to hold the contents of the primary block. A temporary primary block can be constructed at various points in the network as part of processing the custom extension block. This temporary primary block would have only that information necessary to get the bundle to some next known node.

#### **4.10.3. Policy Considerations**

While there is no specific policy consideration, the concept of performing surgery on the primary block of a bundle in transit must be taken with great care.

### **5. IANA Considerations**

This document has no fields registered by IANA.

### **6. Informative References**

- [BIBE] Burleigh, S., "Bundle-in-Bundle Encapsulation", [draft-irtf-burleigh-bibe-00](#) (work in progress), March 2013.
- [BPsec] Birrane, E., Mayer, J., and D. Iannicca, "Bundle Protocol Security", [draft-ietf-dtn-bpsec-00](#) (work in progress), December 2015.
- [RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", [RFC 5050](#), DOI 10.17487/RFC5050, November 2007, <<http://www.rfc-editor.org/info/rfc5050>>.

#### **Author's Address**

Edward J. Birrane  
Johns Hopkins Applied Physics Laboratory

Email: [Edward.Birrane@jhuapl.edu](mailto:Edward.Birrane@jhuapl.edu)



