The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

CCSDS STREAMLINED BUNDLE SECURITY PROTOCOL SPECIFICATION

DRAFT RECOMMENDED STANDARD

CCSDS 734.5-R-1

RED BOOK

March 2018

The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

CCSDS STREAMLINED BUNDLE SECURITY PROTOCOL SPECIFICATION

DRAFT RECOMMENDED STANDARD

CCSDS 734.5-R-1

RED BOOK
March 2018

AUTHORITY

Issue:	Red Book, Issue 1
Date:	March 2018
Location:	Not Applicable

(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
E-mail: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT:)

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
  - -- The **standard** itself.
  - -- The anticipated date of initial operational capability.
  - -- The anticipated duration of operational service.
- Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in the *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the e-mail address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Standard. Its 'Red Book' status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organizations. As such, its technical contents are not stable, and several iterations of it may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document's technical content.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 734.5-R-1	CCSDS Streamlined Bundle Security Protocol Specification, Draft Recommended Standard, Issue 1	March 2018	Current draft

CONTENTS

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE

This document defines a Recommended Standard for the CCSDS Streamlined Bundle Security Protocol (SBSP) specification to define security features for the CCSDS Bundle Protocol (BP) (reference [1]) intended for use in delay-tolerant networks in order to provide Delay-Tolerant Networking (DTN) security services.

SBSP provides authentication, integrity, and confidentiality for bundles along a path through a DTN. SBSP is based upon the IETF Bundle Protocol Security Specification (reference [2]), 'streamlined' to remove complex cases that introduce ambiguities and make implementation difficult.

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

1.2 SCOPE

This Recommended Standard is designed to be applicable to any kind of space mission or infrastructure that deploys the Bundle Protocol Specification for communications between nodes. It is intended that this Recommended Standard become a uniform standard among all CCSDS Agencies. This Recommended Standard is intended to be applied to all systems that claim conformance to the CCSDS Bundle Protocol.

This specification does not address:

– individual cipher suite implementations, as definition and enumeration of cipher suites should be undertaken in separate specification documents;

– implementation of security policy;

– any security policy for SBSP;

NOTE – This document does recommend security policy considerations when building a security policy. Security policies are typically based on the nature and capabilities of individual networks and network operational concepts.

– how to combine SBSP security blocks with protocols other than BP, other BP extension blocks, or other best practices to achieve security in any particular network implementation;

– key management.

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

1.3 ORGANIZATION OF THIS RECOMMENDED STANDARD

This Recommended Standard is organized as follows:

- Section 2 contains an overview of the Bundle Protocol Security Specification.
- Section 3 contains the security block definitions.
- Section 4 describes the security processing.
- Section 5 contains thoughts on key management.
- Section 6 contains policy considerations.
- Section 7 contains security considerations.
- Section 8 contains conformance requirements.
- Section 9 describes the protocol block types and parameters.

1.4 TERMINOLOGY

source: The bundle node from which a bundle originates.

destination: The bundle node to which a bundle is ultimately destined.

forwarder: The bundle node that forwarded the bundle on its most recent hop.

intermediate receiver, **waypoint**: The neighboring bundle node to which a forwarder forwards a bundle.

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

Figure 1-1 below is adapted from the CCSDS Bundle Protocol Specification (reference [1]) and shows four bundle nodes (denoted BN1, BN2, BN3, and BN4) that reside above some Transport Layer(s). Three distinct transport and network protocols (denoted T1/N1, T2/N2, and T3/N3) are also shown.

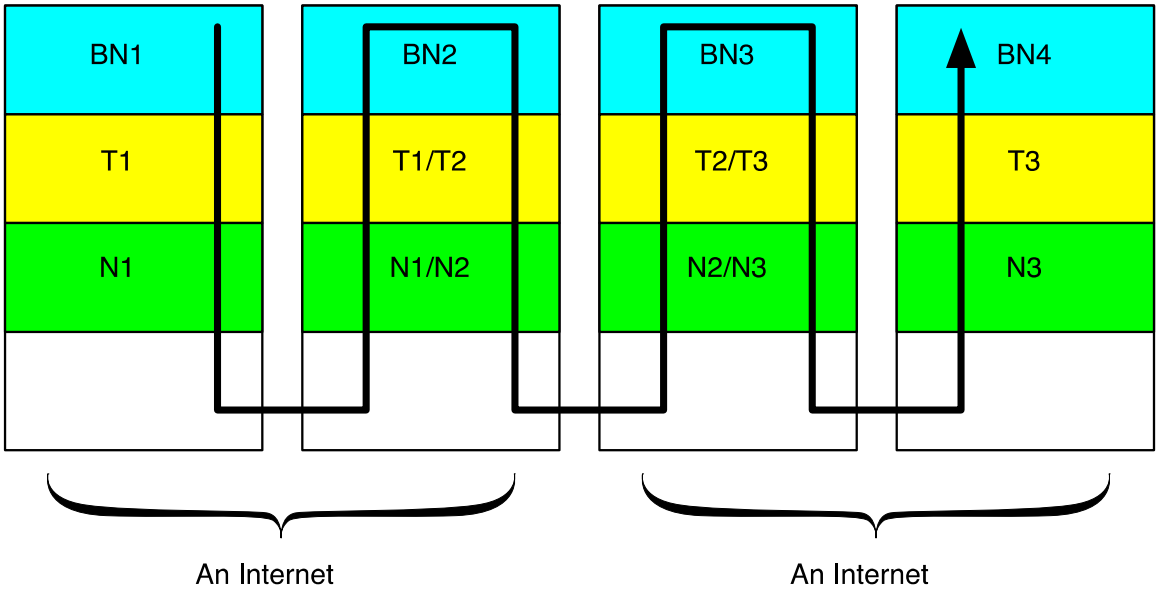

Figure 1-1: Bundle Nodes Sit at the Application Layer of the Internet Model

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

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

The following security-specific DTN terminology is used:

security service: The security features supported by this specification: authentication, integrity, and confidentiality.

security target: The portion of a bundle (e.g., the primary block, payload block, extension block, or entire bundle) that receives a security service as part of a security operation.

security block: A single instance of a SBSP extension block in a bundle.

security operation: The application of a security service to a specific security target, notated as OP(security service, security target), e.g., OP(authentication, bundle) or OP(confidentiality, payload). Every security operation in a bundle must be unique, meaning that a security service can be applied to a security target only once in a bundle. A security operation may be implemented by one or more security blocks.

1.5 NOMENCLATURE

1.5.1 NORMATIVE TEXT

The following conventions apply for the normative specifications in this Recommended Standard:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.5.2 INFORMATIVE TEXT

In the normative sections of this document, informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.6 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

[1] *CCSDS Bundle Protocol Specification.* Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 734.2-B-1. Washington, D.C.: CCSDS, September 2015.

[2] E. Birrane and K. McKeever. *Bundle Protocol Security Specification*. Internet-Draft. Reston, Virginia: ISOC, July 1, 2017.

[3] W. Eddy and E. Davies. *Using Self-Delimiting Numeric Values in Protocols*. RFC 6256. Reston, Virginia: ISOC, May 2011.

[4] K. Scott and S. Burleigh. *Bundle Protocol Specification*. RFC 5050. Reston, Virginia: ISOC, November 2007.

[5] T. Berners-Lee, R. Fielding, and L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. STD 66. Reston, Virginia: ISOC, January 2005.

[6] R. Housley. *Cryptographic Message Syntax (CMS)*. RFC 5652. Reston, Virginia: ISOC, September 2009.

2 OVERVIEW

2.1 GENERAL

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

2.2 BLOCK-LEVEL GRANULARITY

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

Security services within this specification need to provide block level granularity where applicable such that different blocks within a bundle may have different security services applied to them. Bundles containing Endpoint IDentifier (EID) references within extension blocks are prohibited by this security protocol.

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

2.3 MIXED SECURITY POLICY

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

Extension blocks representing security services must have their block processing flags set such that the block (and bundle, where applicable) will be treated appropriately by non-security-aware nodes.

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

2.4 USER-SELECTED CIPHERSUITES

The security services defined in this specification rely on a variety of ciphersuites providing integrity signatures, ciphertext, and other information necessary to populate security blocks. Users may wish to select differing ciphersuites to implement different security services. For example, some users may wish to use a SHA-1-based hash for integrity whereas other users may require a SHA-2 hash instead. The security services defined in this specification need to provide a mechanism for identifying what ciphersuite has been used to populate a security block.

2.5 DETERMINISTIC PROCESSING

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

3 SECURITY BLOCK DEFINITIONS

3.1 GENERAL

3.1.1 Bundles that implement security services may include zero or more instances of the following bundle protocol block types subject to the constraints in this section.

a) Bundle Integrity Block (BIB);

NOTE – The BIB is used to ensure the authenticity and integrity of its security target from the bundle source, which creates the BIB, to the bundle destination, which verifies the BIB authenticator.

b) Bundle Confidentiality Block (BCB).

NOTE – The BCB indicates the manner in which the security target has been encrypted, in whole or in part, at the bundle source in order to protect its content while in transit to the bundle destination.

3.1.2 A *security operation* is defined as the combination of an instance of a Bundle Integrity Block or a Bundle Confidentiality Block together with the block or blocks to which the security operation applies, known as the target block or blocks.

3.1.3 A security operation must not be applied more than once in a bundle.

NOTE – For example, the two security operations: (integrity, payload) and (integrity, payload) are considered redundant and cannot appear together in a bundle. However, the two security operations (integrity, payload) and (integrity, extension_block_1) may both be present in the bundle. Also, the two security operations (integrity, extension_block_1) and (integrity, extension_block_2) are unique and may both appear in the same bundle.

3.1.4 The same security service may be applied to multiple targets. In such a case, all security operations represented in the security block must be applied/evaluated together.

NOTE – Many of the fields in these block definitions use the Self-Delimiting Numeric Value (SDNV) type whose format and encoding is as defined in RFC 6256 (reference [3]).

3.1.5 In any single bundle there must be only one instance of each block type that is the target of a security operation.

3.2 CANONICALIZATION

Each security block must use the Canonical Bundle Block Format as defined in section 4.5.2 of the Bundle Protocol Specification RFC (reference [4]).

NOTE – Each security block thus comprises the following elements:

- Block Type Code;
- Block Processing Control Flags;
- Block EID Reference List (optional);
- Block Data Length;
- Block Type Specific Data Fields.

3.3 GENERIC SECURITY BLOCK STRUCTURE

3.3.1 The structures of the BIB and BCB Block Type Specific Data Fields are identical and shall follow the Generic Security Block Structure (GSBS) as illustrated in figure 3-1.

# Security Targets (SDNV)	Security Targets (Compound)
Cipher Suite ID (SDNV)	Cipher Suite Flags (SDNV)
*Cipher Suite Parameters Length (SDNV)	*Cipher Suite Parameters Data (Compound)
Security Result Length (SDNV)	Security Result Data (Compound)

Figure 3-1: BIB and BCB Block Structure

NOTES:

1 Although the diagram hints at a fixed-format layout, this is purely for the purpose of exposition. All fields are variable in length using SDNVs. In this figure, field names prefaced with '*' are optional, and their inclusion in the block is indicated by the Cipher Suite Flags field.

2 The format of compound fields is defined below in 3.3.3.

3.3.2 The block fields are defined as follows.

3.3.2.1 # Security Targets (SDNV)—The number of security targets for this security block. This value must be at least 1.

3.3.2.2 Security Targets (Compound)—The 'data' portion of each element of the security targets field identifies one of the targets of the associated security operation. As discussed in 3.3.2.1 the elements of this field must reference singleton block types.

3.3.2.3 Cipher Suite ID (SDNV)—The cipher suite used to implement the security service represented by this block and applied to each security target.

3.3.2.4 Cipher Suite flags (SDNV)—The optional security block fields present in the block. The structure of the Cipher Suite Flags field shall be as shown in figure 3-2.

a) bits 7–1 are reserved for future use.

b) bit 0, param, indicates whether or not the Cipher Suite Parameters Length and Cipher Suite Parameters Data fields are present.

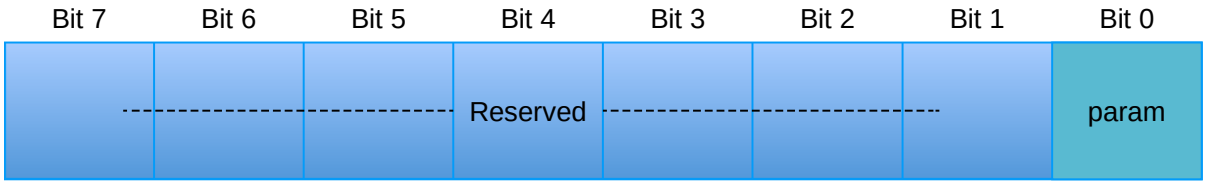

Figure 3-2: Cipher Suite Flags

3.3.2.5 (Optional) Cipher Suite Parameters—If indicated by a value of 1 in the param bit (bit 0) of the Cipher Suite Flags, the following two fields shall be present.

3.3.2.5.1 Cipher Suite Parameters Length (SDNV)—The length of the next field, which is the cipher suite parameters data field.

3.3.2.5.2 Cipher Suite Parameters Data (Compound)—Parameters to be used with the cipher suite in use, e.g., a key identifier or initialization vector (IV).

NOTE – Subsection 3.8 provides a list of parameters and their types. The particular set of parameters that is included in this field and the encoding of those parameters are defined as part of a cipher suite specification.

3.3.2.5.3 Security Result Length (SDNV)—The length of the next field, which is the security result data field.

3.3.2.5.4 Security Result Data (Compound)—The results of the appropriate cipher suite specific calculation (e.g., a signature, Message Authentication Code [MAC], integrity check value, or cipher-text block key).

3.3.3 The format of a compound field is defined as follows.

NOTE – The number and types of items that may appear in the cipher suite parameters and/or security result fields are defined by the particular cipher suite description. A cipher suite may support several instances of the same type within a single block.

3.3.3.1 Each item in any compound field for cipher suite parameters and/or security results shall be represented as a type-length-value tuple.

3.3.3.2 Type shall be a single byte indicating the item.

3.3.3.3 Length shall be the count of data content bytes to follow, represented as an SDNV-encoded integer.

3.3.3.4 Value shall be the data content of the item.

3.4 BLOCK INTEGRITY BLOCK

3.4.1 The block-type code value for Bundle Integrity Blocks must be 0x03.

3.4.2 The block processing control flags value can be set to whatever values are required by local policy.

NOTE – Cipher suite designers should carefully consider the effect of setting flags that either discard the block or delete the bundle in the event that this block cannot be processed.

3.4.3 The security target(s) must uniquely and unambiguously identify a block or set of blocks by block type within the bundle.

3.4.4 The security target of a BIB must not be a security block defined in this specification (e.g., a BCB).

3.4.5 The reserved block type 0x01 shall be used to specify the singleton payload block.

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

3.4.7 The security result field must be present.

3.4.8 The security result must contain the result of applying the cipher suite calculation (e.g., the MAC or signature) to the relevant parts of the security target, as specified in the cipher suite definition.

3.4.9 If the cipher suite does not protect the entire complete, original security target, the cipher suite parameters must specify which bytes of the security target are protected.

NOTES

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

2 For some cipher suites, (e.g., those using asymmetric keying to produce signatures or those using symmetric keying with a group key), the security information may be

checked at any hop on the way to the destination that has access to the required keying information, in accordance with 3.8.

3 When custody transfer is employed, the use of a generally available key is recommended and all nodes should verify each bundle before accepting custody.

3.5 BLOCK CONFIDENTIALITY BLOCK

3.5.1 The block-type code value for Block Confidentiality Blocks must be 0x04.

3.5.2 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.

NOTE – The presence of a BCB in each fragment indicates to a receiving node that the payload portion of each fragment represents cipher-text. Cipher suite designers should carefully consider the effect of setting flags that either discard the block or delete the bundle in the event that this block cannot be processed.

3.5.3 The security target must uniquely identify a singleton block by block type within the bundle. The security target for a BCB may reference the payload block, a non-security extension block, or a BIB block.

NOTE – This is different from a Bundle Integrity Block, whose security target cannot be a Bundle Confidentiality Block.

3.5.4 The reserved block type 0x01 shall be used to specify the singleton payload block.

3.5.5 The cipher suite ID must be documented as a confidentiality cipher suite.

3.5.6 Any additional bytes generated as a result of encryption and/or authentication processing of the security target should be placed in an 'integrity check value' field (see 3.8) in the security result of the BCB.

3.5.7 The security result must be present in the BCB.

NOTES

1 This compound field normally contains fields such as an encrypted bundle encryption key and/or authentication tag (integrity check value).

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

3.5.8 When the security target of a BCB is the bundle payload, the BCB must not alter the size of the payload block body data.

3.5.9 Cipher suites should place any block expansion, such as authentication tags (integrity check values) and any padding generated by a block-mode cipher, into an integrity check value item in the security result field (see 3.8) of the BCB.

3.5.10 If the cipher suite does not protect the entire complete, original security target body data, the BCB for that security target must specify, as part of the cipher suite parameters, which bytes of the security target body data are protected.

3.5.11 The BCB's 'Discard if block cannot be processed' flag may be set independently from its security target's 'Discard if block cannot be processed' flag.

NOTES

1 Whether or not the BCB's 'discard' flag is set is an implementation/policy decision for the encrypting node.

2 Fragmentation, reassembly, and custody transfer are adversely affected by a change in size of the payload due to ambiguity about what byte range of the block is actually in any particular fragment. These requirements for 'in-place' encryption allow fragmentation, reassembly, and custody transfer to operate without knowledge of whether or not encryption has occurred.

3.6 BLOCK INTERACTIONS

NOTES

1 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.

2 If confidentiality is being applied to a target that already has integrity applied to it, then an undesirable condition occurs where a security-aware intermediate node would be unable to check the integrity result of a block because the block contents have been encrypted after the integrity signature was generated. The requirements in this section address this concern by imposing a necessary ordering when applying security operations within a bundle.

3.6.1 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 and also in cases where the BPA is a waypoint adding new security blocks to a bundle that already contains security blocks.

3.6.2 The following processing rules must be followed.

3.6.2.1 If confidentiality is to be applied to a target, it must also be applied to every integrity operation already defined for that target.

NOTE – This means that if a BCB is added to encrypt a block, another BCB must also be added to encrypt any BIB that also targets that block.

3.6.2.2 An integrity operation must not be applied to a security target if a BCB in the bundle shares the same security target.

NOTE – This prevents ambiguity in the order of evaluation when receiving a BIB and a BCB for a given security target.

3.6.2.3 An integrity value must not be evaluated if the BIB providing the integrity value is the security target of an existing BCB block in the bundle.

NOTE – In such a case, the BIB data contains cipher-text as it has been encrypted.

3.6.2.4 An integrity value must not be evaluated if the security target of the BIB is also the security target of a BCB in the bundle.

NOTE – In such a case, the security target data contains cipher-text as it has been encrypted.

3.6.2.5 As stated in 3.4.4, a BIB must not have a BCB as its security target. BCBs may embed integrity results in their cipher suite parameters.

3.7 MULTI-TARGET BLOCK DEFINITIONS

A security block may target multiple security targets if and only if all cipher suite parameters and key information are common for each security operation. The following processing directives apply for these multi-target blocks.

a) If a security block has more than one security target, then each type identifier in the security result TLV must be interpreted as a tuple with the first entry being the security target for which the security result applies and the second entry being the type value enumeration of the security result value.

b) If the security block has a single security target, the type field of every entry in the security result array must simply be the type field and must not be a tuple as described above.

3.8 CIPHER SUITE PARAMETER AND RESULT TYPES

Cipher suite parameter shall be as defined in table 3-1.

Table 3-1: Cipher Suite Parameters and Result Fields

Type	Name	Description	Field
0	Reserved		
1	Initialization Vector (IV)	A random value, typically eight to sixteen bytes.	Cipher Suite Parameters
2	Reserved		
3	Key Information	Material encoded or protected by the key management system and used to transport an ephemeral key protected by a long-term key.	Cipher Suite Parameters
4	Content Range	Pair of SDNV values (offset,length) specifying the range of payload bytes to which an operation applies. The offset must be the offset within the original bundle, even if the current bundle is a fragment.	Cipher Suite Parameters
5	Integrity Signatures	Result of BIB digest or other signing operation.	Security Results
6	Unassigned		
7	Salt	An IV-like value used by certain confidentiality suites.	Cipher Suite Parameters
8	BCB Integrity Check Value (ICV) / Authentication Tag	Output from certain confidentiality cipher suite operations to be used at the destination to verify that the protected data has not been modified. This value may contain padding if required by the cipher suite.	Security Results
9-255	Reserved		

3.9 DISCUSSION—SBSP BLOCK EXAMPLE

An example of SBSP blocks applied to a bundle is shown in table 3-2. In this table the first column represents blocks within a bundle and the second column represents the block type.

Table 3-2: Sample Use of SBSP Blocks

Block in Bundle	ID
Primary Block	0x00
Payload Block	0x01
BIB OP(integrity, target=0x01)	0x02
BCB OP(confidentiality, target=0x01)	0x03

In this example a bundle has two non-security-related blocks: the primary block (0x00) and a payload block (0x01). The following security operations are applied to this bundle:

– an integrity signature applied to the payload block;

– confidentiality for the payload block.

4 SECURITY PROCESSING

4.1 OVERVIEW—DISCUSSION OF CANONICAL FORMS

To verify the signature of a bundle, the exact same bits, in the exact same order, must be input to the calculation upon verification as were input upon initial computation of the original signature value. Consequently, a node must not change the encoding of any URI (reference [5]) in the dictionary field (e.g., changing the DNS part of some HTTP URL from lower case to upper case). Because bundles may be modified while in transit (either correctly or due to implementation errors), canonical forms of security targets need to be defined.

4.2 CANONICALIZATION ALGORITHMS

4.2.1 The three types of blocks that may undergo block canonicalization are the primary block, the payload block, and extension blocks.

4.2.2 Where a block contains SDNVs, the canonical form shall be an eight-byte fixed-width integer field in network byte order representing the ordinal number of the SDNV.

NOTE – The size of eight bytes is chosen because implementations may treat larger SDNV values as invalid, as noted in the BP specification (reference [1]).

4.2.3 The canonical form of the primary block shall be as shown in figure 4-1.

Version	Processing flags (incl. COS and SRR)
Canonical primary block length	
Destination endpoint ID length	
Destination endpoint ID	
Source endpoint ID length	
Source endpoint ID	
Report-to endpoint ID length	
Report-to endpoint ID	
Creation Timestamp (2 x SDNV)	
Lifetime	

Figure 4-1: The Canonical Form of the Primary Bundle Block

NOTE – The values in the canonical form of the primary block are as follows.

4.2.3.1 The version value shall be the single-byte value in the primary block.

4.2.3.2 The processing flags value in the primary block shall be an SDNV that includes the Class Of Service (COS) and Status Report Request (SRR) fields. For purposes of canonicalization, the unpacked SDNV shall be ANDed with mask 0x0000 0000 0007 C1BE.

NOTE – The mask above sets to zero all reserved bits and the ‘bundle is a fragment’ bit.

4.2.3.3 The canonical primary block length value shall be a four-byte value containing the length (in bytes) of this structure, in network byte order.

4.2.3.4 The destination endpoint ID length and value shall be the length (represented as a four-byte value in network byte order) and value of the destination endpoint ID from the primary bundle block.

4.2.3.5 The URI shall be copied from the relevant part(s) of the dictionary block and shall not itself be canonicalized.

4.2.3.6 Although the dictionary entries contain 'null-terminators', the null-terminators shall not be included in the length or the canonicalization of URIs.

4.2.3.7 The source endpoint ID length and value shall be handled similarly to the destination endpoint ID.

4.2.3.8 The report-to endpoint ID length and value shall be handled similarly to the destination endpoint ID.

4.2.3.9 The unpacked SDNVs for the creation timestamp and lifetime shall be copied from the primary block.

4.2.3.10 The fragment offset and total application data unit lengths shall be ignored during canonicalization.

NOTES

1 If the payload data to be canonicalized is less than the complete, original bundle payload, the offset and length are specified in the cipher suite parameters.

2 Essentially, canonicalization of the primary block de-references the dictionary block, adjusts lengths where necessary, and ignores flags that may change in transit.

4.2.4 When canonicalizing the payload block, the block processing control flags value used for canonicalization shall be the unpacked SDNV value with reserved and mutable bits masked to zero. The unpacked value shall be ANDed with mask 0x0000 0000 0000 0077.

NOTE – The mask above sets to zero the reserved bits and the 'last block' bit.

4.2.4.1 Payload blocks shall be canonicalized as-is except where only a portion of the payload data is to be protected.

4.2.4.2 If only a portion of the payload is covered by the security operation, only those bytes of the payload that are covered shall be included in the canonical form.

NOTE – In this case, additional cipher suite parameters are required to specify which part of the payload is protected, as discussed further below.

4.2.5 When canonicalizing an extension block, the block processing control flags value used for canonicalization shall be the unpacked SDNV value with reserved and mutable bits masked to zero. The unpacked value shall be ANDed with mask 0x0000 0000 0000 0057.

NOTE – The mask above sets to zero the reserved bits, the 'last block' flag, and the 'Block was forwarded without being processed' bit.

4.2.5.1 The 'Block was forwarded without being processed' flag shall be ignored during canonicalization.

NOTE – This flag is ignored because the bundle may pass through nodes that do not understand that extension block; this would set this previously unset flag.

4.2.5.2 The block-length shall be canonicalized as its unpacked SDNV value. If the data to be canonicalized is less than the complete, original block data, this field shall contain the size of the data being canonicalized (the 'effective block') rather than the actual size of the block.

4.2.6 DISCUSSION

The canonical forms for the bundle and various extension blocks are not transmitted. Canonical form is simply an artifact used as input to digesting.

The reserved flags are omitted because it cannot be determined if they will change in transit. The masks specified above will have to be revised if additional flags are defined and they need to be protected.

The URI encoding used for canonicalization does not preserve the null-termination convention from the dictionary field, nor is the scheme and Scheme-Specific Part (SSP) canonicalized separately. Instead, the byte array < scheme name > : < SSP > is used in the canonicalization.

The URI encoding will cause errors if any node rewrites the dictionary content (e.g., changing the DNS part of an HTTP URL from lower case to upper case). This could happen transparently when a bundle is synched to disk using one set of software and then read from disk and forwarded by a second set of software. Because there are no general rules for canonicalizing URIs (or IRIs), this problem may be an unavoidable source of integrity failures.

All SDNV fields here are canonicalized as eight-byte unpacked values in network byte order. Length fields are canonicalized as four-byte values in network byte order. Encoding does not need optimization since the values are never sent over the network.

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

Cipher suites may define their own canonicalization algorithms and require the use of those algorithms instead of the ones provided in this specification.

Every bundle has a primary block that contains the source and destination endpoint IDs, and possibly other EIDs (in the dictionary field), which cannot be encrypted. If endpoint ID confidentiality is required, then bundle-in-bundle encapsulation can solve this problem in some instances.

Similarly, confidentiality requirements may also apply to other parts of the primary block (e.g., the current custodian). That is supported in the same manner.

4.3 BUNDLES RECEIVED FROM OTHER NODES

4.3.1 GENERAL

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

4.3.2 RECEIVING BCB BLOCKS

4.3.2.1 If the receiving node is the destination of the bundle, the node must decrypt any BCBs remaining in the bundle.

4.3.2.2 If the receiving node is not the destination of the bundle, the node may decrypt the BCB if directed to do so as a matter of security policy.

4.3.2.3 If the relevant parts of an encrypted payload cannot be decrypted (i.e., the decryption key cannot be deduced or decryption fails), then the bundle must be discarded and processed no further.

4.3.2.4 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.

4.3.2.5 If either 4.3.2.3 or 4.3.2.4 applies, requested status reports may be generated to reflect bundle or block deletion.

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

4.3.2.7 If a BCB contains multiple security targets, all security targets must be processed if the BCB is processed by the node.

NOTE – The effect of this is to be the same as if each security target had been represented by an individual BCB with a single security target.

4.3.3 RECEIVING BIB BLOCKS

4.3.3.1 If the bundle has a BIB and the receiving node is the destination for the bundle, the node must verify the security target in accordance with the cipher suite specification.

4.3.3.2 If the bundle has a BIB and the receiving node is not the bundle destination, the receiving node may attempt to verify the value in the security result field. If a payload integrity check fails at a waypoint, it is recommended that the bundle be processed in the same way as if the check had failed at the destination.

4.3.3.3 A BIB must not be processed if the security target of the BIB is also the security target of a BCB in the bundle.

NOTE – Given the order of operations mandated by this specification, when both a BIB and a BCB share a security target, 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.

4.3.3.4 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.

NOTE – Security policy is beyond the scope of this document.

4.3.3.5 If the security target is the payload or primary block and is removed as a function of security policy, the bundle may be discarded.

4.3.3.6 If a BIB check fails, the security target shall be processed according to the security policy.

4.3.3.7 If a BIB check fails, a bundle status report indicating the failure may be generated.

4.3.3.8 If a BIB contains multiple security targets, all security targets must be processed if the BIB is processed by the node.

NOTE – The effect of this is to be the same as if each security target had been represented by an individual BIB with a single security target.

4.4 BUNDLE FRAGMENTATION AND REASSEMBLY

4.4.1 If it is necessary for a node to fragment a bundle, and security services have been applied to that bundle, then the fragmentation rules described in the BP specification (reference [1]) must be followed.

NOTE – Only the payload may be fragmented; security blocks, like all extension blocks, can never be fragmented.

4.4.2 Integrity and confidentiality operations are not to be applied to a bundle representing a fragment (i.e., a bundle whose 'bundle is a fragment' flag is set in the Bundle Processing Control Flags field).

NOTE – Specifically, a BCB or BIB must not be added to a bundle fragment, even if the security target of the security block is not the payload. When integrity and confidentiality must be applied to a fragment, it is recommended that encapsulation be used instead.

4.5 PAYLOAD-LEVEL SECURITY

4.5.1 The payload of a bundle may be protected via the utilization of Cryptographic Message Syntax (CMS).

NOTE – It is expected that this mechanism would be used only in terrestrial networks and not for bundles traversing space links.

4.5.2 If CMS is implemented, it must only exist within the payload block, as opposed to within the protocol-level bundle security implementation. (A reference with regards to implementation guidelines is provided in 4.5.3.)

4.5.3 The payload block must exclusively contain a CMS envelope, which must contain valid CMS data, as defined in RFC 5652 (reference [6]), and encoded in X.690 BER or DER encoding.

4.5.4 The block processing control flags value may be set to whatever values are required by local policy.

4.5.5 A CMS envelope may include multiple CMS security operations within the envelope, to allow for multiple nested operations to be performed on payload data.

4.5.6 The security services provided by CMS shall be considered successful if all services in the CMS envelope are validated. If any single service encapsulated in the envelope fails to validate, then the entire envelope must be considered to have failed to validate and must be disposed in accordance with security policy.

NOTE – Table 4-1 is a CMS Example. In this example the payload block encapsulates a second bundle, wrapping it within CMS. The ultimate destination of the outer block is responsible for the decapsulation of the CMS data prior to retransmission of the inner block.

Table 4-1: Sample Usage of Bundle-in-Bundle Encapsulation with CMS Data

<table>
<tr><th>Block in Bundle</th><th>ID</th></tr>
<tr><td>Primary Block</td><td>0x00</td></tr>
<tr><td>Payload Block

Signed-Data {
Digest Algorithm(s),
Enveloped-Data {
<table><tr><th>Block in Bundle</th><th>ID</th></tr><tr><td>Primary Block</td><td>0x00</td></tr><tr><td>Payload Block</td><td>0x01</td></tr></table>},
Encrypted Encryption Key(s)
},
Signature(s) and Certificate Chain(s)
}</td><td>0x01</td></tr>
</table>

5 POLICY CONSIDERATIONS

5.1 OVERVIEW

When implementing SBSP, 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.

5.2 KEY POLICIES

5.2.1 If a bundle is received that contains more than one instance of a given security operation, in violation of SBSP, 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.

5.2.2 BPAs in the network must determine 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.

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

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

ANNEX A

PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT (PICS) PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of [Specification]. The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation claiming conformance must satisfy the mandatory requirements referenced in the RL.

A1.2 ABBREVIATIONS AND CONVENTIONS

The RL consists of information in tabular form. The status of features is indicated using the abbreviations and conventions described below.

Item Column

The item column contains sequential numbers for items in the table.

Feature Column

The feature column contains a brief descriptive name for a feature. It implicitly means ‘Is this feature supported by the implementation?’

Status Column

The status column uses the following notations:

- M mandatory;
- O optional;
- C conditional;
- X prohibited;
- I out of scope;
- N/A not applicable.

Support Column Symbols

The support column is to be used by the implementer to state whether a feature is supported by entering Y, N, or N/A, indicating:

Y Yes, supported by the implementation.

N No, not supported by the implementation.

N/A Not applicable.

The support column should also be used, when appropriate, to enter values supported for a given capability.

A1.3 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called an ICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference X*i*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

A2 PICS PROFORMA FOR CCSDS STREAMLINED BUNDLE SECURITY PROTOCOL

A2.1 GENERAL INFORMATION

A2.1.1 Identification of ICS

Date of Statement (DD/MM/YYYY)	
ICS serial number	
System Conformance statement cross-reference	

A2.1.2 Identification of Implementation Under Test

Implementation Name	
Implementation Version	
Special Configuration	
Other Information	

A2.1.3 Identification of Supplier

Supplier	
Contact Point for Queries	
Implementation Name(s) and Versions	
Other information necessary for full identification, e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)	

A2.1.4 Identification of Specification

[CCSDS Document Number]	
Have any exceptions been required? NOTE – A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming.	Yes [ ] No [ ]

A2.2 REQUIREMENTS LIST

Classes				
Item	Description	Reference	Status	Support
1	SBSP Support	3.1.1	M	

PDUs								
Item	PDU	Ref.	Sender End-System		Receiver End-System		Relay	
			Status	Support	Status	Support	Status	Support
2	CBF	3.2	M		M			
3	GSBS	3.3	M		M		C1	
4	BIB	3.4	O.1		O.1		O	
5	BCB	3.5	O.1		O.1		N/A	

O.1: At least one of these options must be supported in order to claim compliance with this specification.

C1: If intermediate BIB verification is supported on a relay then the relay must be able to parse the Generic Security Block Structure.

Parameters of GSBS-PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
6	Number of security targets	3.3.2.1	M		>= 1	
7	Optional ciphersuite parameters	3.3.2.5, 3.8	M		0-1, 3-5, 7-8	

Security Processing				
Item	Description	Reference	Status	Support
8	Canonicalization	4.2	M	
9	BCB Decryption	4.3.2.1, 4.3.2.2, 4.3.2.3, 4.3.2.4, 4.3.2.6, 4.3.2.7	M	
10	BCB Status Reports	4.3.2.5	O	
11	BIB Reception	4.3.3.1, 4.3.3.3, 4.3.3.4, 4.3.3.6, 4.3.3.8	M	
12	BIB Optional Checking	4.3.3.2	O	
13	BIB Discard if no Primary Block Left	4.3.3.5	O	
14	BIB Send Status Report on Failure	4.3.3.7	O	

Fragmentation and Reassembly				
Item	Description	Reference	Status	Support
15	Fragmentation	4.4.1, 4.4.2	M	

Payload-Level Security				
Item	Description	Reference	Status	Support
16	Use of CMS on payload	4.5.1	O	
16.1	CMS format and processing	4.5.2, 4.5.3, 4.5.4, 4.5.5, 4.5.6	c:m	

Policy Considerations					
Item	Description	Reference	Status	Options	Treatment
17	Dealing with violations	5.2.1	M	Discard bundle; discard affected block; favor one security option (specify method of determination).	
18	Security operations to apply	5.2.2	M	Specify security operations to apply and criteria for determining them.	
19	Conflict resolution	5.2.3	M	Discard the bundle; discard the security target; replace the security operation; other (specify)	

ANNEX B

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

B1 SECURITY CONSIDERATIONS

B1.1 GENERAL

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

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

The threat model described here is assumed to have a set of capabilities identical to those described by the Internet Threat Model in RFC 3552 (reference [D1]), but the SBSP threat model is scoped to illustrate threats specific to SBSP operating within DTN environments and therefore focuses on Man-In-The-Middle (MITM) attackers.

B1.2 SECURITY CONCERNS WITH RESPECT TO THE CCSDS DOCUMENT

B1.2.1 Data Privacy

Data privacy is provided by the BCB feature of this specification.

B1.2.2 Data Integrity

Data integrity is provided by the BIB feature of this specification.

B1.2.3 Authentication of Communicating Entities

Authentication between communicating entities may be accomplished using the BIB feature of this specification.

B1.2.4 Control of Access to Resources

Resource access controls are not directly addressed by this specification, although implementations may use features of this specification, such as BIBs, to authenticate nodes as part of a resource access control scheme.

B1.2.5 Availability of Resources

No mechanisms are defined in this specification to verify or assist with the verification of availability of resources.

B1.2.6 Auditing of Resource Usage

No mechanisms are defined in this specification to audit or assist with the auditing of resource usage by the protocol.

B1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

B1.3.1 Attacker Capabilities and Objectives

SBSP 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 annex subsection, any MITM node is assumed effectively to be security-aware even if it does not implement the SBSP protocol. There are three classes of MITM nodes, which are differentiated based on their access to cryptographic material:

- Unprivileged Node: Mallory has not been provisioned within the secure environment and has access only to cryptographic material that has been publicly shared.
- Legitimate Node: Mallory is within the secure environment and therefore has access to cryptographic material that has been provisioned to Mallory (i.e., K_M) as well as material which has been publicly shared.
- Privileged Node: Mallory is a privileged node within the secure environment and therefore has access to cryptographic material that 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.

Mallory's operating as a privileged node would be tantamount to compromise; SBSP does not provide mechanisms to detect or remove Mallory from the DTN or SBSP secure environment. It is up to the SBSP implementer or the underlying cryptographic mechanisms to provide appropriate capabilities if they are needed. It should also be noted that if the implementation of SBSP 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.

B1.3.2 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 SBSP provides against this action is the BCB, which encrypts the contents of its security target, providing confidentiality of the data. Of course, it should be assumed that Mallory is able to attempt offline recovery of encrypted data, so the cryptographic mechanisms selected to protect the data should provide a suitable level of protection.

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

B1.3.3 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-SBSP data such as the primary block, payload blocks, or block processing control flags as defined in reference [1]. 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 SBSP, 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 SBSP security operations are implemented by placing blocks in a bundle, there is no in-band mechanism for detecting or correcting certain cases where Mallory removes blocks from a bundle. If Mallory removes a BCB block, but keeps the security target, the security target remains encrypted and there is a possibility that there may no longer be sufficient information to decrypt the block at its destination. If Mallory removes both a BCB (or BIB) and its security target there is no evidence left in the bundle of the security operation. Similarly, if Mallory removes the BIB but not the security target there is no evidence left in the bundle of the security operation. To obviate each of these cases, at endpoints in the network, the implementation of SBSP must be combined with policy configuration that describes the expected and required security operations that must be applied on transmission and are expected to be present on receipt. This or other similar out-of-band information is required to correct for removal of security information in the bundle.

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

B1.3.4 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 rerouting the packets throughout the network are essentially a special case of the modification attacks described in this annex subsection, where the attacker is modifying fields within the primary block of the bundle. Given that SBSP cannot encrypt the contents of the primary block, 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 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 SBSP, their definition is beyond the scope of this specification.

B1.3.5 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, SBSP, 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.

SBSP 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.

B1.4 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

When not applying the Streamlined Bundle Security Protocol, the system must rely on security measures provided at the convergence layer adapter interfaces. For space applications these may be non-existent or merely physical because of the lack of integration between payload and ground system interfaces. If no security is applied at the BP or lower layers, then applications may be open to MITM attacks, replay attacks, or a general loss of integrity of transported bundles.

B2 SANA CONSIDERATIONS

B2.1 GENERAL

The recommendations of this document request SANA to create the following registries:

B2.2 BUNDLE BLOCK TYPES REGISTRY

B2.2.1 Registry Name: Bundle Block Types Registry.

B2.2.2 Registry Purpose: The registry named 'Bundle Block Types' documents the type values to be used to identify blocks in the CCSDS Bundle Protocol.

B2.2.3 Registry Structure: 3 Columns: Value; Description; Reference.

B2.2.4 The Registry Data Types shall be:

- Value: integer; allowed values 0—255 inclusive;
- Description: A string of text describing the block type;
- Reference: A string of text containing a reference to the document that defines the block type.

B2.2.5 Registry Category: WG/Local.

B2.2.6 Review Authority: SIS Area or their designee.

B2.2.7 Registration Procedure: Change Requires a CCSDS approved document.

B2.2.8 Initial Registry Values: The initial registry should be filled with the following values:

Value	Description	Reference
0	Reserved	Ref: BP for CCSDS
1	Payload block	Ref: BP for CCSDS
2	Available for allocation / unassigned	
3	Bundle Integrity Block	Ref: CCSDS SBSP
4	Bundle Confidentiality Block	Ref: CCSDS SBSP
5—9	Available for allocation / unassigned	
10	Custody Transfer Extension Block	Ref: BP for CCSDS
11—18	Available for allocation / unassigned	
19	Enhanced Class of Service Block	Ref: BP for CCSDS
20—191	Available for allocation / unassigned	
192—255	Private / Experimental	

B2.3 CCSDS SBSP CIPHER SUITE FLAGS REGISTRY

B2.3.1 Registry Name: CCSDS SBSP Cipher Suite Flags.

B2.3.2 Registry Purpose: The registry named ‘Cipher Suite Flags’ documents the flag values to be used to identify features / capabilities of particular CCSDS SBSP Cipher Suites. The value of the cipher suite flags parameter (the logical AND of all the is encoded in a Self-Defined Numeric Value (SDNV) and included in the Generic Security Block Structure used to construct Bundle Integrity and Bundle Authentication Blocks.

B2.3.3 Registry Structure: 3 Columns: Value; Description; Reference.

B2.3.4 The Registry Data Types shall be:

- Value: integer interpreted as the bit position of a Boolean flag which if set to 1 signals the presence / implementation of a particular cipher suite capability or feature.
- Description: A string of text describing the capability or feature indicated by the flag.

B2.3.5 Reference: Free text containing a reference to the document that defines the block type.

B2.3.6 Registry Category: WG/Local.

B2.3.7 Review Authority: SIS Area or their designee.

B2.3.8 Registration Procedure: Change Requires a CCSDS approved document.

B2.3.9 Initial Registry Values: The initial registry should be filled with the following values:

Value (Bit Position from least significant bit)	Description	
0	Block contains parameters	Ref: CCSDS SBSP
>=1	Available for allocation / unassigned	Ref: CCSDS SBSP

B2.4 CCSDS SBSP CIPHER SUITE PARAMETERS AND RESULTS TYPE REGISTRY

B2.4.1 Registry Name: CCSDS SBSP Cipher Suite Parameters and Results Type Registry.

B2.4.2 Registry Purpose: The registry named 'CCSDS SBSP Cipher Suite Parameters and Results Type Registry' documents the allowable values that may be used to identify cipher suite specific parameters in the Cipher Suite Parameters Field of a security block in the CCSDS SBSP specification. Cipher suite parameters are carried as part of the Cipher Suite Parameters Data field in the Generic Security Block Structure used to construct Bundle Integrity and Bundle Authentication Blocks.

B2.4.3 Registry Structure: 3 Columns: Value; Description; Reference.

B2.4.4 Registry Data Types:

- Value: integer that is used in the Cipher Suite Parameters Data Field to identify the type of cipher suite parameter data the field contains.
- Description: A string of text describing the capability or feature indicated by the flag.

B2.4.5 Reference: Free text containing a reference to the document that defines the cipher suite parameter data.

B2.4.6 Registry Category: WG/Local.

B2.4.7 Review Authority: SIS Area or their designee.

B2.4.8 Registration Procedure: Change Requires a CCSDS approved document.

B2.4.9 Initial Registry Values: The initial registry should be filled with the following values.

Value	Description	Reference
0	Reserved	Ref: CCSDS SBSP
1	Initialization Vector	Ref: CCSDS SBSP
2	Reserved	Ref: CCSDS SBSP
3	Key Information	Ref: CCSDS SBSP
4	Content-Range	Ref: CCSDS SBSP
5	Integrity Signature	Ref: CCSDS SBSP
6	Available for allocation / unassigned	Ref: CCSDS SBSP
7	Salt	Ref: CCSDS SBSP
8	BCB Integrity Check Value	Ref: CCSDS SBSP
9—191	Available for allocation / unassigned	Ref: CCSDS SBSP
192—250	Private / experimental use	Ref: CCSDS SBSP
251—255	Reserved	Ref: CCSDS SBSP

B3 PATENT CONSIDERATIONS

There are no known patents covering the Streamlined Bundle Security Protocol as described in this document and its normative references.

ANNEX C

CIPHERSUITE AUTHORSHIP CONSIDERATIONS

(INFORMATIVE)

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

- 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.

- One-Way Traffic: Depending on the application environment, it is possible that only a one-way connection may exist between two endpoints; 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.

- 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.

ANNEX D

INFORMATIVE REFERENCES

(INFORMATIVE)

[D1] E. Rescorla and B. Korver. *Guidelines for Writing RFC Text on Security Considerations*. RFC 3552. Reston, Virginia: ISOC, July 2003.

ANNEX E

ABBREVIATIONS

(INFORMATIVE)

Term	Meaning
BCB	Bundle confidentiality block
BER	Basic Encoding Rules
BIB	Bundle integrity block
BP	Bundle Protocol
BPA	Bundle Protocol agent
CBF	Canonical Bundle Block Format
CMS	cryptographic message syntax
COS	class of service
DER	Distinguished Encoding Rules
DTN	Delay-Tolerant Networking
EID	endpoint identifier
GSBS	generic security block structure
HTTP	Hypertext Transfer Protocol
ICV	integrity check value
IETF	Internet Engineering Task Force
IRI	Internationalized Resource Identifier
IV	initialization vector
MITM	man in the middle
RFC	Request for Comments
RL	requirements list
SBSP	Streamlined Bundle Security Protocol
SDNV	self-delimiting numeric value
SRR	status report request
SSP	scheme-specific part
TLV	type-length-value
URI	Uniform Resource Identifier
URL	Uniform Resource Locator