BPSoC, Interoperability Cipher Suites

IETF-103

Edward Birrane
Edward.Birrane@jhuapl.edu
443-778-7423
Overview

- **BPSeq**
  - Updates from Last DTNWG.
  - Updates from CCSDS review.
  - Discussion points

- **Interoperability Cipher Suites**
  - Updates

- **Open questions**
1. Do we need to add a graphic to show multiple security sources?
   - No

2. May certain cipher suites alter the size of the target block’s data fields?
   - Yes.

3. Do we need language to explicitly allow cipher suites to remove blocks from a bundle?
   - No. BpSec should not disallow it. Individual cipher suite documents will describe how and when this would occur.

4. Do we require that a single node add *either* a BCB *or* a BIB for a target, but not both?
   - No need to require this, but it is a recommended practice.
Block-type specific fields

- BpBis will be updated to always represent these as CBOR byte strings.
- This will allow a common handling mechanism when converting plaintext to cipher text.
  - The CBOR byte string length bytes will NOT be considered part of the data sent to the cipher suite for processing. They do not represent user data.
  - The entire plaintext CBOR byte string (including length) will be discarded and replaced by the CBOR encoded cipher text.
- Cipher suites MAY generate cipher text that is not the same size as the original plain text.
Additional minor comments
- Most review requests relate to cipher suite selection and background context material that is not relevant to this document and specific to CCSDS.

Significant comment: Security Associations
- Currently: each BIB or BCB is associated with a cipher suite Id and parms.
- Could also associate BIBs and BCB with other kinds of use cases and events
- Consider compressing ALL non-security-result information into a single security association identifier.
  - The same bundle could define the security association.
  - Some other bundle could define the security association.
  - Some out-of-band mechanism could define the security association.

Does not necessarily change information from original BP Sec.
- Consolidates cipher suite parms into a single place and not per-BIB/BCB.
- Allows an easier way to talk about rekey, out-of-band config, etc…
- Familiar terminology from IKE, etc…
What is a Security Association?

- **An Identifier**
  - The Security Association Id (SAID) is a scoped one-way association.
  - It MUST be unique within its scope, which is:
    - A set of block types from a set of sources to a set of destinations.
    - For example: All Payload Blocks from EID1 to EID2

- **A definition block**
  - An association is an identified set of security-related information
    - Extract existing security-related information from BIB/BCB and place it in an association block.
    - Place scope and SAID information in that block.
  - BIB and BCB blocks now reference an SAID
    - Individual BIB/BCB blocks no longer need to carry cipher suite information and parms.
  - Added benefits
    - Some security policy provided by scoping rules.
    - Security associations can be communicated out-of-band. This was also the intent for cipher suite IDs, but security association IDs is a cleaner way of reasoning about this.
Security Association Block

**Original Block Layout**
- BIB or BCB block 1
  - Cipher Suite ID
  - Security Source
  - Cipher Suite Parameters
  - Security Results

**Proposed Block Layout**
- Security Association Block
  - Security Association ID
    - EID Scope
    - Block Type Scope
    - Cipher Suite ID
    - Security Source
    - Security Association Parameters

**Original Block Layout**
- BIB or BCB block 2
  - Cipher Suite ID
  - Security Source
  - Cipher Suite Parameters
  - Security Results

- BIB or BCB block 1
  - Security Association ID
  - Security Results

- BIB or BCB block 2
  - Security Association ID
  - Security Results
Security Association Block Specifics

- **SAID**
  - CBOR Unsigned Integer

- **Security Association Flags**
  - CBOR Unsigned Integer. Determines inclusion of optional fields.

- **EID Scope (optional) – Which destinations the SA applies to.**
  - If missing, SA can apply to any destinations (pursuant to policy)
  - CBOR Array with each element an encoded EID (pursuant to BpBis encoding rules).

- **Block Type Scope (optional) – Which block types the SA applies to.**
  - If missing, SA can apply to any block types (pursuant to policy)
  - CBOR Array with each element an encoded block type (pursuant to BpBis encoding rules).

- **Cipher Suite Id, Security Source, Association Parameters**
  - All optional.
  - These fields are defined as they were for BIB/BCB.
  - Just moved them from BIB/BCB to the SAB.
Proposed Changes to BIB/BCB

- Replace Cipher Suite ID with Security Association ID
- Security Association Flags replace Cipher Suite Flags
  - Currently only 1 field defined: Security Source
- Remove Cipher Suite Parameters from BIB/BCB
  - Security Association Block (or out of band mechanism) captures cipher suite parms.
- Unchanged Items
  - Security Targets
  - Security Source
  - Security Results
**Interoperability Cipher Suites**

**Reminder**
- **BIB-HMAC256-SHA256**
  - The integrity cipher suite provides a signed hash over the security target based on the use of the SHA-256 message digest algorithm [RFC4634] combined with HMAC [RFC2104] with a 256 bit truncation length. This formulation is based on the HMAC 256/256 algorithm defined in [COSE] Table 7: HMAC Algorithm Values.
- **BCB-AES-GCM-256**
  - The confidentiality cipher suite provides cipher text to replace the data contents of the target block using the AES cipher operating in GCM mode [AES-GCM]. This formulation is based on the A256GCM algorithm defined in [COSE] Table 9: Algorithm Value for AES-GCM.

**Changes**
- Updated to explain CBOR byte string updates discussed earlier.
Questions

- Do we want to use security associations?
- Do we want to use them as described in this document?
- Are there any proposed changes to the interop cipher suites?
- What are the next steps?
Backup Material from Last Time
A bundle might not contain all of its security at creation.

Nodes, by security policy, may encrypt/decrypt a payload or extension blocks.

Destinations may not know extra security occurred, but may need to see source-signed material.
Context: We have a bundle with a BIB providing plain-text signatures on several blocks.

- This will happen when signatures are added by same node, with same key info.
- Prevents having 3 BIBs in the bundle (and thus, having redundant info).
Later, another nodes wants to encrypt Target 2.
- By BPSec it MUST encrypt block-specific fields of target 2 AND BIB signature on target 2.

We cannot simply encrypt the BIB itself
- We would hide the plain-text signatures for targets 1 and 3.

We cannot simply encrypt pieces of the BIB
- In BIB structure, information for target 2 would exist in multiple byte ranges. This adds a lot of processing complexity to support.
Proposed solution

- Split the BIB.
  - *BIB1 contains the original signatures NOT being encrypted*
  - *BIB2 contains any signature that must be encrypted.*
- The original conditions that justified grouping the targets into a single BIB no longer apply.
- Processing can now continue without issue.
Simple BPSeq Example

Single Integrity Block holds signatures for multiple other blocks.

Confidentiality block encrypts its target and holds a signature on the encrypted target.

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

Figure 3: Security at Bundle Creation
Waypoint Encrypts Block B5, B6.

Figure 3: Security at Bundle Creation

1. Split BIB

2. Encrypt

3. New BCB

Figure 4: Security At Bundle Forwarding
Bpbis Consideration: Encoding Block Data

- BPBis block captured as a CBOR array of 5-6 items:
  - \{type, id, flags, crc_type, type-specific-fields, crc (opt)\}
  - Type-specific-fields have no mandated CBOR encoding
    - *Except for payload block, which must be BYTE STRING.*

- Ex: block with 3 fields with values 0x1, 0x2, 0x3
  - Encoded as a CBOR byte string (h’010203’)
    - 0x43010203
    - 4 bytes...
  - Encoded as a CBOR array: [1,2,3]
    - 0x83010203

Is it secure to “parse” the plain-text block-type-specific data to determine it is a CBOR byte string?
Length-Encoding Cipher-Text

N bytes

K bytes

N bytes