BPSec,
Interoperabilty Cipher Suites

IETF-102

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

JOHNS HOPKINS UNIVERSITY
Applied Physics Laboratory

Overview

- BPSec
  - Security AD review comments and updates
  - Other review comments and updates
  - Changes by Section
  - Discussion points
- Interoperability Cipher Suites
  - Updates
  - Open questions
- BpBis Request

BPSec Review Comments from IETF

- Security Area “Early Review” Disposition: Almost-Ready
  - Three comments with the document
  - Comment 1: Encryption and Integrity Protection
  - Comment 2: Clarify some use of RFC 2119 wordings
  - Comments 3: Security Considerations

- Next 3 slides address comments
  - I summarize major portions of the comments
  - Overview of what changed in the spec.
    - *Few spec changes*
    - *More clarification*

- Full Comments can be found at:
  - https://www.ietf.org/mail-archive/web/secdir/current/msg08138.html

CMT 1: Encryption and Integrity Protection

"… the BCB does not also provide integrity protection of its ciphertext since the draft says that, while multiple security operations on the same block are invalid, doing integrity protection and confidentiality on the same block is valid. This opens up some insecure options that don't need to be allowed, like integrity protect then encrypt, or integrity protect and encrypt. I think it would be a good idea to ***mandate*** *that whatever ciphersuite is used for a BCB (again, the draft does not specify ciphersuites) that it provides* ***authenticated encryption****. Then update the uniqueness requirement in section 3.2.*

… My suggestion to use an AEAD cipher seems to conflict somewhat with the fragmentation/reassembly text which says that application of a confidentiality cipher suite MUST NOT alter the size of the payload. That is going to have to be reconciled somehow. ***This document should not allow anything other than encrypt-then-authenticate*** *(and it should do so by mandating AEAD ciphers) and if that requires some rewrite of the fragmentation/reassembly text then so be it."*

- Mandate BCBs use AEAD cipher suites
- Clarify text on what the BIB block actually does
- Clarify rationale for multiple security sources
- Revisit impact of altering payload size

CMT 2: Clarify some use of RFC 2119 wordings

“*Section 2.2: "A bundle MAY have multiple security blocks and these blocks MAY have different security sources." … What I'm reading is an admonition to not assume uniformity in bundles, which seems like an important statement that is the opposite of the literal MAY text.*

Section 3.3: "A set of security operations may be represented by a single security block if and only if the following conditions are true...." That sounds kind of normative. Do the authors mean "A set of security operations SHALL be represented by a single security block...."?

Regarding the optional "Security Source" in the Abstract Security Block in section 3.6: "If the security source field is not present then the source MAY be inferred from other information...." And that means I can choose to not implement this optional inference. In which case, what do I do? I think some instruction to implementers is needed but I'm not sure what it is.

Basically, I think the whole document should be searched for "may" (case insensitively) and each instance looked at closely.”

- Made changes to “MAY” vs ”may” and other language.
  - Additional review here appreciated.

CMT 3: Security Considerations

“The security considerations are thorough and well done although the first three paragraphs in section 8 seem to boil down to the fact that the DTN is assumed to be completely under the control of an attacker. I think that's all that needs to be said there.“

- Migrated one paragraph to the introduction
- Added observation that at least some nodes in a DTN should be assumed to be compromised from a BPSec point of view.

Other Review Comments

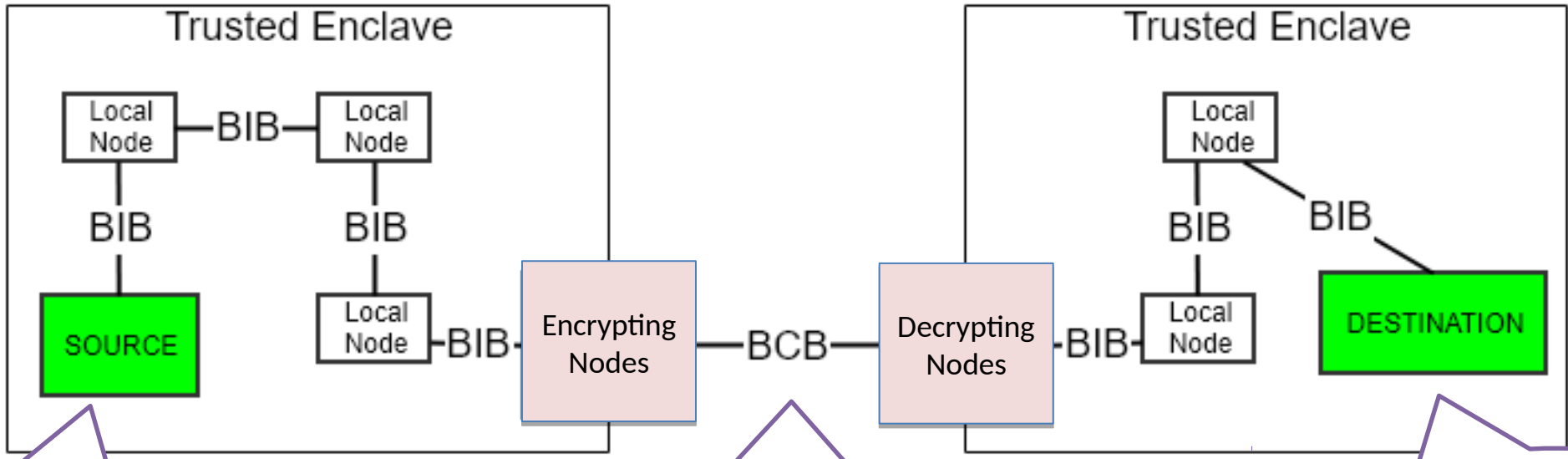
- CCSDS SEA-SEC working group review
  - Reviewed earlier version of BP security standards
  - Reviewed comments that would apply to BPSec.
- Minor text updates
  - Clarify some text in the introduction.
- Update citations
- Terminology change request
  - Extension blocks define “security target” but being a target is usually a bad thing.
  - Request to change the term “security target” to something else.
    - *“Security Recipient”?*

BPSec Changes from -06 to -07

- Section 1: Introduction
  - Updated citations
  - Clarified integrity is for plain-text only.
  - Clarified confidentiality includes signature on the cipher text.
  - Pulled scoping paragraph from the security considerations section to this section for context.

- Section 2: Design Decisions
  - Clarified rationale for multiple security sources (graphic on next slide)
  - Corrected some RFC2119 language

BPSec Ex: Multiple Security Sources



A bundle might not contain all of its security at creation.

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

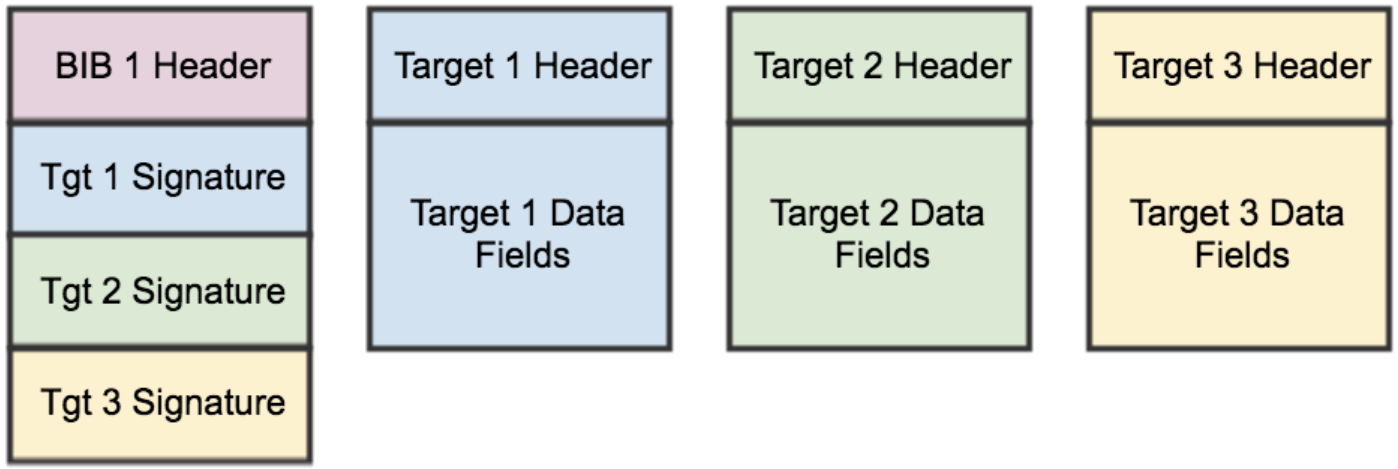
Destinations may not know extra security occurred, but may need to see source-signed material.

BPSec Changes from -06 to -07

- Section 3: Security Blocks
  - Updated citations and corrected from RFC2119 language.
  - Clarified that BIB applies to plain-text and BCB authenticates cipher text.
  - Added requirement that BCB cipher suites be AEAD cipher suites
  - Clarified actions to take is optional security source field is not present.
    - *“If the security source field is not present then the source MUST be inferred from other information, such as the bundle source, previous hop, or other values defined by security policy.”*
  - Removed requirement that BCB not alter the size of the target block data.
    - *This is a matter of cipher suite selection by networks.*
    - *Any cipher suite could choose to place overflow in BCB security result fields to avoid changing the length of the target block data fields.*
    - *Altering the size of a target block, to include the payload block, does not break anything in the specification anymore.*
    - *Adding a BCB changes byte offsets of blocks in the bundle regardless.*
  - Clarify processing in corner case with integrity block multiplicity and encryption (next slide)

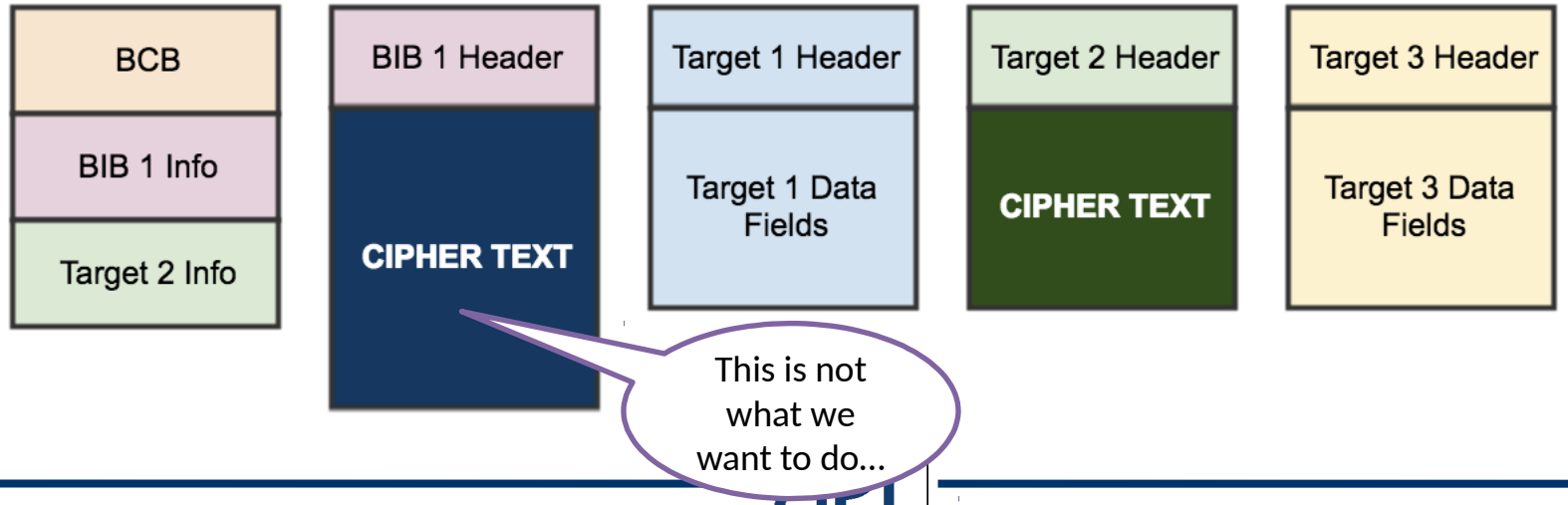
Multiple Integrity W/ Encryption (1/3)

- 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).



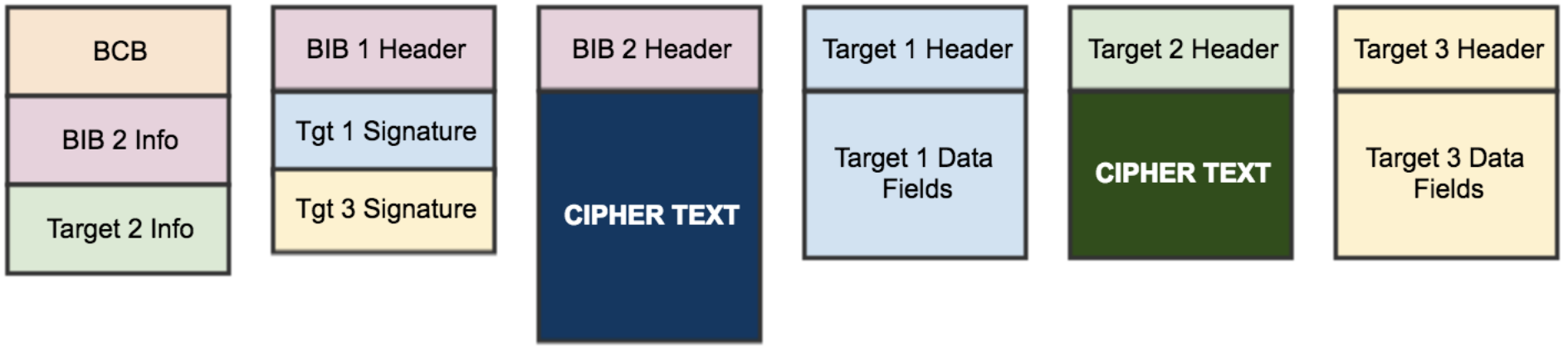
Multiple Integrity W/ Encryption (2/3)

- 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



Multiple Integrity W/ Encryption (3/3)

- 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 BPSec Example

Single Integrity Block holds signatures for multiple other blocks.

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

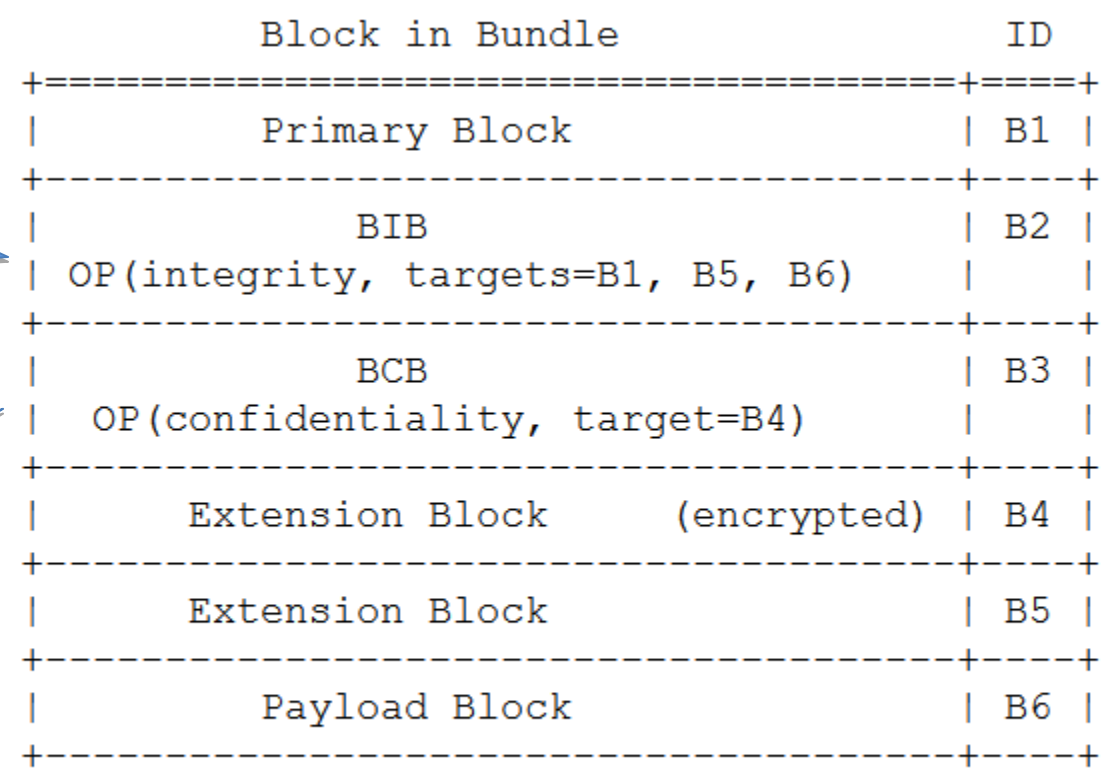


Figure 3: Security at Bundle Creation

Waypoint Encrypts Block B5, B6.

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

Figure 3: Security at Bundle Creation

1. Split BIB

2. Encrypt

3. New BCB

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

Figure 4: Security At Bundle Forwarding

BPSec Changes from -06 to -07

- Section 4: Canonical Forms
  - Updated citations and corrected some RFC2119 language.
- Section 5: Security Processing
  - Updated citations and corrected from RFC2119 language.
  - Relaxed processing language on when to process BCB and BIB
    - *When BCB and BIB share a target, BCB MUST be processed first*
    - *Replaced text that said “all BCBs must be processed first”.*
  - Clarified that “decryption failure” means failure to authenticate cipher text.
- Section 6: Key Management
  - No Changes

BPSec Changes from -06 to -07

- Section 7: Security Policy Considerations
  - Added term “plain-text integrity” rather than “integrity” to clarify some processing
  - Added caveat that BCBs should be able to change target block size, but this must be made in consideration of other bundle processing in the network.
  - Added instruction that cipher suites must detail how extra information (cipher text bytes in excess of plain-text size and other generated data) is added to the BCB and/or the cipher-text itself.
- Section 8: Security Considerations
  - Updated citations and corrected some RFC2119 language.
  - Moved scoping paragraph up to introduction
  - Added sentence that there is an assumption here that some or all of the DTN nodes are under the control of an attacker.

BPSec Changes from -06 to -07

- Section 9: Cipher Suite Authorship Considerations
  - Updated citations and corrected some RFC2119 language.
  - Updated guidance on how to process underflow and overflow when generated cipher text.

- Section 10: Defining Other Security Blocks
  - Updated some RFC2119 language.

BpSec Questions

1. Do we need to add a graphic to should multiple security sources?
   - Recommendation: No

2. May certain cipher suites alter the size of the target block's data fields?
   - May be an issue for some networks, but not others.

3. Do we need language to explicitly allow cipher suites to remove blocks from a bundle?
   - Example: remove a target block on encryption and put it back on decryption.
     Recommendation: BPSec should just not disallow it. It can be documented in a cipher suite.

4. Do we require that a single node add *either* a BCB *or* a BIB for a target, but not both? Currently this is just a recommendation:
   - *"In cases where a security source wishes to calculate both a plain-text integrity mechanism and encrypt a security target, a BCB with a cipher suite that generates such signatures as additional security results SHOULD be used instead."*

Interoperability Cipher Suites

- Required for BPSec publication
  - Similar to request for security associations in CCSDS
  - Interoperability != operational.
- Published draft of BPSec interoperability cipher suites
- Integrity
  - 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.*
- Confidentiality
  - 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.*

Updates

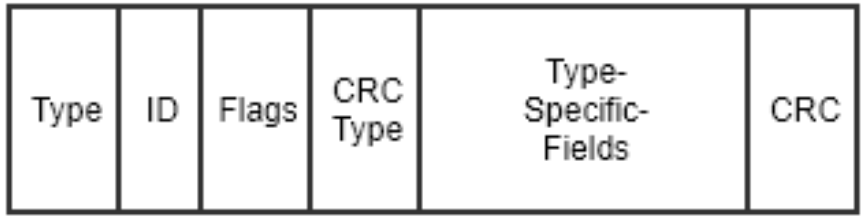
- Section 1: Introduction
  - Explicitly state that these cipher suites generate CBOR-encoded values.
- Section 3: Cipher Suite BIB-HMAC256-SHA256
  - Updated citations for COSE, BPSec
  - Updated terminology to match field named from BPBis.
- Section 4: Cipher Suite BCB-AES-GCM-256
  - Updated citations for COSE, BPSec
  - Updated terminology to match field named from BPBis.
  - Updated BCB suite from AES128 to AES256.
  - Updated encryption/decryption to:
    - *Specify CBOR encoding*
    - *Specify how to handle block-type-specific fields (Byte String or other)*
    - *Updated description of how to handle failed decryption*
  - Added recommendation on not re-using IV values.

To Do

- To Do
  - Update encryption instructions based on:
    - *Resolution of how BpBis captures block-type-specific fields.*
    - *Any additional guidance on allowing block size expansion.*
  - Update security result fields in BCB to capture additional results based on above.

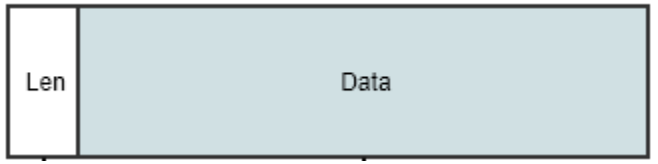
Bpbis Consideration: Encoding Block Data

- 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.*

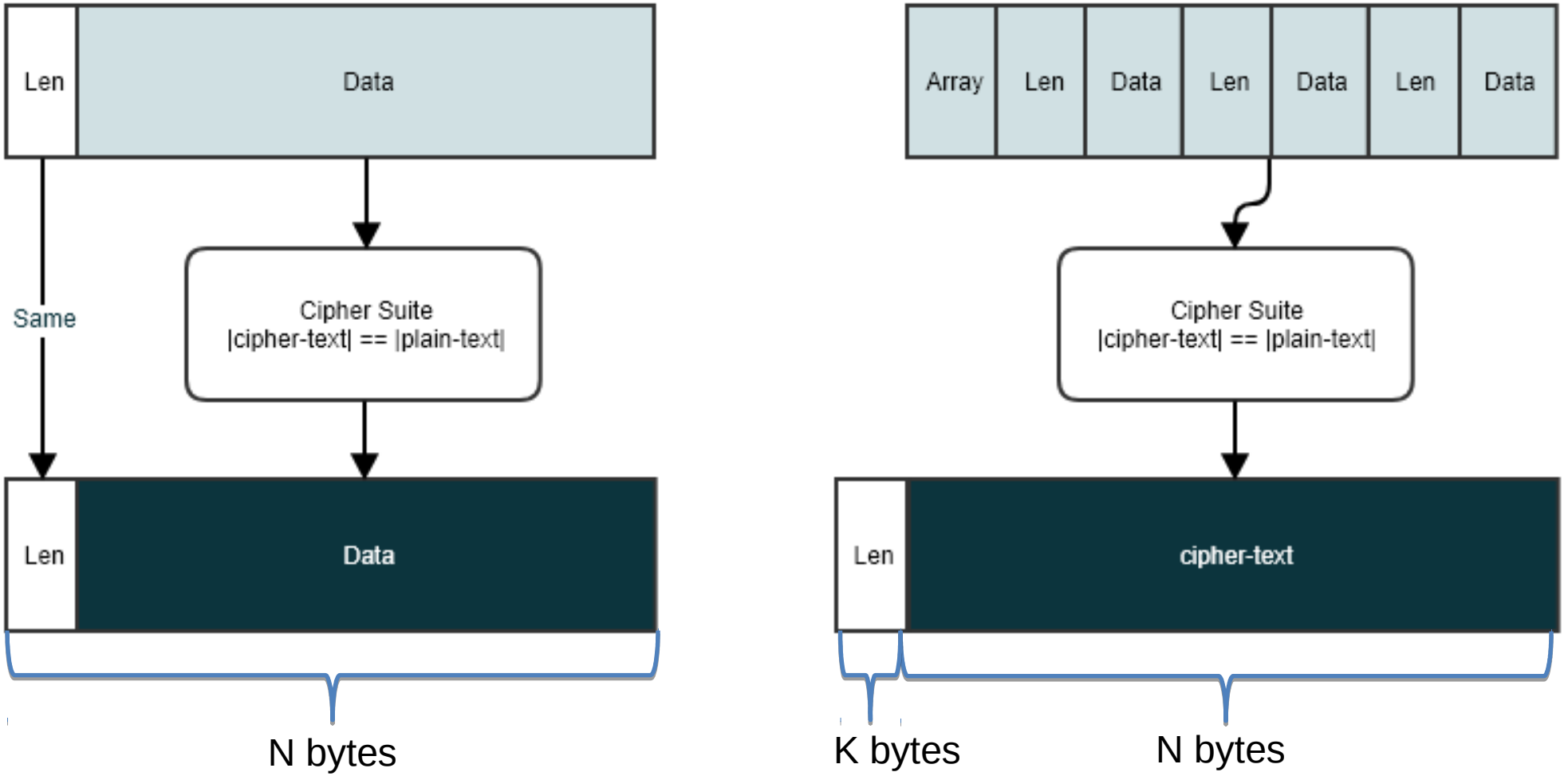


Is it secure to "parse" the plain-text block-type-specific data to determine it is a CBOR byte string?

- 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*



Length-Encoding Cipher-Text



Questions

- Can BpBis always represent block-type-specific fields as Byte Strings?
  - Seems to be the case from list traffic

- Is re-using the “length” field of the CBOR byte string seen as a violation?
  - E.g., is this an example of a cipher suite “processing” plain text?