JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Bundle Protocol Security COSE Context

IETF 112 DTN WG

Brian Sipos
JHU APL

Background

- BPSec and its Default Security Context are usable but intentionally limited in scope:
  - A limited number of symmetric-keyed encryption and MAC algorithms.
  - Defines a variable additional authenticated data (AAD) scope.
  - No explicit key identifiers are available.
- For internet-facing nodes, possibly as subnetwork gateways, there is a need for PKI-integrated security.
  - This was indicated also by SECDIR review of BPSec draft.
- Don’t want to reinvent the wheel, and CBOR Object Signing and Encryption (COSE) already provides syntax and semantics for current and future security algorithms.

Goals for the BPSec COSE Context

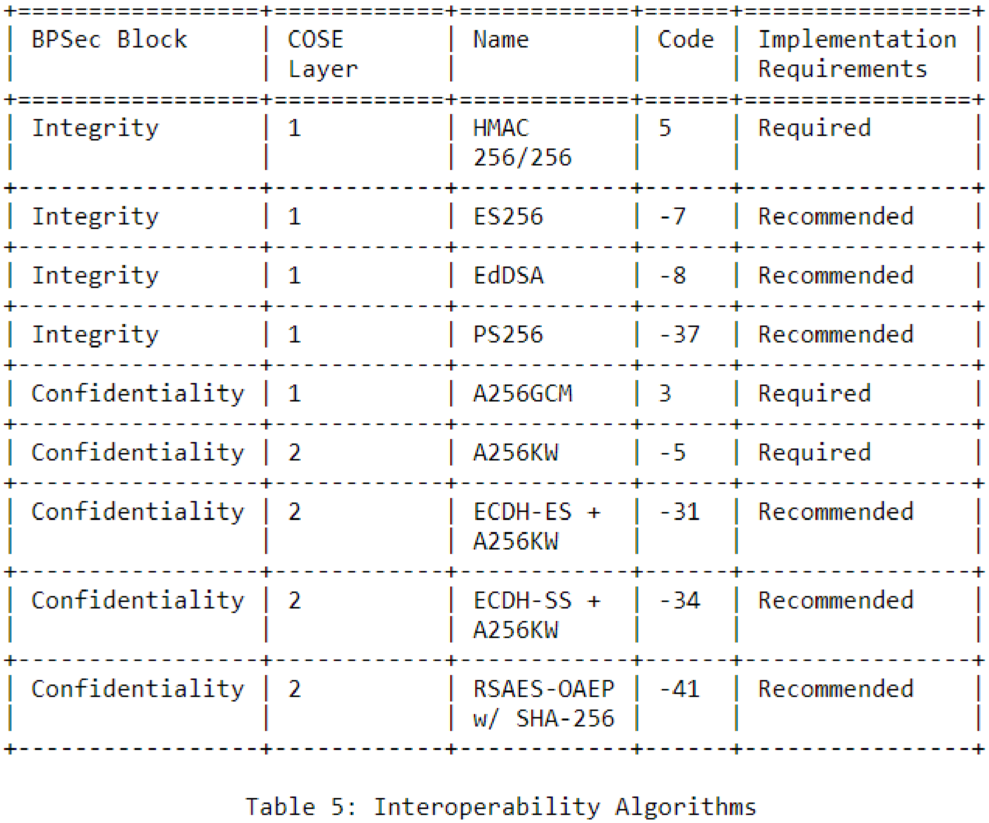
- No not alter BPSec structures or requirements.
  - This is purely an extension within the existing security context mechanism.
- Handle current symmetric-keyed and PKIX algorithms.
  - Leverage existing algorithm definitions.
- Follow algorithm-use and key-use best practices.
  - Avoid key overuse, use random content encryption keys.
- Inherit future gains made by COSE off-the-shelf algorithms.

Proposed COSE Context Contents

- One BPSec context codepoint defined to use in BIB and BCB.
- Parameter and result types defined for each BPSec block type:
  - AAD scope parameter (same semantics as Default SC)
  - De-duplicated COSE header parameters
  - Integrity results (COSE MAC and Signature)
  - Confidentiality results (COSE Encrypt with AEAD)
- Public keys in context parameters to de-duplicate data.
  - Potential future extensions could provide additional supporting data (e.g. OCSP stapling).
- Full COSE messages in each target's result.
  - Reuse COSE message tags as result type codes.
  - Allows an application to use any current or future COSE algorithm types (and combinations).
  - Allows multiple recipients for a single security block (both BIB and BCB).
  - Interoperability requirements are defined in a COSE Profile (next slide).

Interoperability Profile

- Required algorithms for AES-GCM-256, AES key-wrap, and HMAC-SHA2-256.
- Recommended algorithms for Elliptic Curve, Edwards Curve, and RSA signing and key-wrap/key-generation.
- Additional public key material can be included in an “additional header map”, applying to all results in the block.



BPSec Block	COSE Layer	Name	Code	Implementation Requirements
Integrity	1	HMAC 256/256	5	Required
Integrity	1	ES256	-7	Recommended
Integrity	1	EdDSA	-8	Recommended
Integrity	1	PS256	-37	Recommended
Confidentiality	1	A256GCM	3	Required
Confidentiality	2	A256KW	-5	Required
Confidentiality	2	ECDH-ES + A256KW	-31	Recommended
Confidentiality	2	ECDH-SS + A256KW	-34	Recommended
Confidentiality	2	RSAES-OAEP w/ SHA-256	-41	Recommended

Table 5: Interoperability Algorithms

Next Steps

- This is not intended to replace or supersede existing BPSec interoperability contexts (`draft-ietf-dtn-bpsec-interop-sc`).
- The point here is to allow BPSec in a PKIX environment in the very near term.
  - COSE is a known quantity with existing coding and processing tools.
  - Identifying bundle security purpose and validation of a Node ID within a PKIX certificate are already defined in TCPCLv4.
- Some secondary questions remain:
  - E.g. how does a security acceptor handle a BIB signed by a key with a certificate for a different Node ID than the security source? Base BPSec doesn’t really deal with identity logic.
  - A BIB with an “x5t” reference can include the signing certificate (chain). Should a BCB with an “x5t” recipient also include the recipient certificate itself? This is comparable to S/MIME logic.