BPSeq COSE Contexts

BRIAN SIPOS
RKF ENGINEERING SOLUTIONS
IETF110
Goals for COSE/BPSec

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 Security Contexts

One context codepoint with result types defined for each BPSec block type:
- COSE Integrity results (MAC and Signature)
- COSE Confidentiality results (AEAD Encrypt)

Security parameters:
- Additional authenticated data (AAD) scope parameter identical to BPSec Default Security Contexts.
- Public keys in parameters to de-duplicate data (e.g., when signing multiple blocks).
- 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).
Proposed COSE Profile

Required algorithms for AES-GCM-256, AES key-wrap, and HMAC-SHA2-256.

Recommended algorithms for EC and RSA signing and key-wrap/key-generation.

- Additional public key material can be included as security parameters, applying to all results in the block.

<table>
<thead>
<tr>
<th>BPSeq Block</th>
<th>COSE Layer</th>
<th>Name</th>
<th>Code</th>
<th>Implementation Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Integrity</td>
<td>HMAC 256/256</td>
<td>5</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>Integrity</td>
<td>EdDSA</td>
<td>-8</td>
<td>Recommended</td>
</tr>
<tr>
<td></td>
<td>Integrity</td>
<td>PS256</td>
<td>-37</td>
<td>Recommended</td>
</tr>
<tr>
<td></td>
<td>Confidentiality</td>
<td>A256GCM</td>
<td>3</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>Integrity or Confidentiality</td>
<td>A256KW</td>
<td>-5</td>
<td>Required</td>
</tr>
<tr>
<td></td>
<td>Integrity or Confidentiality</td>
<td>ECDH-ES + A256KW</td>
<td>-31</td>
<td>Recommended</td>
</tr>
<tr>
<td></td>
<td>Integrity or Confidentiality</td>
<td>RSAES-OAEP w/ SHA-256</td>
<td>-41</td>
<td>Recommended</td>
</tr>
</tbody>
</table>

Table 4: Interoperability Algorithms
Desired WG Direction

Adoption as WG Draft?

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.
- Validation of a Node ID within a PKIX certificate are already defined in TCPCLv4.

Some secondary questions remain:
- 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?
- Should a mode of operation be to include return-path encryption certificate (as S/MIME does)?
- Etc.