Security Efforts and Extension Block Processing

Angela Hennessy
Laboratory for Telecommunications Sciences (LTS)
Joint Work With:

- Amy Alford
- Kelley Burgin
- Cherita Corbett
Bundle Security Protocol (BSP)
RFC 6257

• Four types of security blocks:
  Bundle Authentication Block (BAB)
  Payload Integrity Block (PIB)
  Payload Confidentiality Block (PCB)
  Extension Security Block (ESB)

• Mandatory Ciphersuites:
  BAB-HMAC
  PIB-RSA-SHA256
  PCB-RSA-AES128-PAYLOAD-PIB-PCB
  ESB-RSA-AES128-EXT
Implementation of PIB, PCB & ESB

- Uses the OpenSSL crypto library
- Mandatory ciphersuites use the Cryptographic Message Syntax (CMS), defined in RFC 5652
- Requires OpenSSL version 1.0.0 or later
Elliptic Curve Ciphersuites

• Internet-Draft uses standard algorithms:
  
  Digital Signatures: ECDSA
  Key Agreement: ECDH
  Encryption: AES

• Two choices for parameters:
  
  NIST P-256 (secp256r1)
  NIST P-384 (secp384r1)
Extension Integrity Block (EIB)

• There is no method in BSP to digitally sign an extension block
• May want to prevent tampering with information in extension blocks
• Same algorithm as PIB
Extension Block Processing in DTN2
Extension Block Processing in DTN2

API/Storage

API_BP::prepare()

API_BP::generate()

API_BP::finalize()

Convergence Layer

consume()

validate()

recv_blocks

prepare()

generate()

finalize()

xmit_blocks
Extension Block Processing in DTN2

- Current work-around: change the API bundle_recv_flag from SRC_API to SRC_PEER
- Use API blocks to change configuration of the daemon
- Allows a user to define per-bundle security policies
Extension Block Processing in DTN2
Proposed Changes

• Remove the api_blocks list
• Add blocks to the recv_blocks list, regardless of where the blocks came from
• Move processing of blocks from consume() to validate()
• Simplify existing block processors
  – Age Extension Block
  – Coding Router
Questions?
Security Block Structure

- Four types of security blocks are defined
- Each block may have several ciphersuites
- Associated to each block is a **Security-Source** and **Security-Destination**

<table>
<thead>
<tr>
<th>type</th>
<th>flags</th>
<th>EID ref list</th>
</tr>
</thead>
<tbody>
<tr>
<td>block data len</td>
<td>ciphersuite ID</td>
<td></td>
</tr>
<tr>
<td>ciphersuite flags</td>
<td>correlator</td>
<td></td>
</tr>
<tr>
<td>params len</td>
<td>security params data</td>
<td></td>
</tr>
<tr>
<td>result len</td>
<td>security result data</td>
<td></td>
</tr>
</tbody>
</table>
Bundle Authentication Block (BAB)

Hop-by-hop Authentication

- Authenticatedates the bundle along one hop of the communications path
- Covers the entire bundle
- Uses a symmetric key-based algorithm
- Each node shares a secret key with each of its neighbors
Payload Integrity Block (PIB)

End-to-end Authentication

- Authenticates the bundle along the entire communications path
- Source computes an RSA signature with the CMS SignedData content type
- Intermediate nodes can verify the signature
Payload Confidentiality Block (PCB)

End-to-end Encryption

- Encrypts the payload data along the entire communications path
- AES in Galois/Counter Mode for content encryption
- RSA encryption of the AES key with the CMS EnvelopedData content type
Extension Security Block (ESB)

End-to-end Encryption

- Encrypts metadata or extension blocks along the entire communications path
- AES in Galois/Counter Mode for content encryption
- RSA encryption of the AES key with the CMS EnvelopedData content type
Example: PCB & CMS

<table>
<thead>
<tr>
<th>type</th>
<th>flags</th>
<th>EID ref list</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>block data len</th>
<th>ciphersuite ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ciphersuite flags</th>
<th>correlator</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>params len</th>
<th>IV</th>
<th>salt</th>
<th>security params data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>key-info</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>result len</th>
<th>ICV</th>
<th>security result data</th>
</tr>
</thead>
</table>

Chain of encrypted keys

Recipient's RSA public key

Recipient's ephemeral key

CMS's ephemeral key

BSP's BEK

CMS's BEK

Encrypted BEK

Version

Recipient ID (issuer and s.no.)

Key Encryption Algorithm

Content type

Content Encryption Alg.

Encrypted Ephemeral key

Encrypted Content Info

Encrypted EnvelopedData
Key Management Issues

• BSP does not cover key management
• Distributing keys is a challenge in DTNs
• Keys could be pre-placed on each node, or swapped between nodes
• Access to revocation checking services cannot be assumed