Extended IKEv2 Payload Format

draft-smyslov-ipsecme-ikev2-extended-pld

Valery Smyslov
svan@elvis.ru

IETF 116
Problems with Existing Format

- Payload Length field occupies 2 bytes, so payload size is limited to 64 Kbytes
  - might not be enough for some PQ algorithms
  - no problem with Message size, which is limited to 4 Gbytes
- Many payloads contain substantial redundancy
  - Payload Length field occupies 2 bytes, while most payloads are shorter
  - most parameters occupy 2 bytes, while less than 256 values are defined
  - zero-filled RESERVED fields
Existing Proposals

• A Larger Internet Key Exchange version 2 (IKEv2) Payload
draft-nir-ipsecme-big-payload

• Beyond 64KB Limit of IKEv2 Payloads
draft-tjhai-ikev2-beyond-64k-limit

• Compact Format of IKEv2 Payloads (expired)
draft-smyslov-ipsecme-ikev2-compact (expired)
Extended Payload Format Overview

• Three formats for new Generic Payload Header
  – for small payloads (up to 64 bytes)
  – for medium size payloads (up to 8 Kbytes)
  – for large payloads (up to 512 Mbytes)

• **No RESERVED fields**

• Revise some existing payloads headers to reduce their size
  – remove unnecessary fields

• Special Format for some payloads (SA, some status notifies)
Extended Generic Payload Header Format

1. Small payloads (2 bytes, 6 bits for Payload Length)

<table>
<thead>
<tr>
<th>Next Payload</th>
<th>C</th>
<th>0</th>
<th>Payload Length</th>
</tr>
</thead>
</table>

2. Medium size payloads (3 bytes, 13 bits for Payload Length)

<table>
<thead>
<tr>
<th>Next Payload</th>
<th>C</th>
<th>1</th>
<th>0</th>
<th>Payload Length</th>
</tr>
</thead>
</table>

3. Large payloads (5 bytes, 29 bits for Payload Length)

<table>
<thead>
<tr>
<th>Next Payload</th>
<th>C</th>
<th>1</th>
<th>1</th>
<th>Payload Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload Length (cont)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Revise some Payload Headers

The following payload headers are revised:

• Key Exchange Payload
  – no RESERVER field (2 bytes)

• Identification, Authentication, Configuration Payloads
  – no RESERVER field (3 bytes)

• Traffic Selector Payload
  – no RESERVER field (3 bytes)
  – no Number of TSs field (1 byte)

• Traffic Selector
  – no Selector Length field (2 bytes)
Special Format for some Payloads

Special format for:

• **SA Payload**
  – SA Payload grows quickly as more and more new transforms are defined and offered by initiators

• **Notify Payload with some Status Type Notification containing no data**
  – Exchange of such payloads is a common way to negotiate support for various protocol extensions, so initial IKEv2 messages grow up as more and more extensions are defined

Both payloads contain a lot of redundancy and can be effectively compacted.
SA Payload

• No RESERVED fields
• No generic header in Proposal substructure
• Encode Transform substructure as variable-length structures
Transform Encoding

1-byte: for Encryption, Key Exchange, PRF, ESN
Transform Types for limited number of Transform IDs

1-byte: for some future Transform Types (e.g. for G-IKEv2) and limited number of Transform IDs

2-bytes: for Additional Key Exchange Transform Types and for other Transform Types with Transform IDs that don’t fit into 1-byte encoding

3-bytes: for Transform IDs that don’t fit into 1-byte and 2-bytes encodings

5-512 bytes: for remaining Transform IDs or in case there are Attributes (other than Key Length)
# Transform Encoding Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Format</th>
<th>Length</th>
<th>Transform Types</th>
<th>Transform IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short</td>
<td>0tttvvvv</td>
<td>1</td>
<td>13-20</td>
<td>0-15</td>
</tr>
<tr>
<td>Encryption (128)</td>
<td>100vvvvv</td>
<td>1</td>
<td>1</td>
<td>11-42</td>
</tr>
<tr>
<td>Encryption (256)</td>
<td>101vvvvv</td>
<td>1</td>
<td>1</td>
<td>11-42</td>
</tr>
<tr>
<td>KE</td>
<td>110vvvvv</td>
<td>1</td>
<td>4</td>
<td>0, 14-44</td>
</tr>
<tr>
<td>PRF</td>
<td>1110vvvvv</td>
<td>1</td>
<td>2</td>
<td>2-15</td>
</tr>
<tr>
<td>ESN</td>
<td>111110vv</td>
<td>1</td>
<td>5</td>
<td>0-3</td>
</tr>
<tr>
<td>Long 1</td>
<td>11110ttt</td>
<td>2</td>
<td>1-31</td>
<td>0-63</td>
</tr>
<tr>
<td></td>
<td>ttvvvvvv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long 2</td>
<td>1111110t</td>
<td>3</td>
<td>1-31</td>
<td>0-4095</td>
</tr>
<tr>
<td></td>
<td>ttttvvvv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>vvvvvvvv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full</td>
<td>1111111t</td>
<td>5 up to 512 (in case of attributes)</td>
<td>any</td>
<td>any</td>
</tr>
</tbody>
</table>
Example of Compact SA Payload

SA Payload with one Proposal and three Transforms:

- **ENCR_AES_CCM_16** (256 bits key)
- **PRF_HMAC_SHA2_256**
- 4096-bit MODP Group

Original: **40 bytes**

Compact: **9 bytes**
Notify Payload

Outline: encode notification in one octet (limited to first 256 status notifications) and omit all other fields from Notify Payload

Example: Notify Payload with IKEV2_FRAGMENTATION_SUPPORTED notification.
Negotiation

If new format is used from the very beginning then the following options exist

- **New status notify** `EXTENDED_PAYLOAD_FORMAT`
  - extended format cannot be used in `IKE_SA_INIT`
  - suitable if only large payloads are needed

- **New initial exchange** `X_IKE_SA_INIT`
  - functionally equivalent to `IKE_SA_INIT`, but may contain payloads in extended (compact) format
  - old responders would return `INVALID_SYNTAX` notify
Transport Issues

Transport issues for transferring large payloads (> 64 Kbytes) are out of scope. Possible solutions:

• IKE over TCP combined with IKE fragmentation (to solve limitation on 64 Kbytes on a single IKE message over TCP)

• Mixed Mode (defined in draft-tjhai-ikev2-beyond-64k-limit: IKE over TCP + IKE fragmentation combined with plain ESP or ESP over UDP) can be used to avoid ESP performance degradation when used with TCP encapsulation
Discussion

• Get rid of SPI Size in Proposal substructure, Delete and Notify payloads (can be deducted from Protocol ID)?

• Get rid of Proposal Num in Proposal substructure?

• New Payload types or reuse existing types?
  – 9+ new payload formats – too many?

• Certificates consume a lot of space, can be compressed (out of scope)
  – RFC 8879 is an example of certificate compression
Thank you!

• Comments?
• Questions?
• Any interest in this work?