Diet-ESP: A Flexible and Wide Range Security protocol

draft-mglt-6lo-diet-esp-requirements draft-mglt-6lo-diet-esp

D. Migault, T. Guggemos, S. Raza, C. Bormann

19/07/2016- IETF96- Berlin

Motivations

The current de-facto IoT security protocol is DTLS1.2 (DICE profile).

- Reasonable choice for:
  - Web based IoT applications,
  - End-to-end security
- but in our view DTLS1.2/DICE does not address all IoT segments
- There is a need for ESP in IoT
- Diet-ESP is the ESP implementation for IoT

ESP packet Description

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ----
|               Security Parameters Index (SPI)                 | ^Int.
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Cov-
|                      Sequence Number                          | |ered
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | ----
|                    Payload Data* (variable)                   | |   ^
~                                                               ~ |   |
|                                                               | |Conf.
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Cov-
|               |     Padding (0-255 bytes)                     | |ered*
+-+-+-+-+-+-+-+-+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |   |
|                               |  Pad Length   | Next Header   | v   v
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ------
|         Integrity Check Value-ICV   (variable)              |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Diet-ESP

Diet-ESP aims at compressing:

- ESP Header
- ESP Trailer

Diet-ESP is not something new:

- ROHC/6LowPAN provides ways to compress the ESP Header
- ROHCoverIPsec provides ways to compress the clear text data

Diet-ESP

What is new with Diet-ESP is that:

- It enables to compress Padding, Pad length and Next Header fields
- Compression occurs before the encryption (similarly to ROHCoverIPsec)
- Compression occurs after the ESP fields are added (unlike ROHCoverIPsec)
  - There is no decompression of these fields. (one way compression)
- It takes advantage of IKEv2 to agree on the compression rules
  - Results in a light-compression framework

Diet-ESP/ROHC

Currently Diet-ESP is based on ROHC:

- ROHC compressor/decompressor
  - With initialized states
  - Without synchronization, initialization exchanges
- ROHC profiles

Open Discussion are:

- Does Diet-ESP defines ROHC compressor or a specific compressor ?
  - Other alternatives exists like SCHC
  - Our profiles will be called rules
- Can we consider the profiles as an extension of ROHCoverIPsec ?
  - This would make possible ROHC/Diet-ESP and Diet-ESP

Diet-ESP / ESP

Diet-ESP is based on ESP:

- Diet-ESP is based on standard ESP
- Diet-ESP is able to send ESP packet
  - Enable natural fall back to uncompressed ESP
  - Preserve ESP interoperability
- Diet-ESP DOES NOT modify cryptographic parameters, algorithms
  - Crypto is left untouched

Diet-ESP / AES-CCM

How Diet-ESP works with AES-CCM

```
- 1:   Decompress ESP header.
- 2:   Generate Diet-ESP ICV and check ICV send in the packet.
- 3:   Check anti-replay
- 4:   Remove compressed header.
- 5:   Decrypt the Diet-ESP payload.
[...]
```

Questions

- Does ESP requires 32/64 bit alignment ?
- How Padding is generated ?
- Does an AES-CCM ESP packet has Padding ?
- How the ICV is built ?

Bit Alignment

Does ESP requires 32/64 bit alignment ?

- RFC4303 section 2.4

```
o Padding also may be required, irrespective of encryption
  algorithm requirements, to ensure that the resulting
  ciphertext terminates on a 4-byte boundary.  Specifically,
  the Pad Length and Next Header fields must be right aligned
  within a 4-byte word, as illustrated in the ESP packet
  format figures above, to ensure that the ICV field (if
  present) is aligned on a 4-byte boundary.
```

- draft-mglt-diet-esp-requirements

```
IP extension headers MUST have 32 bit Byte-Alignment in IPv4 (section
3.1 of [RFC0791] - Padding description) and a 64 bit Byte-Alignment
in IPv6 (section 4 of [RFC2460]).  As ESP [RFC4303] is such an
extension header, padding is mandatory to meet the alignment
constraint.
```

Padding Generation

How Padding is generated ?

- RFC4303 section 2.4

```
If Padding bytes are needed but the encryption algorithm does not
specify the padding contents, then the following default processing
MUST be used.  The Padding bytes are initialized with a series of
(unsigned, 1-byte) integer values.  The first padding byte appended
to the plaintext is numbered 1, with subsequent padding bytes making
up a monotonically increasing sequence: 1, 2, 3, ....  When this
padding scheme is employed, the receiver SHOULD inspect the Padding
field.  (This scheme was selected because of its relative simplicity,
ease of implementation in hardware, and because it offers limited
protection against certain forms of "cut and paste" attacks in the
absence of other integrity measures, if the receiver checks the
padding values upon decryption.)
```

Padding in AES-CCM

Does an AES-CCM ESP packet has Padding ?

- RFC4309

```
Padding:
The encrypted payload contains the ciphertext.

AES CCM mode does not require plaintext padding.  However, ESP does
require padding to 32-bit word-align the authentication data.  The
Padding, Pad Length, and Next Header fields MUST be concatenated
with the plaintext before performing encryption, as described in
[ESP].  When padding is required, it MUST be generated and checked
in accordance with the conventions specified in [ESP].
```

Padding in AES-CCM

How ICV is built ?

- RFC4309 section 3.3 Autentication Data

```
AES CCM provides an encrypted ICV.  The ICV provided by CCM is
carried in the Authentication Data fields without further encryption.
Implementations MUST support ICV sizes of 8 octets and 16 octets.
Implementations MAY also support ICV 12 octets.
```

Next

- ESP Payload Compression?

* Implicit IV (presentation of Yoav) * Tunnel Header compression (similar to BEET-MODE/draft-mglt-6lo-diet-esp-payload-compression) * Transport Header Compression

Thank you for your attention