Diet-ESP

draft-mglt-ipsecme-diet-esp-requirements-00.txt,
draft-mglt-ipsecme-diet-esp-01.txt,
draft-mglt-ipsecme-diet-esp-iv-generation-00.txt,
draft-mglt-ipsecme-diet-esp-payload-compression-00.txt

D. Migault, T. Guggemos

24/07/2014- IETF90- Toronto

Table of Content

IPsec Architectures

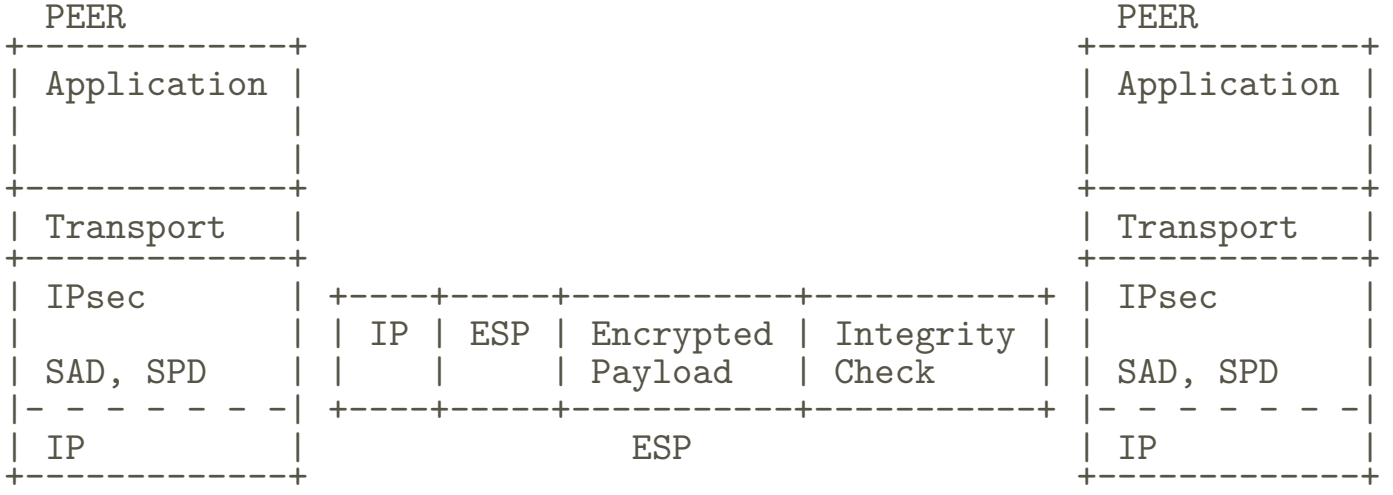
IPsec [RFC4301] described an architecture

- End-to-End Security

- Virtual Private Network Security (VPN)

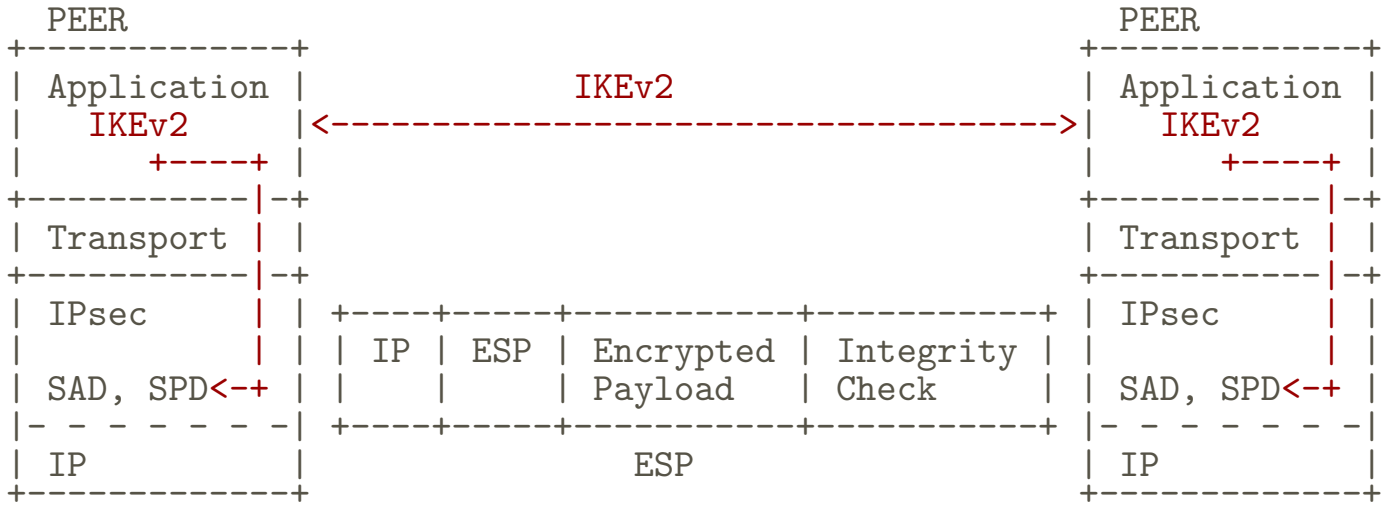
IPsec Protocol Suite

- IPsec/ESP [RFC4303] describes the format of IPsec packets
- AES-CTR [RFC3686], AES-CBC [RFC3602] AES-CCM [RFC4309] describe how to encrypt the data



IPsec Protocol Suite

- IPsec/IKEv2 [RFC5996] describes how peers agree on IPsec/ESP material



Why IPsec for IoT?

IPsec provides the following advantages:

- IPsec security is performed at the IP layer
- IPsec makes security transparent for transport and applications layer
- IPsec has no sessions and is well designed for sleeping nodes
- IPsec secures the whole device (like a firewall)

IPsec Security is usually implemented in the Kernel which makes:

- IPsec very popular for securing infrastructure
- IPsec less popular than TLS/DTLS for securing applications
- Kernel / User land should not be an issue for IoT applications

IPsec is well designed for IoT security so Diet-ESP is designed as IPsec for IoT

Diet-ESP Requirements

Identified requirements to make Diet-ESP successful for IoT are:

- Flexible Compression
- Enable low code complexity
- Easy to use / configure
- Keep the same level of security as Standard IPsec

Full list of requirements are listed in [draft-mglt-ipsecme-diet-esp-requirements]

Diet-ESP Architecture

Diet-ESP design principles:

- Keep Standard ESP as defined in [RFC4303]
- Compress / Decompress the ESP Packet

Diet-ESP Architecture

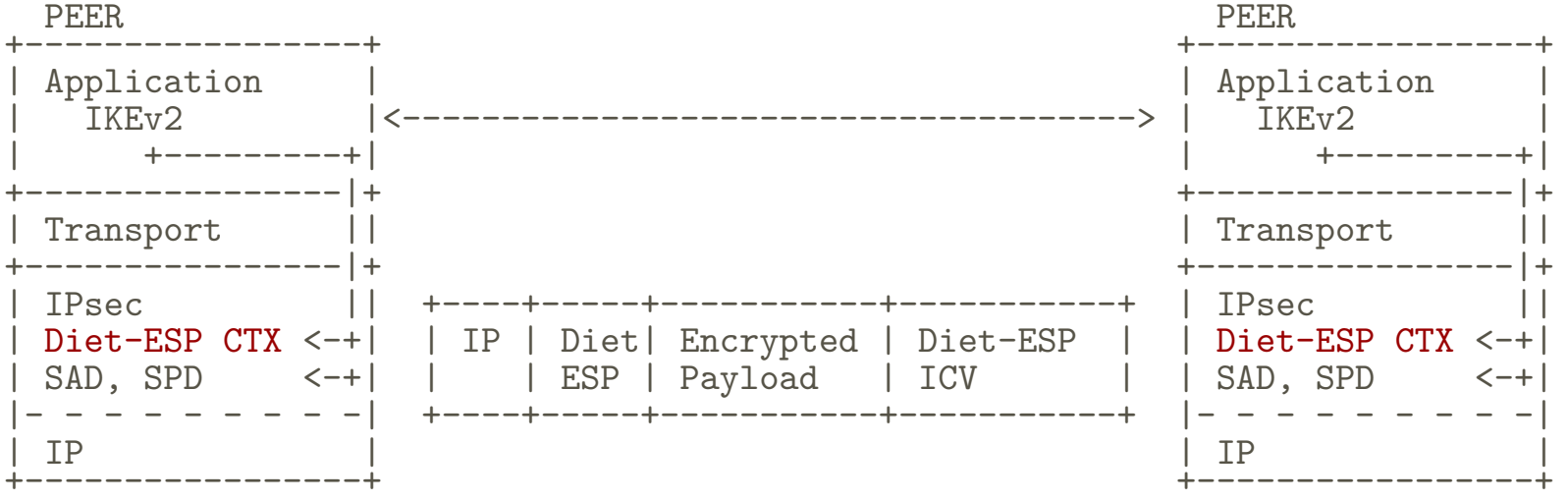
Diet-ESP Compression / Decompression

- Defined by the Diet-ESP Context
- Use IPsec information (Traffic Selectors)
- Determinist, stateless

Diet-ESP Context:

- Small – Diet-ESP Context is stored in two Bytes
- Defines hardware alignement / size ESP fields are compressed
- Security considerations of compression is documented
- Suggested default value to ease application developpers
- Can be negociated with IKEv2

Diet-ESP Architecture

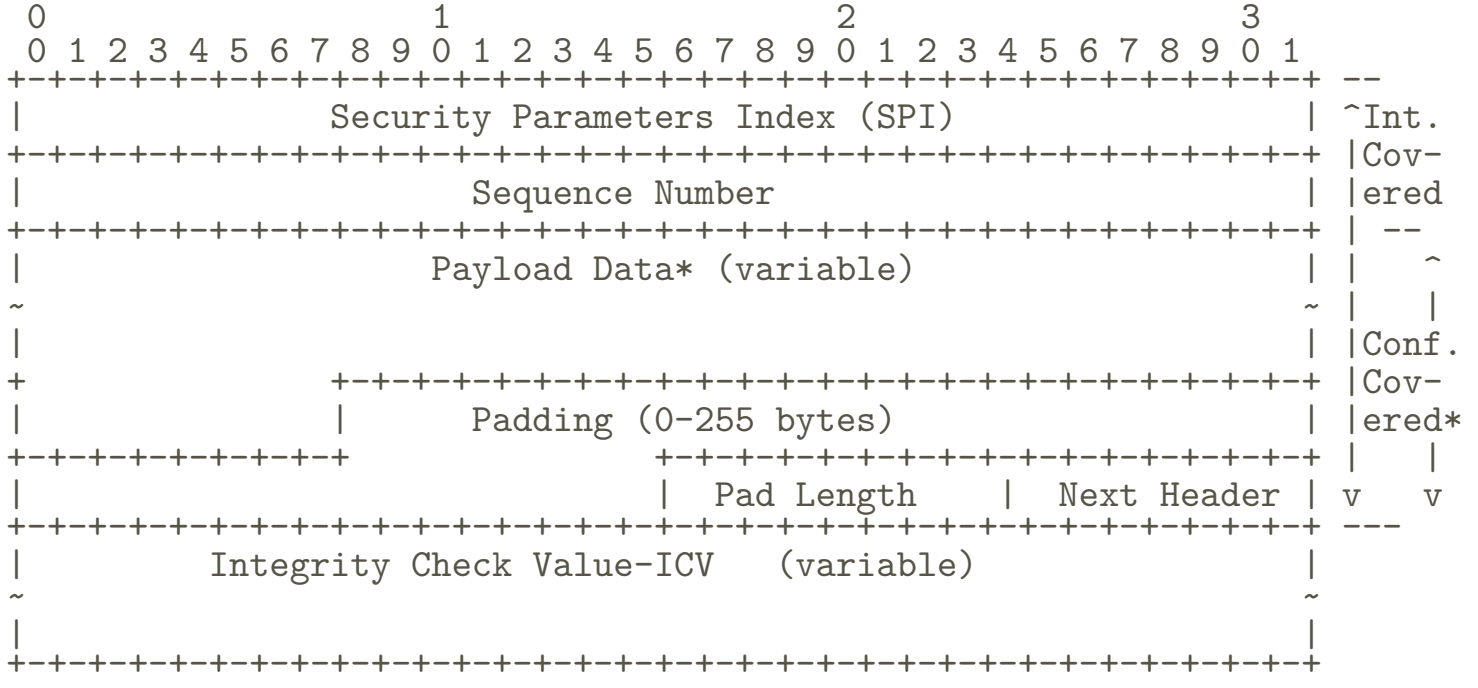


Position toward ROHC/6LowPAN

Position toward ROHC, ROHCoverIPsec, 6LowPAN, 6LowPANoverIPsec:

- Described with ROHC [RFC3095], [RFC5225] and ROHCoverIPsec [RFC5856], [RFC5857], [RFC5858] terminology
- Takes advantage of the negociated IPsec parameters
  - Stateless compression / decompression
  - No in-band information
- Compress EVERY ESP fields
- Designed for constraint devices, not only for bandwidth optimization.
- Mostly compatible with ROHC, 6LowPAN

Compression Performance



Compression Performance

Evaluation of the IP payload for sending 1 Byte of Data:

- An example – see [EX] for more detail
- A temperature sensor sending information every hour / day

Protocol	Total Size (Bytes)	ESP	IV	UDP	ICV
ESP	40 [80]	11	8	8	12
Diet-ESP	5 [5]	0	0	0	4
6LowPAN(overIPsec)	26 [29]	4	8	1	12
ROHC(overIPsec)	24 [0]	3	8	0	12

Development Work

We have two implementations of the previous Diet-ESP version:

- Python
- Contiki

Measurements showed:

- Memory footprint: 578 Bytes
- Simulation with Cooja shows with a $20\,\%$ accurancy that time and energy consumption is "linear" to the number of byte sent

Future Work:

- Update the current version to new version and new compressions
- Build a C implementation for the "Server Side"
- Define IKEv2 extention

Diet-ESP Context

- ALIGN: define the hardware supported alignment
- SPI_SIZE: Number of LSB SPI sent
- SN_SIZE: Number of LSB SN sent
- NH: defines if NH is present or not
- PAD: if the PadLength field is present or not
- Diet-ESP_ICV_SIZE: Number of LSB of the Diet-ESP_ICV
- IV_COMPRESSION: defines if the IV is compressed
- COMPRESS_ESP_PAYLOAD: defines if IPsec TS are used by compressors
- CHECKSUM_LSB: Number of LSB of the transport checksum field

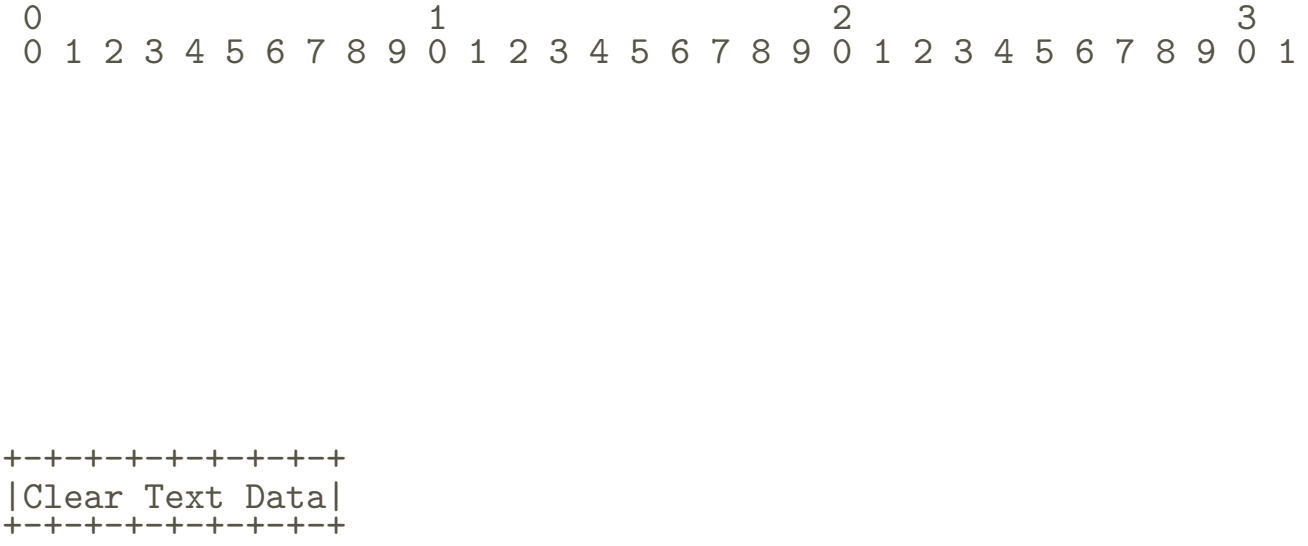
Ex: ESP Packet for 1 Byte Data

Use Case: A sensor sends 1 byte of Temperature data every hour

- Data is sent over UDP Port Source, Port Destination
- IPsec Security Association is negotiated for UDP Port Source, Port Destination
- Encryption is AES-CTR [RFC3686] with an 8 Byte IV
- Integrity Check Value AES-XCBC-MAC [RFC3566] 12 Byte ICV

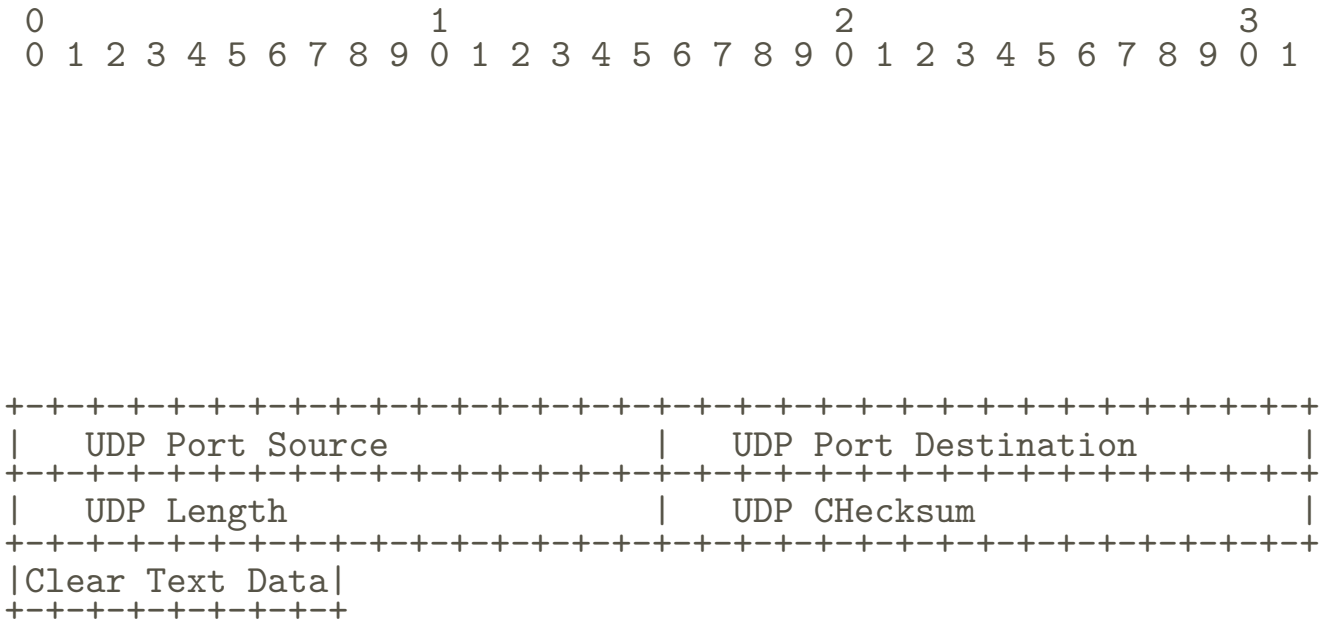
Ex: ESP Packet for 1 Byte Data

1 Byte Data



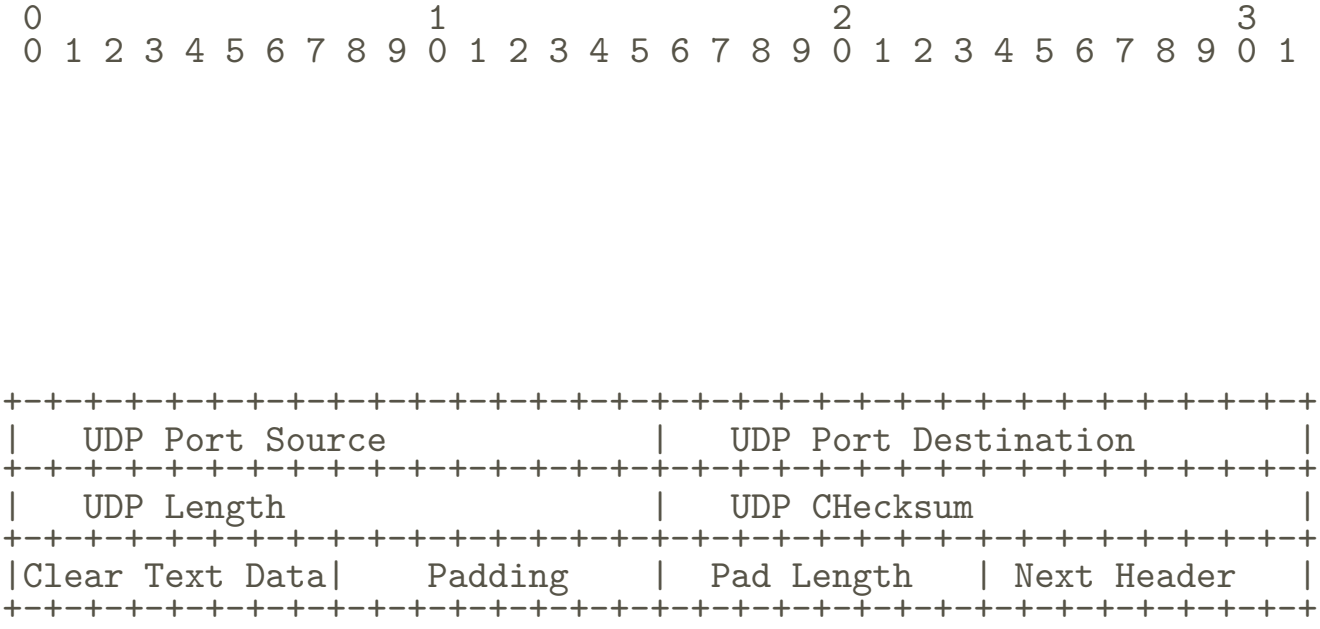
Ex: ESP Packet for 1 Byte Data

UDP Encapsulation



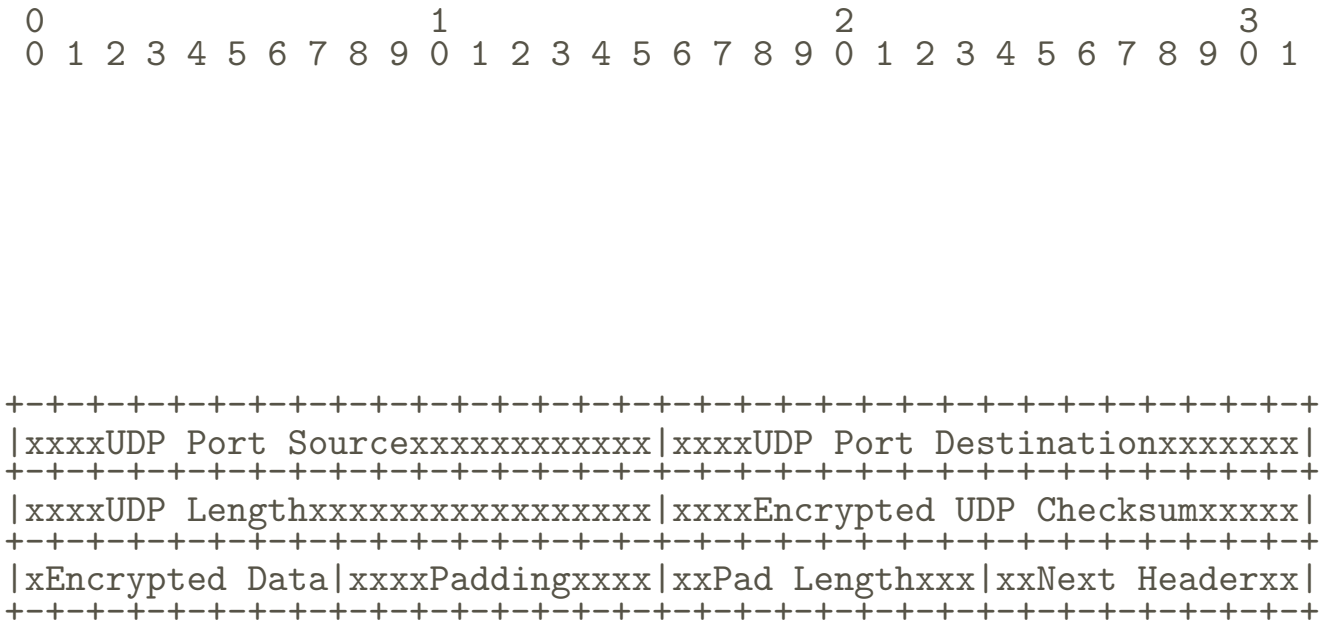
Ex: ESP Packet for 1 Byte Data

Append ESP Trailer



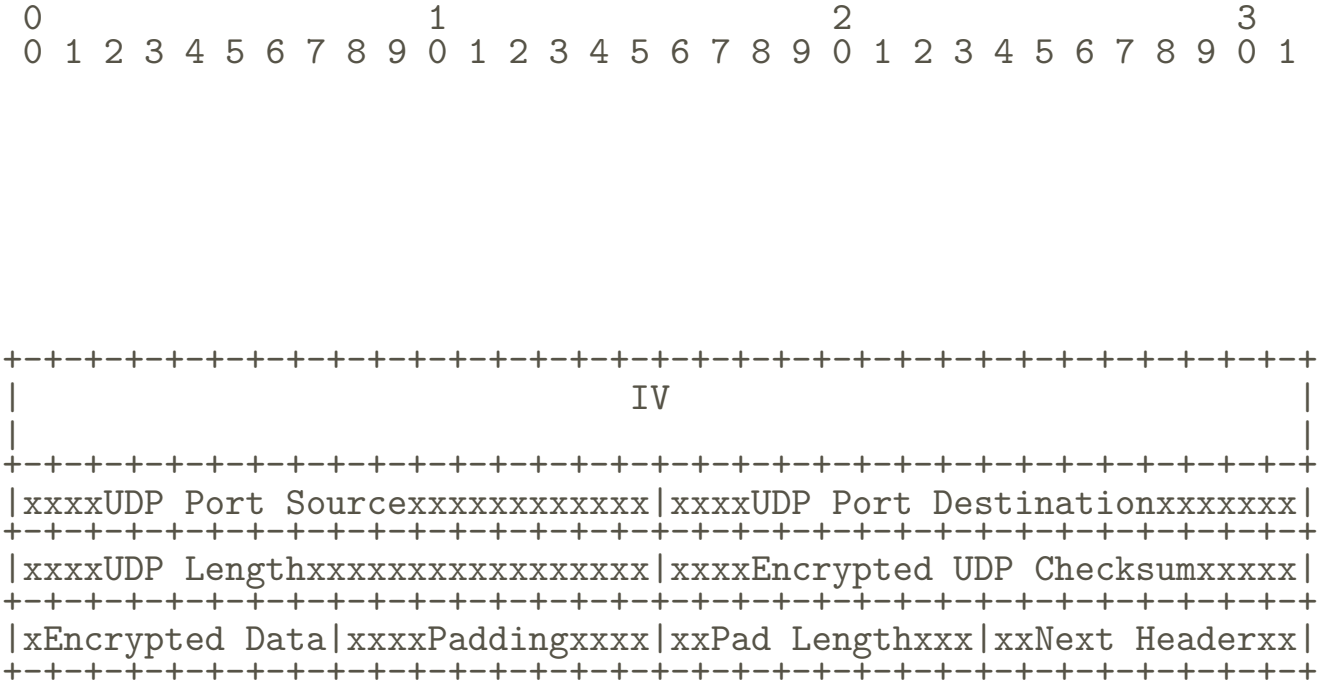
Ex: ESP Packet for 1 Byte Data

AES-CTR Encryption



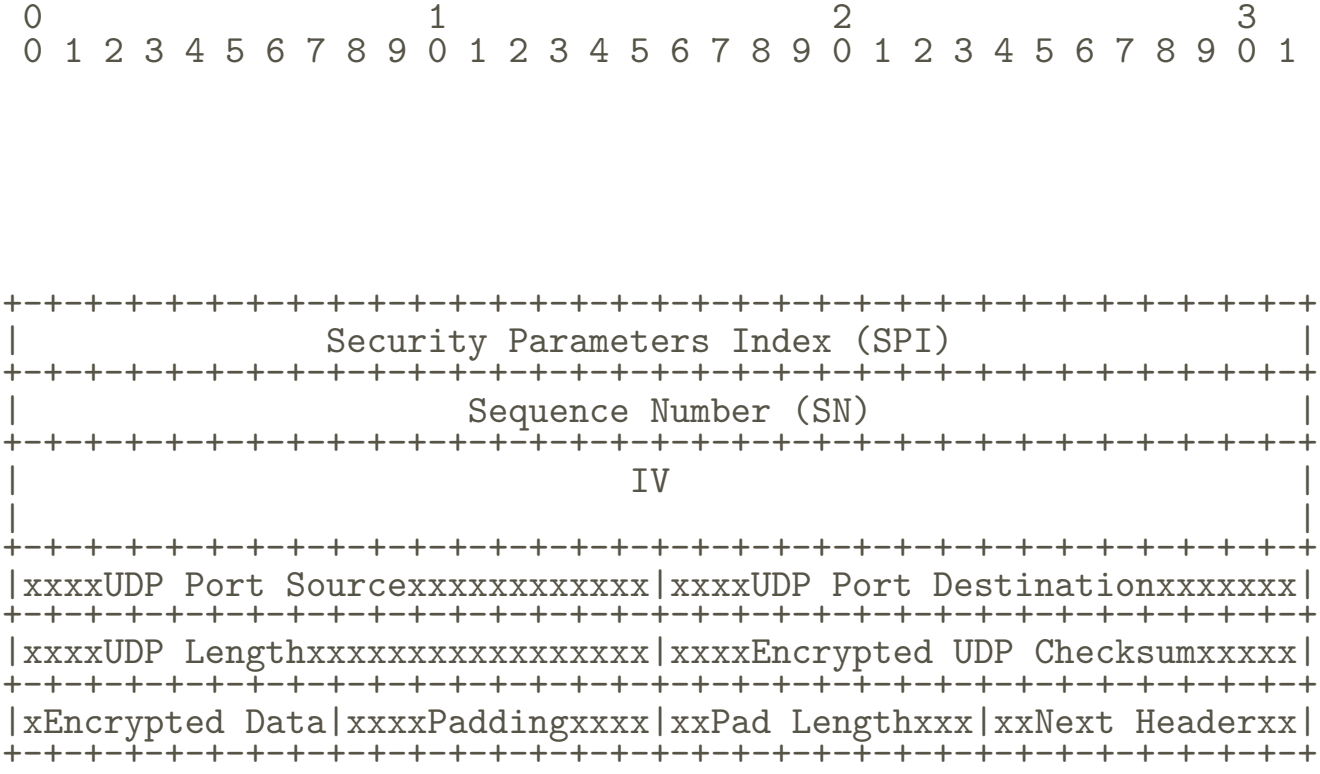
Ex: ESP Packet for 1 Byte Data

Adding IV



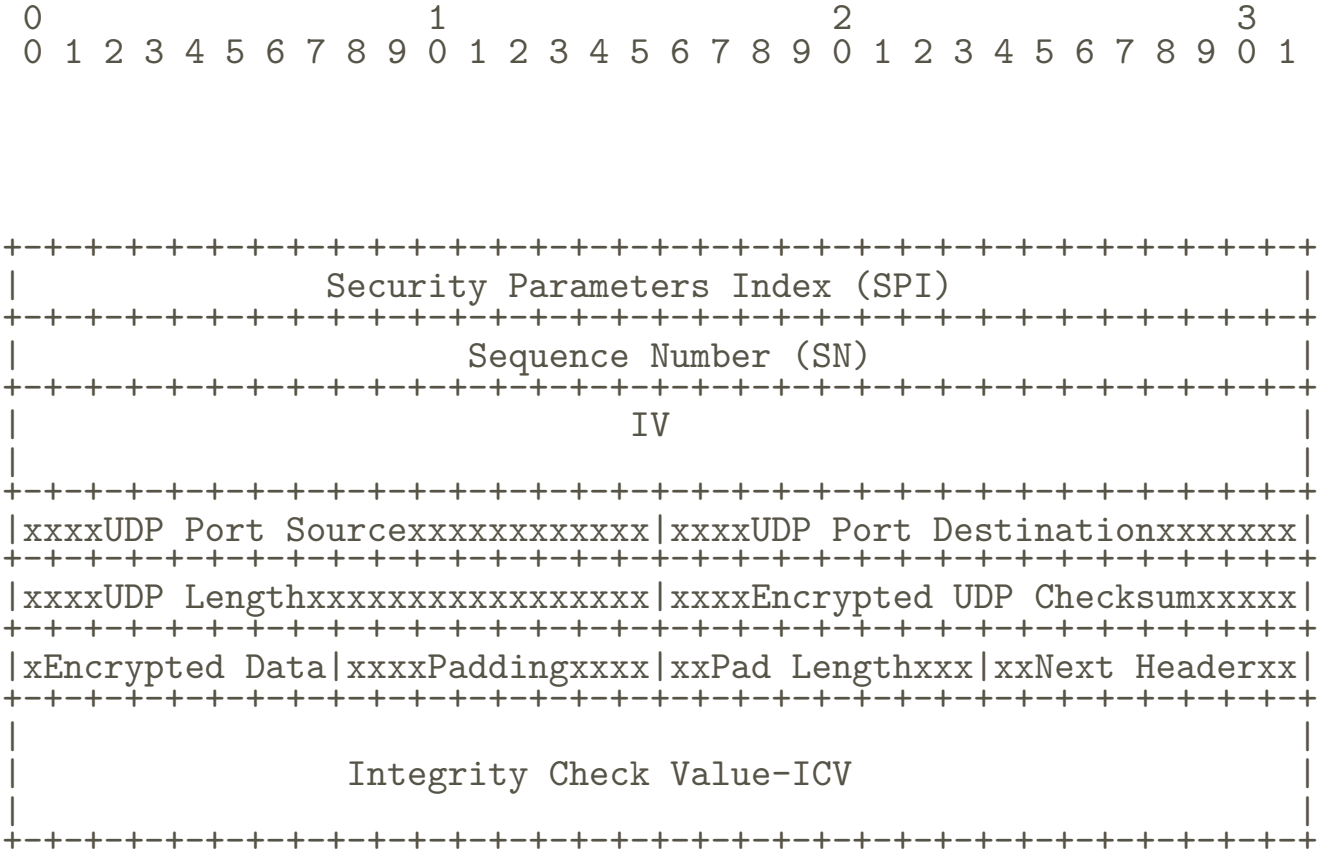
Ex: ESP Packet for 1 Byte Data

Adding ESP Header



Ex: ESP Packet for 1 Byte Data

Appending ICV



Ex: Diet-ESP Packet for 1 Byte Data

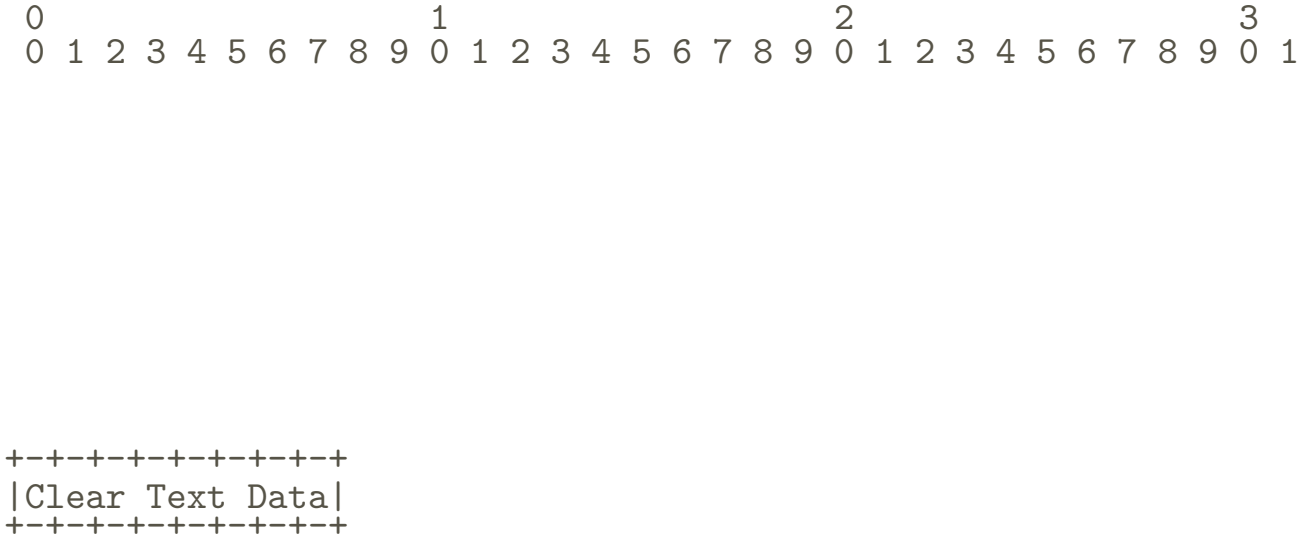
Definition Diet-ESP Context:

- ALIGN: 8 bit (Sensor does not require 32 bit alignment)
- SPI_SIZE: 0 (IP addresses identify the connection)
- SN_SIZE: 0 (derived from time)
- No Next Header field, No Pad Length field
- Diet-ESP_ICV_SIZE: 4 bytes instead of 12.

- IV Compression [draft-mglt-ipsecme-diet-esp-iv-generation]
  - IV_COMPRESSION: Set IV is generated locally on both peers
- ESP Payload Compression [draft-mglt-ipsecme-diet-esp-payload-compression]
  - COMPRESS_ESP_PAYLOAD (UDP, Ports are stored in SAD, SPD)
  - CHECKSUM_LSB: 0 (ICV provides Integrity Check)

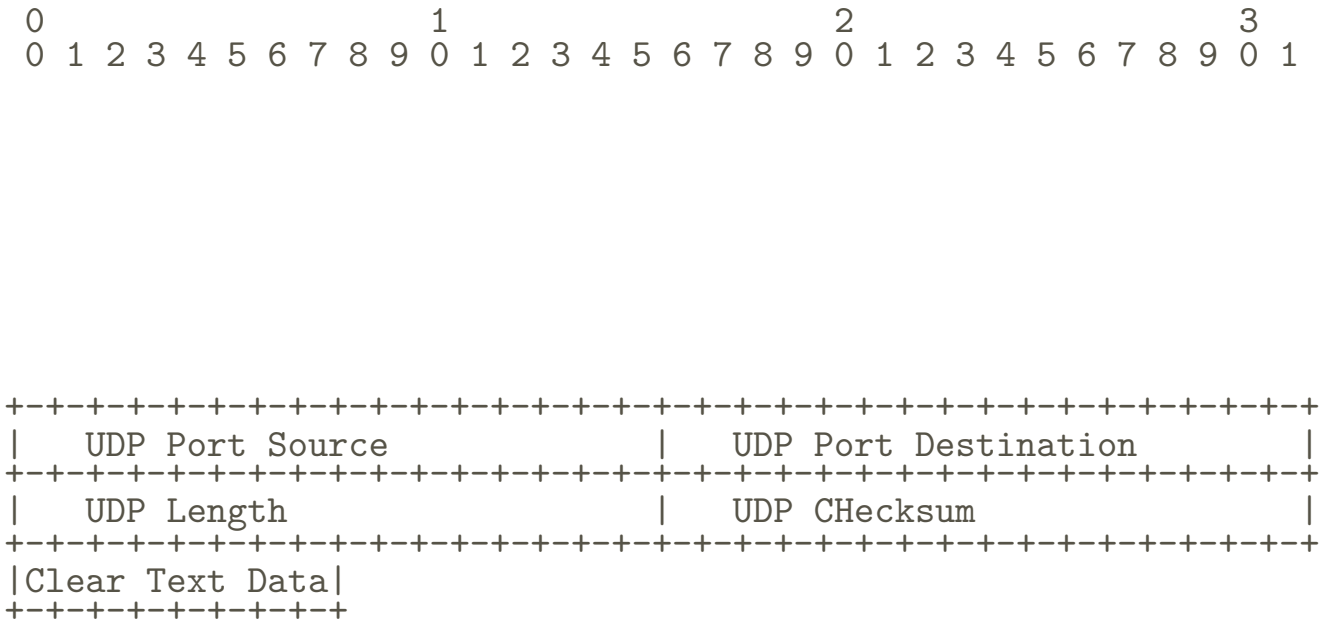
Ex: Diet-ESP Packet for 1 Byte Data

1 Byte Data



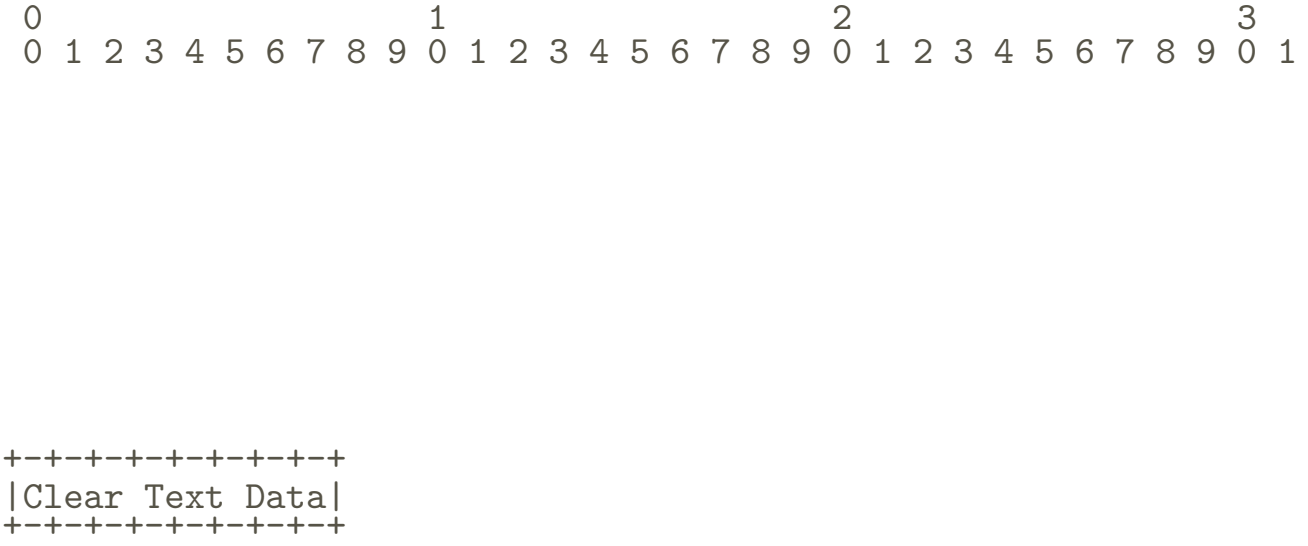
Ex: Diet-ESP Packet for 1 Byte Data

UDP Encapsulation



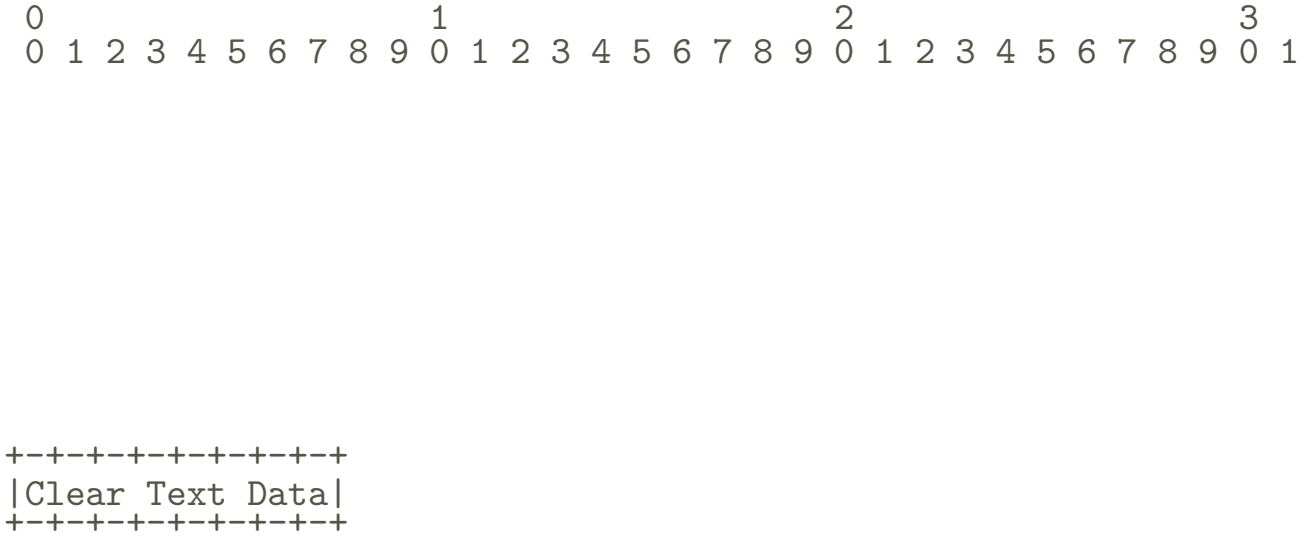
Ex: Diet-ESP Packet for 1 Byte Data

ESP Payload Compression



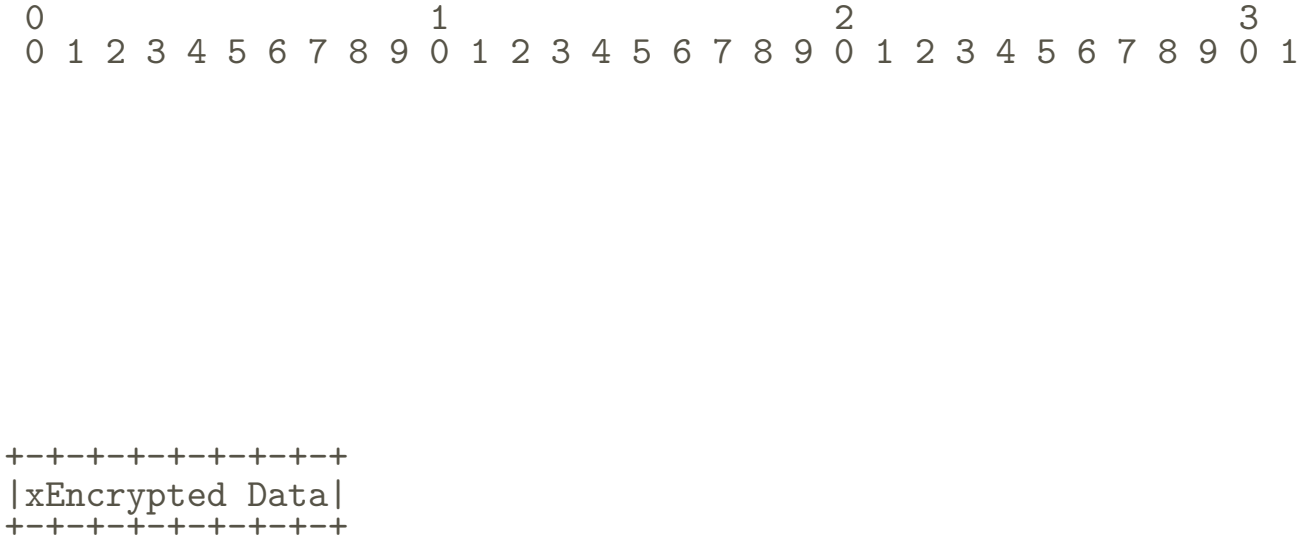
Ex: Diet-ESP Packet for 1 Byte Data

Append Compressed ESP Trailer (ESP Compression Phase 1)



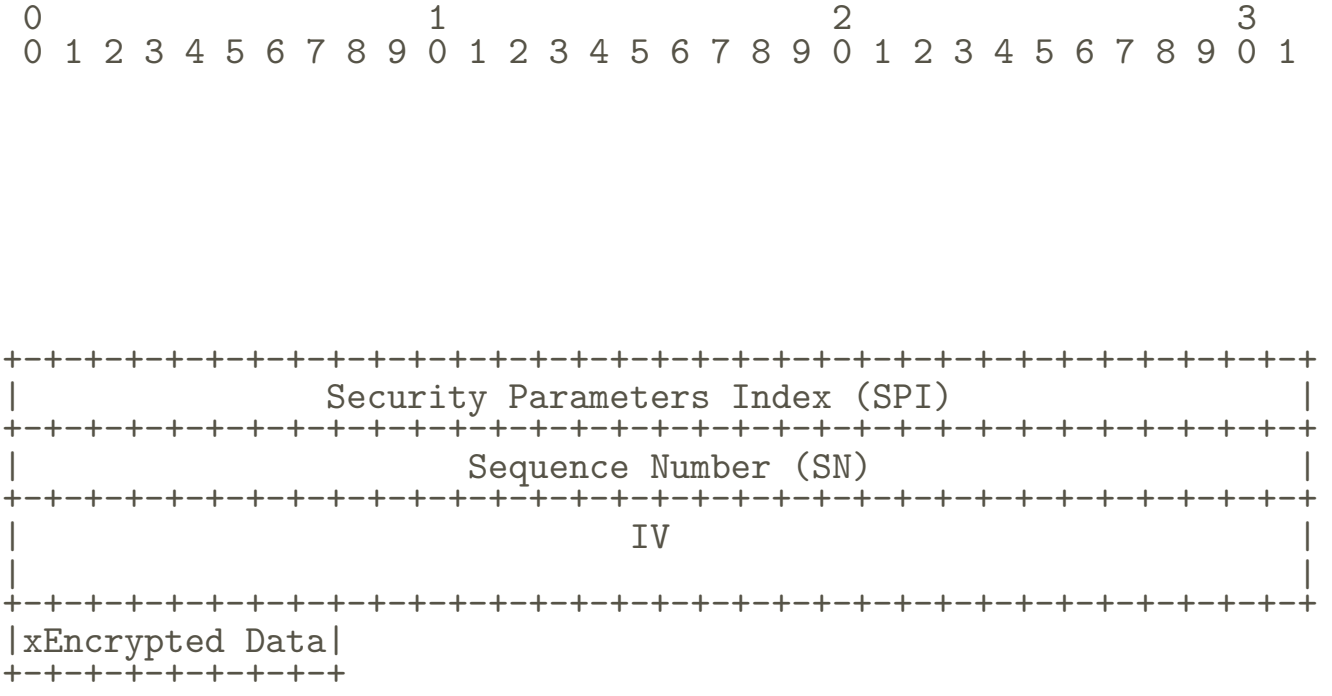
Ex: Diet-ESP Packet for 1 Byte Data

AES-CTR Encryption



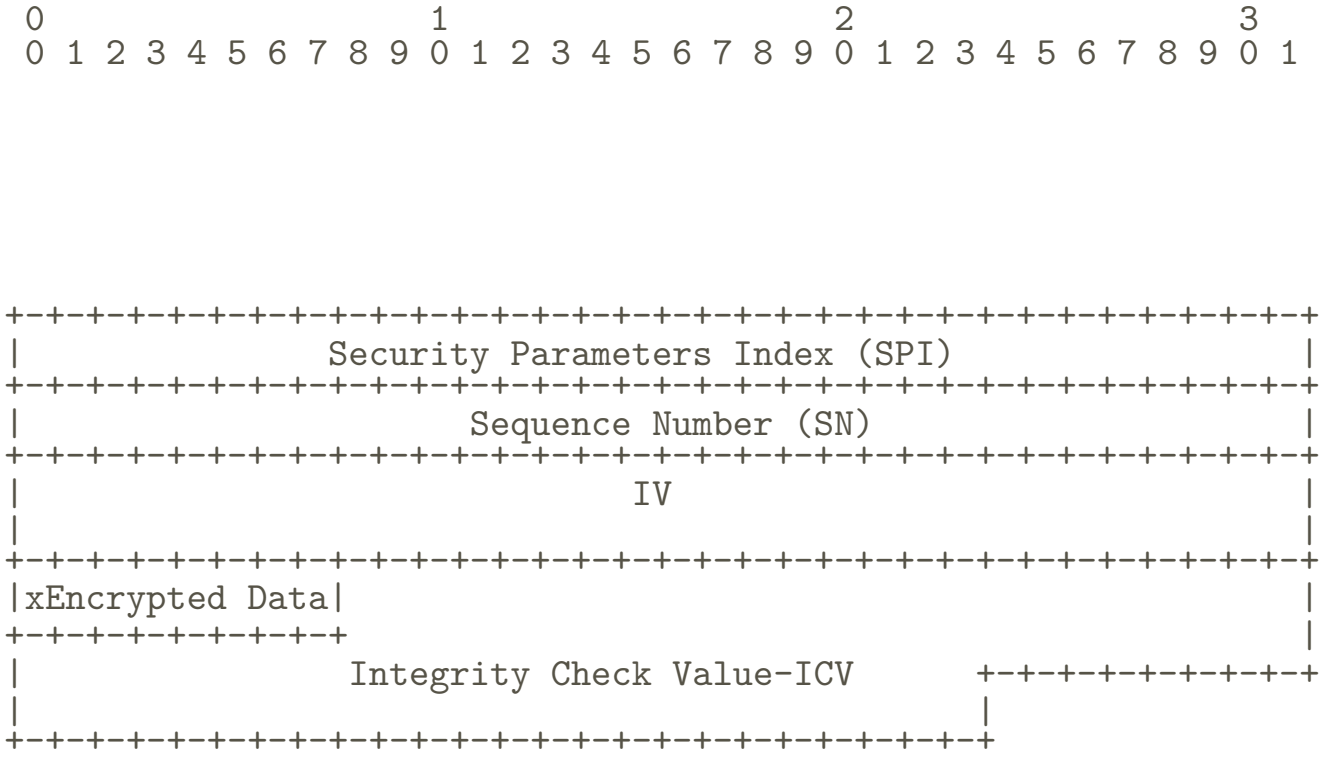
Ex: Diet-ESP Packet for 1 Byte Data

Adding ESP Header



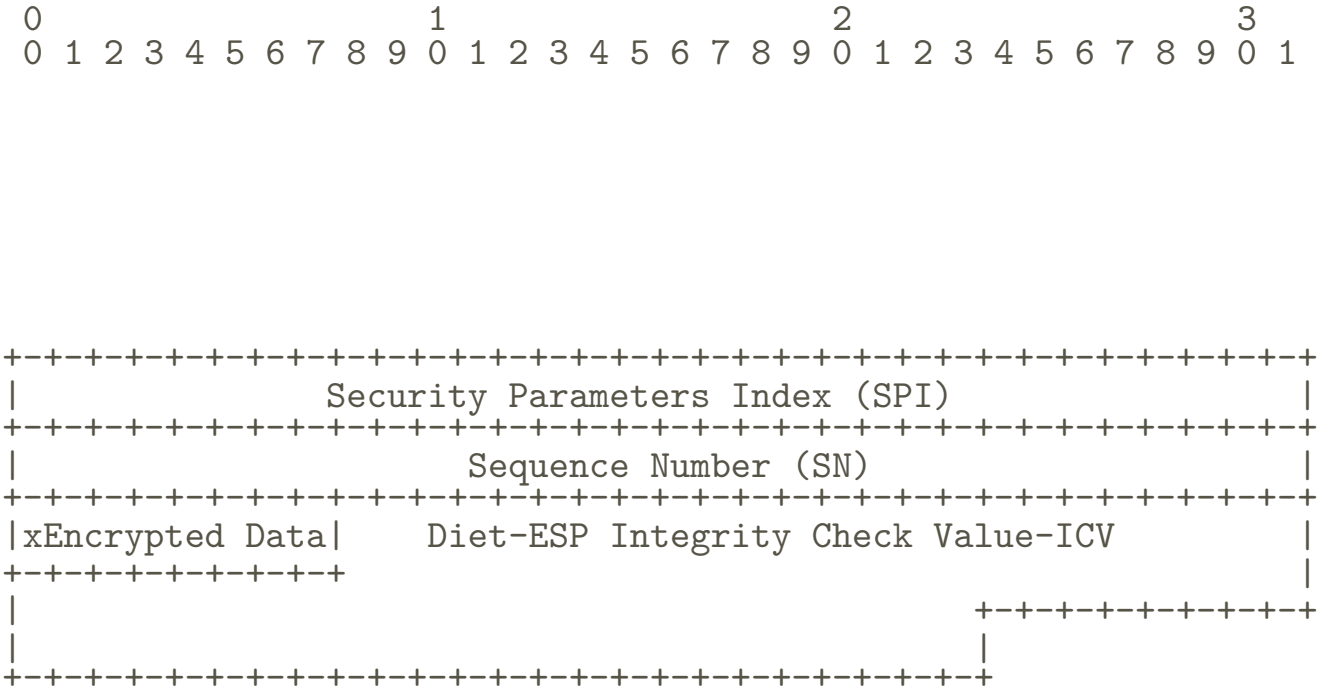
Ex: Diet-ESP Packet for 1 Byte Data

Appending ICV



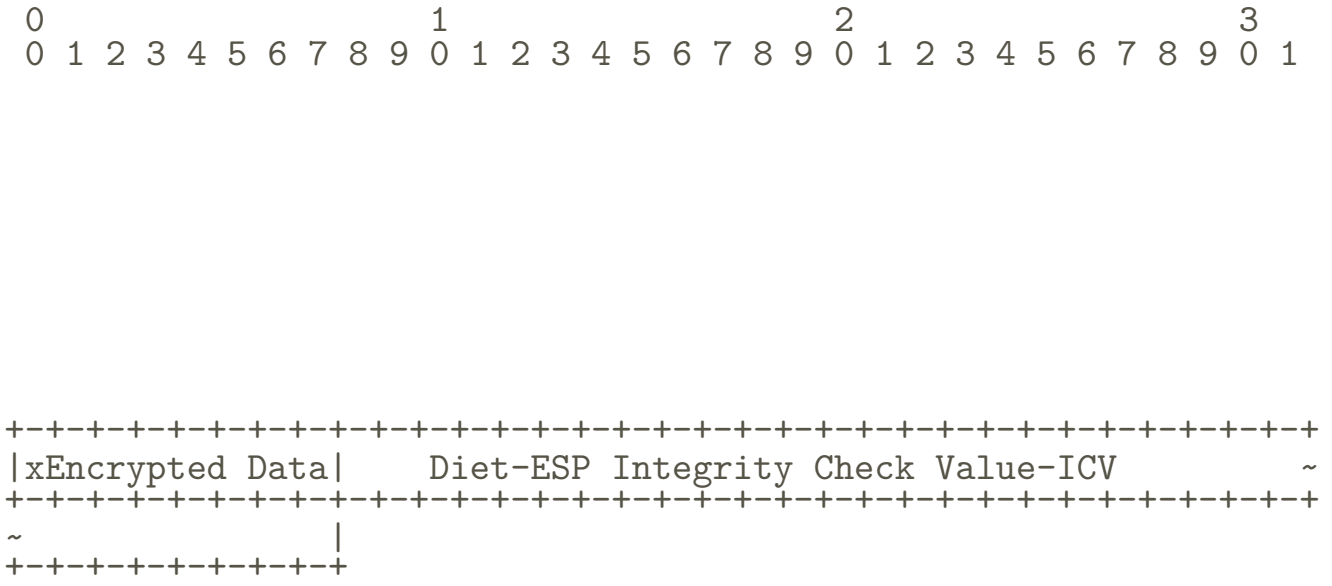
Ex: ESP Packet for 1 Byte Data

IV Compression



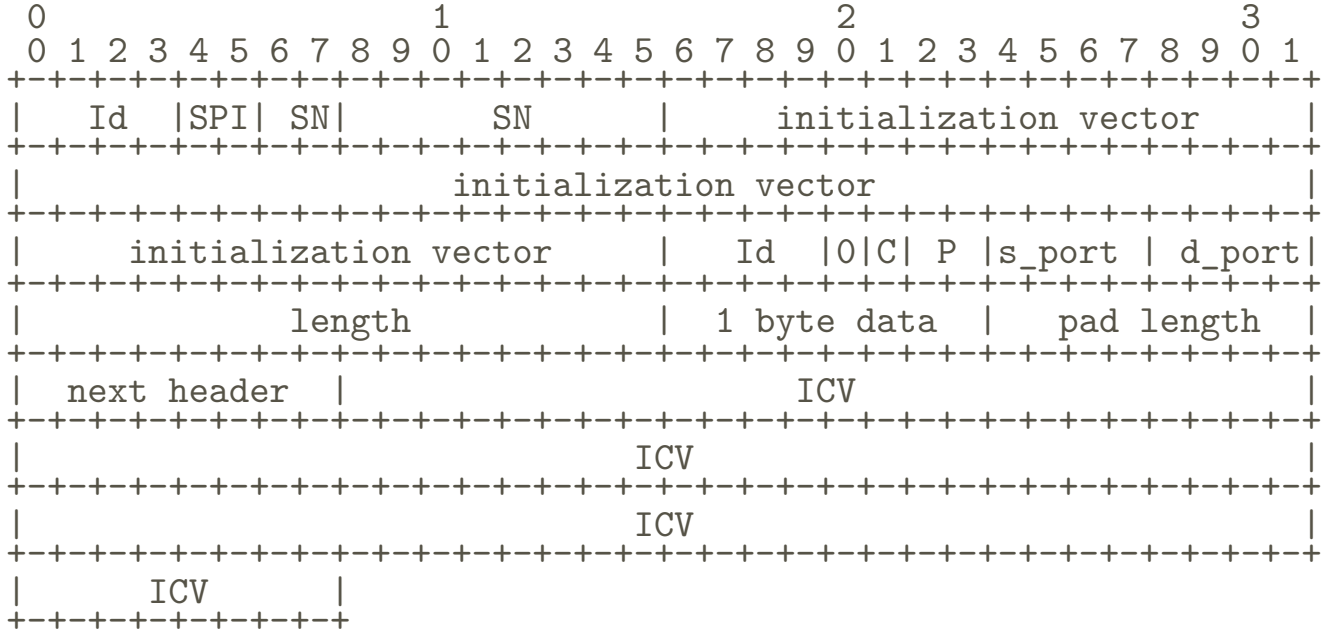
Ex: Diet-ESP Packet for 1 Byte Data

Compressing ESP Header / Diet-ESP ICV (ESP Compression Phase 2)



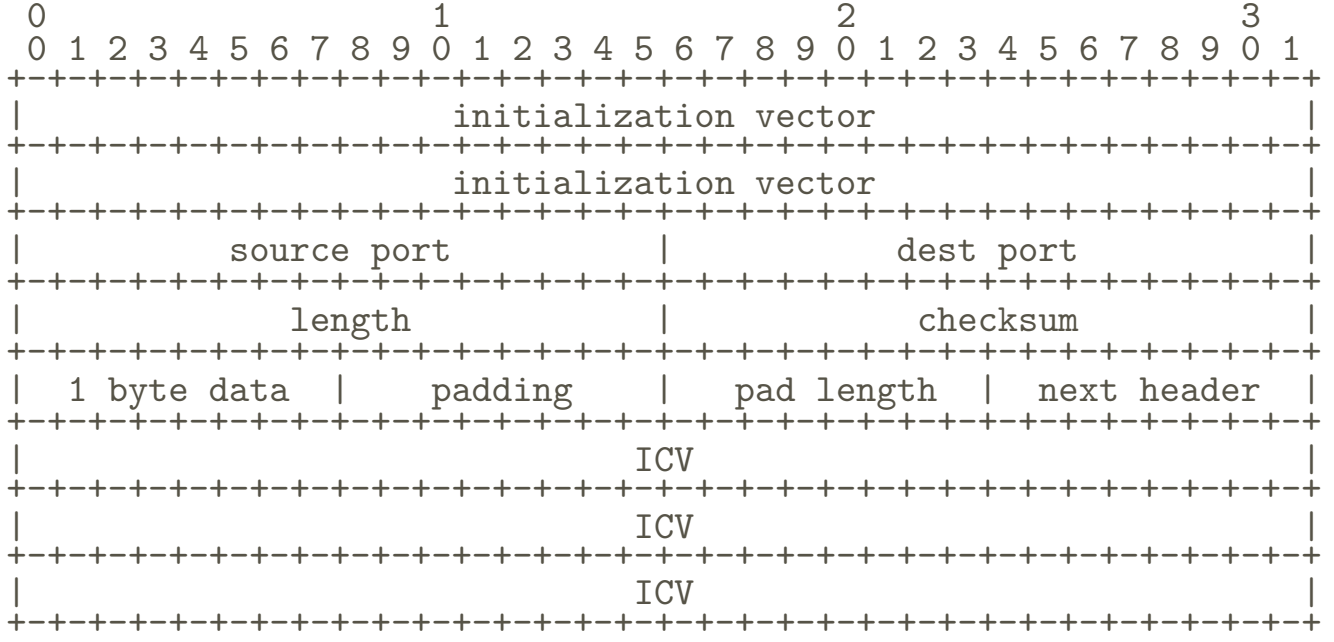
Ex: 6LowPan Packet for 1 Byte Data

(Maximum compression)



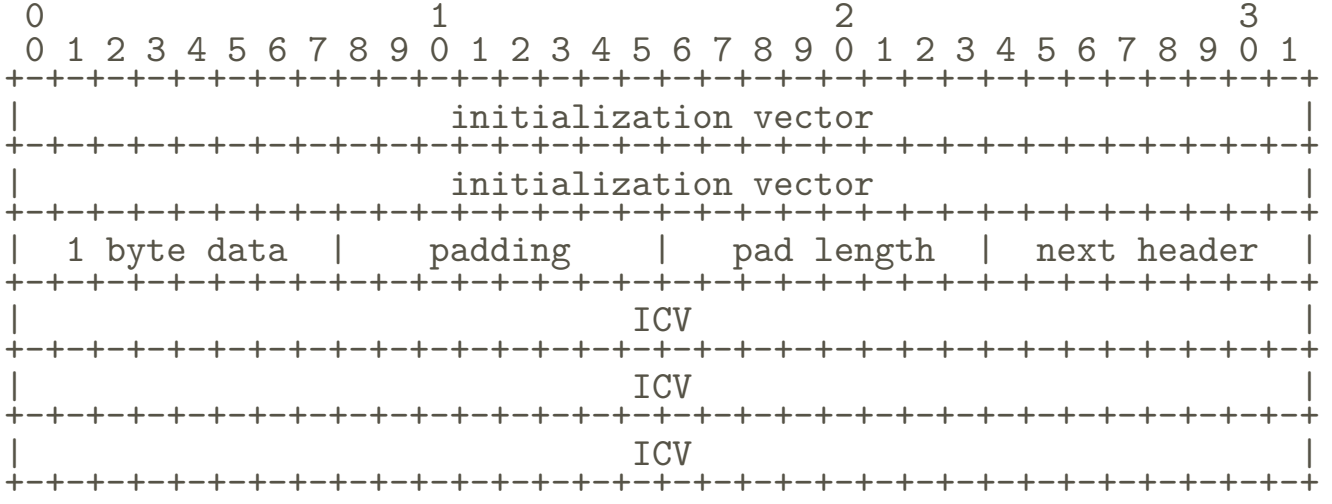
Ex: ROHC Packet for 1 Byte Data

(Maximum compression)



Ex: ROHC + ROHCoverIPsec Packet for 1 Byte Data

(Maximum compression)



Compression Performance

Evaluation of the IP payload for sending 1 Byte of Data:

- An example – see [EX] for more detail
- A temperature sensor sending information every hour / day

Protocol	Total Size (Bytes)	ESP	IV		UDP	ICV
ESP	40 [80]	11	8		8	12
Diet-ESP	5 [5]	0	0		0	4
6LowPAN(overIPsec)	26 [29]	4	8		1	12
ROHC(overIPsec)	24 [0]	3	8		0	12