

ESP Header Compression (EHC)

draft-mgmt-ipsecme-diet-esp-07

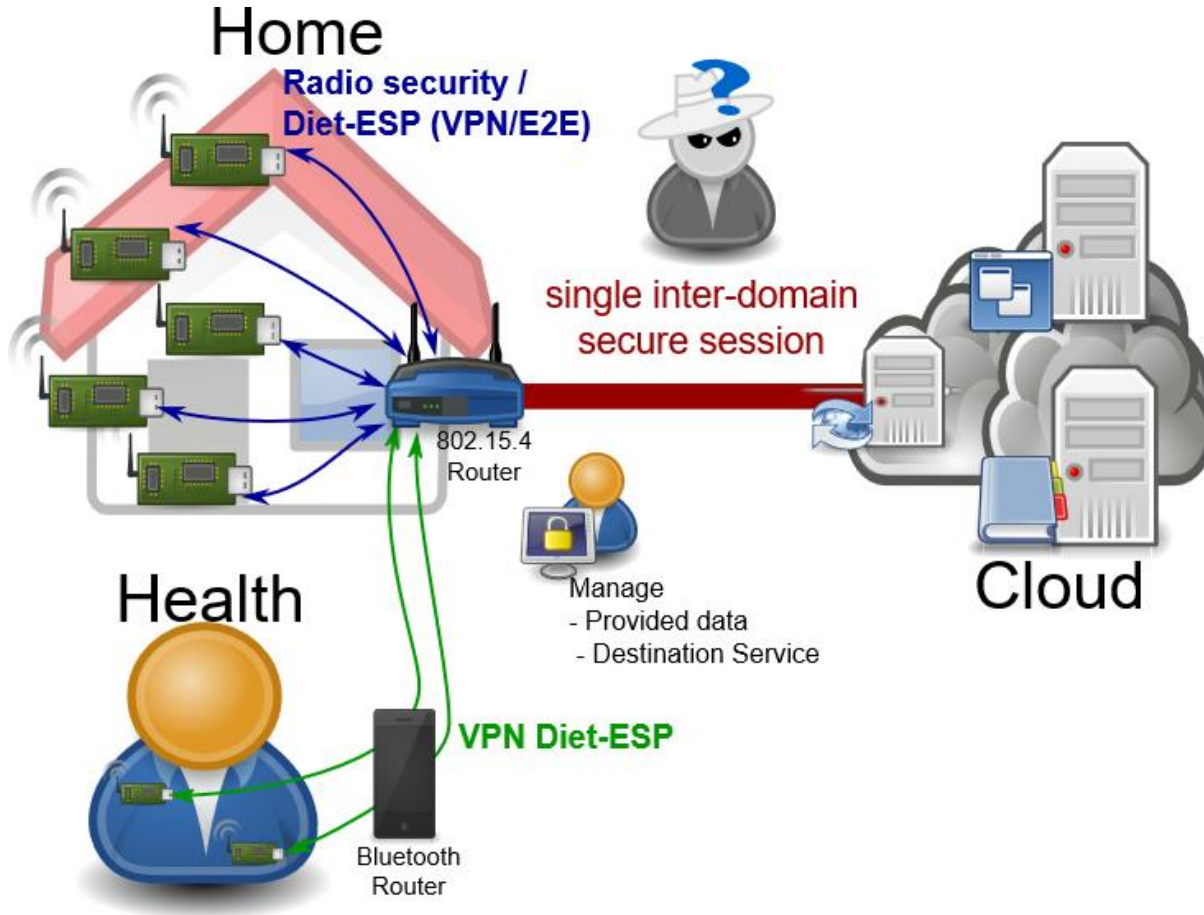
Tobias Guggemos

ipsecme@IETF 104

28.03.2019

Can you imagine a scenario in IoT,
where IPsec/ESP would be useful?

Scenarios for IPsec



- Communication with a server (data center or controller)
- Device to Device (D2D) communication
- Inter-Domain communication (Smart Fabrics/Cities/etc.)
- Multicast / Group Communication
- long term sessionless communications

IPsec's Advantage: Flexibility of Key Exchange

- Minimal IKEv2 is already there
- G-IKEv2 works quite well, too
- HIP is standardized for IEEE 802.15.4
- static SAs

So why is IPsec/ESP not widely deployed in IoT?

Problem: Packet Size

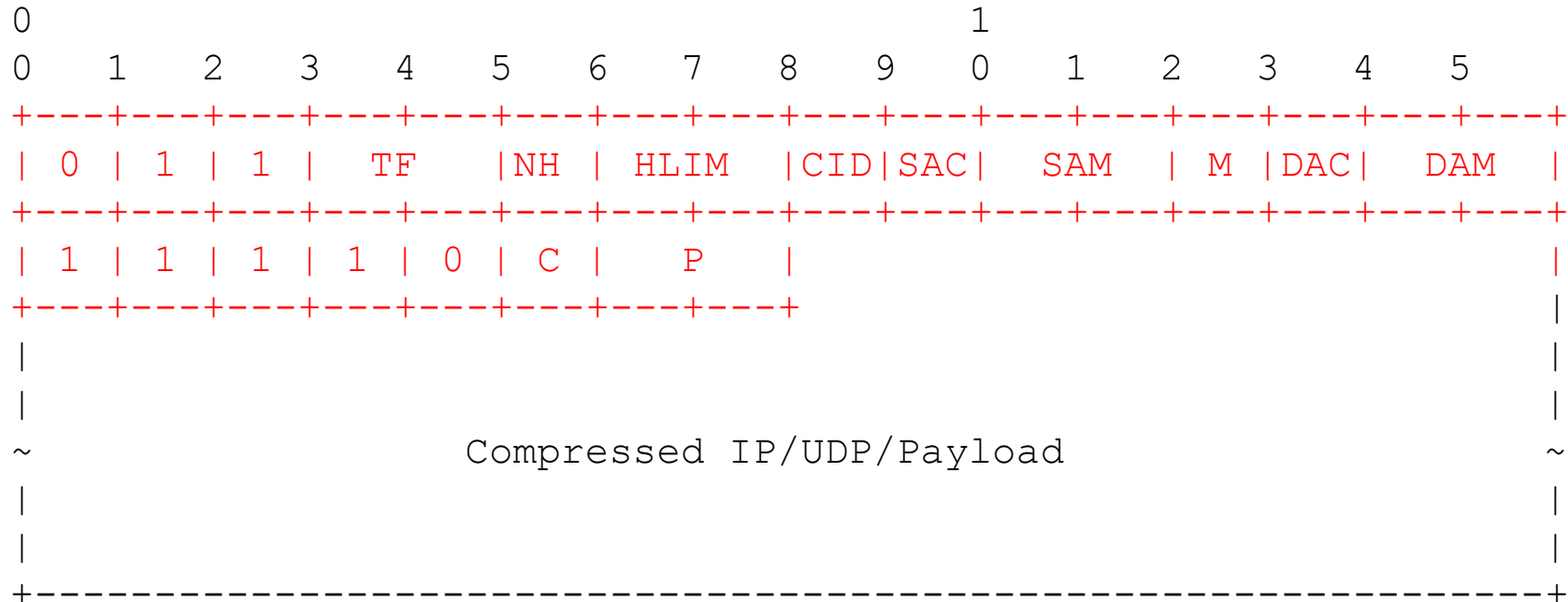
Energy cost of IoT communication:

- Increases with the number of radio frames
- Full / empty radio frames have the same cost
- Security protocol overhead may be larger than a radio frame
- Typical payloads:
 - 802.15.4: 102 Byte
 - LORA: 59 – 230 Byte
 - SigFox: 12 Byte
 - Bluetooth LE: >23 Byte

Solution: Header Compression (1)

Stateless (6LoWPAN)

- Compression Information is sent along with the packet on the wire



Solution: Header Compression (2)

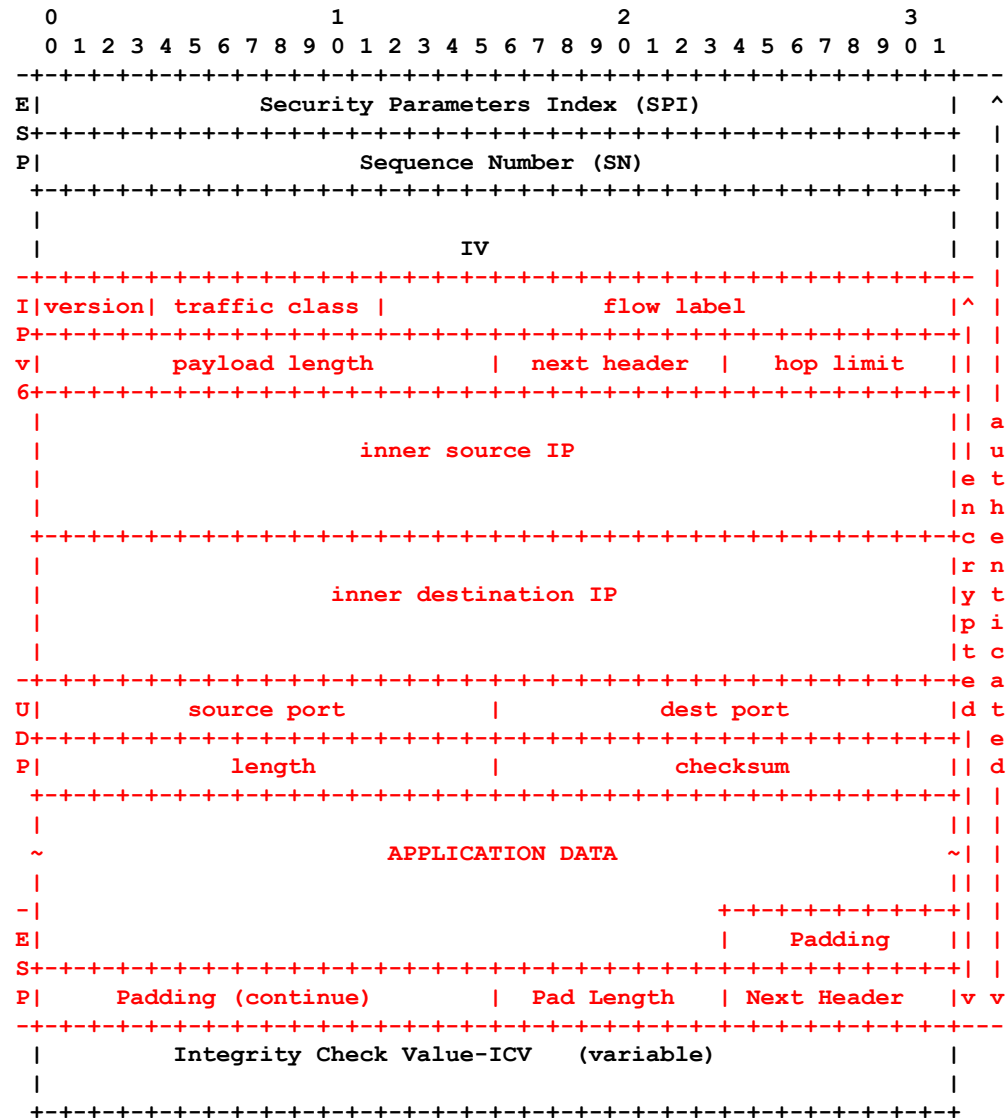
Statefull: RObust HC (ROHC), Static Context HC (SCHC)

- Compression is agreed using a seperate channel
- This can be static (SCHC) or dynamic (ROHC)

Works with functions and context:

- **Functions:** compression technique for a specific field:
 - e.g. „rule 1: delete upd destination port if it is equal to 80“
- **Context:** pre-known knowledge for decompression
 - e.g. „if packet is compressed with rule 1, set the udp destination port to 80“

Problems (with IPsec)



Example of an IP Tunnel

- All shown techniques compress between L2 and L3
- At that time, ESP payload is already encrypted

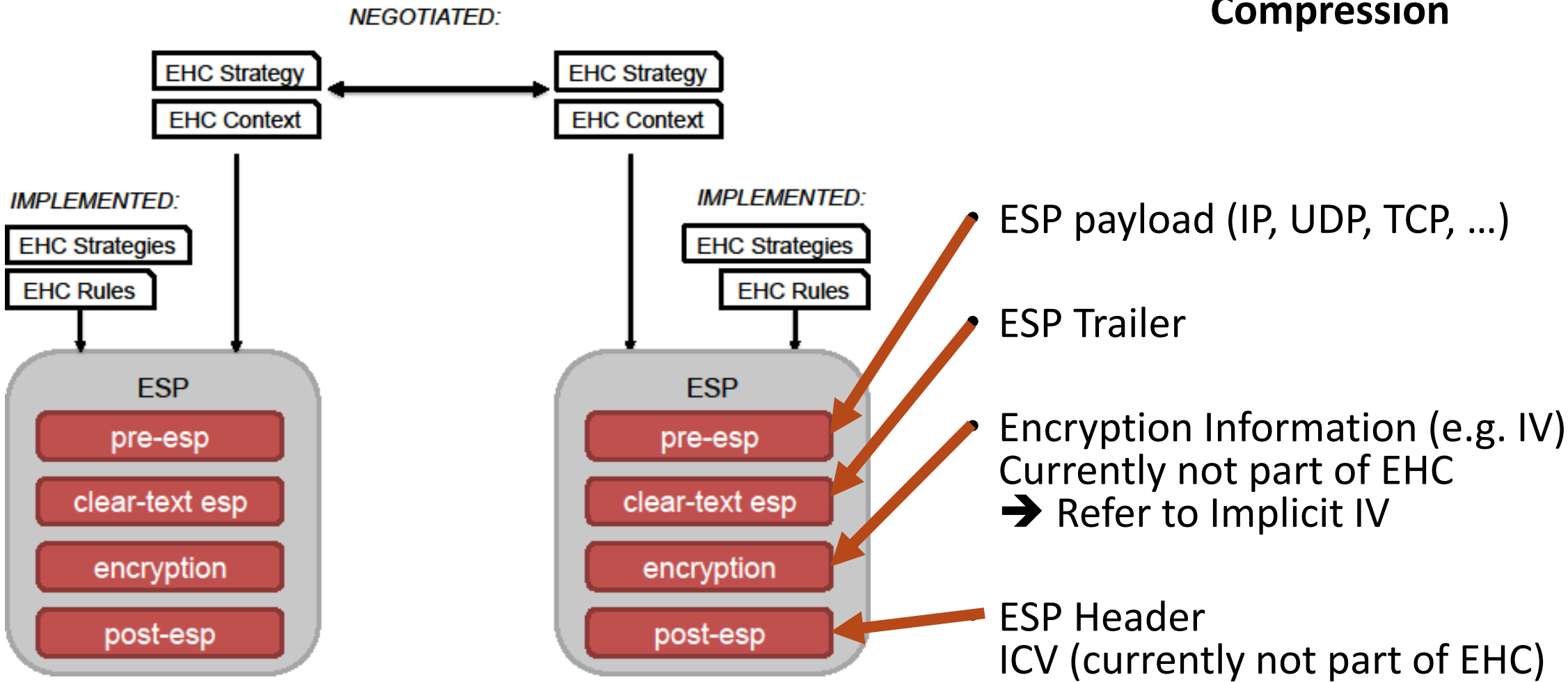
➔ Only ESP header can be compressed without touching the ESP implementation

Good News: ESP Header Compression (EHC)

- ☺ IPsec already has a separate channel to agree on (and update) a state
 - IKEv2, G-IKEv2, (even HIP could be used)
 - Static
 - ☺ IPsec already has a **static** state (IPsec SAs)
 - ☺ The state already holds some context (Traffic Selector)
 - ☺ We have done this before (ROH Cover IPsec RFC5856)
-
- We just need to define how to make use of it!
 - (Unfortunately) this requires a few specifications
- ➔ ESP Header Compression (EHC)
draft-mglt-ipsecme-diet-esp-07

ESP compression Layers

Compression



EHC Actions

| Function | Compression | Decompression |
|-----------------|-----------------|----------------------|
| send-value | No | No |
| elided | Not send | Get from EHC Context |
| lsb(_lsb_size) | Sent LSB | Get from EHC Context |
| lower | Not send | Get from lower layer |
| checksum | Not send | Compute checksum. |
| padding(_align) | Compute padding | Get padding |

- Define the function, how to compress any field
- Derived from ROHC and SCHC specifications

Some examples:

- send-value: IPv4/TCP Options
- elided: IP address, ports
- lsb: ESP/TCP sequence number
- lower: IP/UDP/TCP length
- Checksum: UDP/TCP checksum
- padding: ESP/TCP padding

EHC Rules (example Inner IPv6)

| EHC Rule | Field | Action | Parameters |
|--------------|----------------|--------|------------|
| IP6_OUTER | Version | elided | ip_version |
| | Traffic Class | lower | |
| | Flow Label | lower | |
| IP6_VALUE | Version | elided | ip_version |
| | Traffic Class | elided | ip6_tc |
| | Flow Label | elided | ip6_fl |
| IP6_LENGTH | Payload Length | lower | |
| IP6_NH | Next Header | elided | l4_proto |
| IP6_HL_OUTER | Hop Limit | lower | |
| IP6_HL_VALUE | Hop Limit | elided | ip6_hl |
| IP6_SRC | Source Address | elided | ip6_src |
| IP6_DST | Dest. Address | elided | ip6_dst |

- Map Actions with the available EHC Context
- Defined for every header we consider compressible
 - ESP
 - IPv6 / IPv4
 - UDP / UDP-Lite
 - TCP
 - Anything else?

EHC Context (example: Inner IPv6)

| Attribute | In SA | Possible Values |
|---------------|-------|----------------------------|
| ip6_tcfl_comp | No | "Outer", "Value", "UnComp" |
| ip6_tc | No | IPv6 Traffic Class |
| ip6_fl | No | IPv6 Flow Label |
| ip6_hl_comp | No | "Outer", "Value", "UnComp" |
| ip6_hl | No | Hop Limit Value |
| ip6_src | Yes | IPv6 Source Address |
| ip6_dst | Yes | IPv6 Destination Address |

- Defined for every header we consider compressible
 - ESP
 - IPv6 / IPv4
 - UDP / UDP-Lite
 - TCP
 - Anything else?

EHC Strategy

Problem:

- We still need to exchange (and potentially update) all the context and rules via IKEv2
 - For ESP/IPv6/IPv4/UDP/TCP/UDP-Lite, that's 44 header fields (and thus rules/context)
- More difficult than necessary, as most the valuable values are already there

Solution:

- Strategy defines, how to pre-fill context and rules from available SA values (e.g. IP addresses in Traffic Selector)
- Currently one Strategy for IoT (Diet-ESP) defined, but easy to extend
- ➔ Exchanges only max. 9 fields (instead of 44!) to build a compression context

