Enhanced ESP
IPsec Workshop 2025

draft-ipsecme-eesp

Steffen Klassert

Why a new security protocol?

- ESP lacks flexibility and extensibility for modern needs
  - No version number in packet
  - New features must be negotiated
  - Not transparent to the network
- Lot of proposals to extend ESP
  - Suffer from ESP limitations

» **Extending ESP can’t fit the requirements!**

Requirements

EESP Requirements I (high level)

- High Speed Networks
- AI Workloads
- Datacenter Networks
- High Performance Computing
- SDN/SD-WAN Networks
- Cloud Computing
- Machine Lerning
- Internet of Things

EESP Requirements I (high level)

- High Speed Networks
- AI Workloads
- Datacenter Networks
- High Performance Computing
- SDN/SD-WAN Networks
- Cloud Computing
- Machine Lerning
- Internet of Things

» A slide full of Buzzwords :-)

EESP Requirements (tech)

- Performant in HW and SW
- Parallelizable (HW + SW)
- No additional SA negotiation
- Fast lookups
- Reduce fast memory usage (HW)
- Support replay protection
- QoS with minimal overhead
- Enable SDN / Telemetry
- Must coexist with ESP
- Futureproof (extensible)

Design

Performance I

Sub SAs and Session ID

- Sub Child SA ID encoded on Session ID
- Parallelizable with replay protection
  - Separate sub SA for:
    - Each CPU core
    - Ech QoS class
- No extra negotiation needed
  - Keys are generated by a KDF
  - Keys can be generated ‘on the fly’ by HW
    - No need for additional fast memory

» **Parallelization with minimal overhead!**

Performance II

Packet Format

- No Trailer
  - Transport Mode:
    - Trailer info placed in front of the L4 payload
  - Tunnel Mode
    - Info can be derived from the IPv4/IPv6 header
- Only AEAD algorithms
  - Can use implicit IV (on the seq. nr.)
- Full 64 bits sequence number in the header
  - No implicit parts needed

» **Easy to parse the packet!**

Performance III

Group Keys and Key Derivation

- draft-xia-ipsecme-eesp-stateless-encryption-01
- Discussion later!

» **Significant reduction of negotiation and SPD/SADB state overhead!**

Futureproof

Flexibility / Extensibility

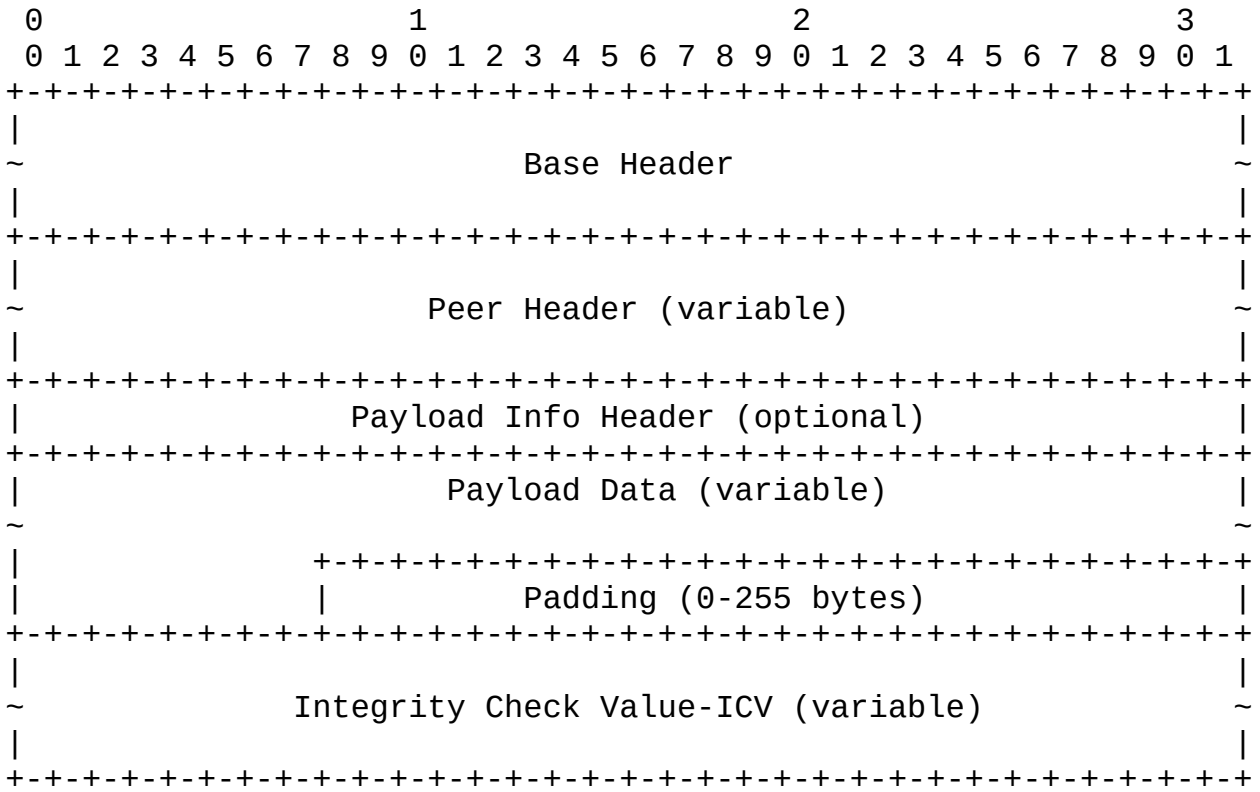
- 4 bits Version Number
- Variable Header Options possible
    - IPv6 extension header style

EESP Requirements (again)

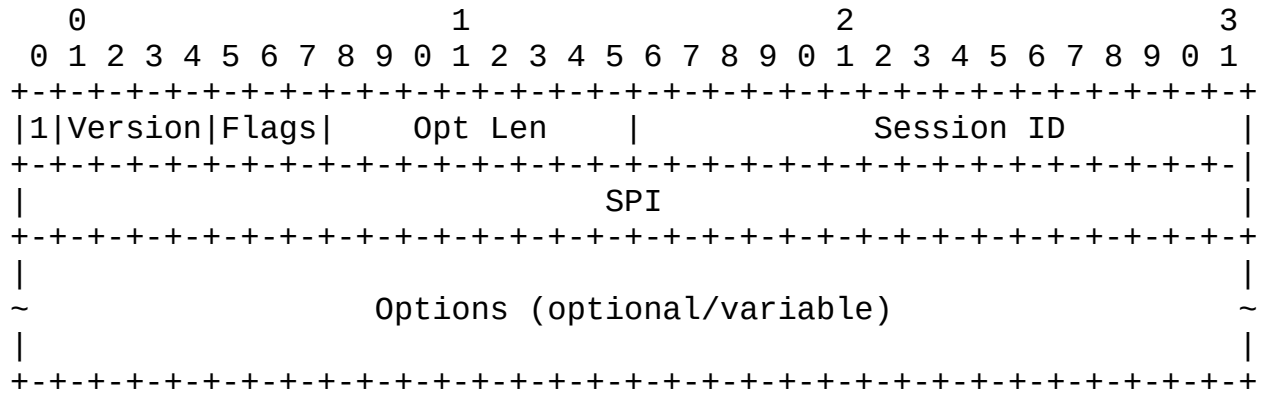
- Performant in HW and SW ✓
- Parallelizable (HW + SW) ✓
- No additional SA negotiation ✓
- Fast lookups ✓
- Reduce fast memory usage (HW) ✓
- Support replay protection ✓
- QoS with minimal overhead ✓
- Enable SDN / Telemetry ✓
- Must coexist with ESP (IKEv2 ✓)
- Futureproof (extensible) ✓

EESP Packet Format

Toplevel Packet Format

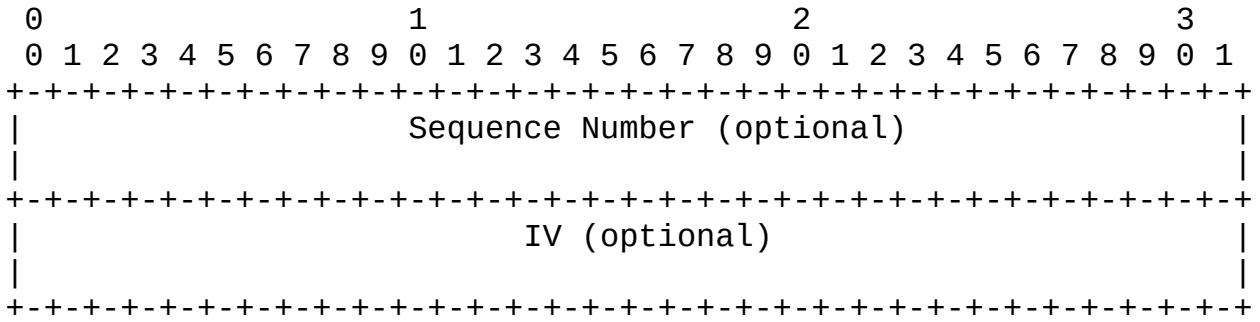


EESP Base Header with Options



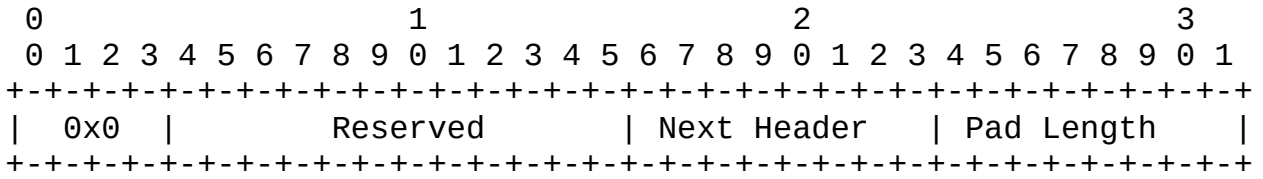
- 1: ESP/EESP distinguisher
- Version (4 bits)
- Flags
  - 1 bit – Packet Format
  - 2 bits – Reserved
- Opt Len: Overall Options length
- Session ID (16 bits)
- SPI: same as in ESP
- Options: TLV encoded

EESP Peer Header



- Sequence Number: 64 bit, if present
- IV: Depends on algorithm type
    - Optional because implicit IV can be used (RFC 8750)

Full Packet Format - Payload Info Header



- **Required for Transport, BEET and IPTFS**
- Can’t derive “Next Header” from outer IP (set to ESP protocol)
- Can’t determine “Pad Length” from outer IP length
  - IP Length (Total) == IP+EESP+Payload+Pad+ICV

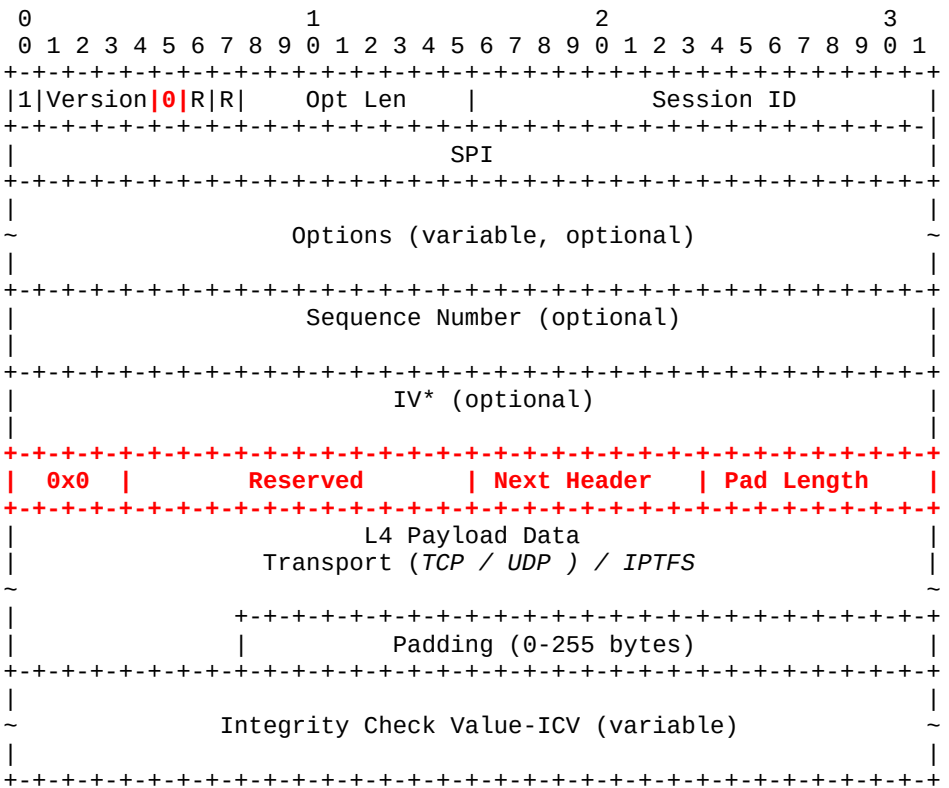
Payload Info Header depends on the Mode

Optimized Packet Format

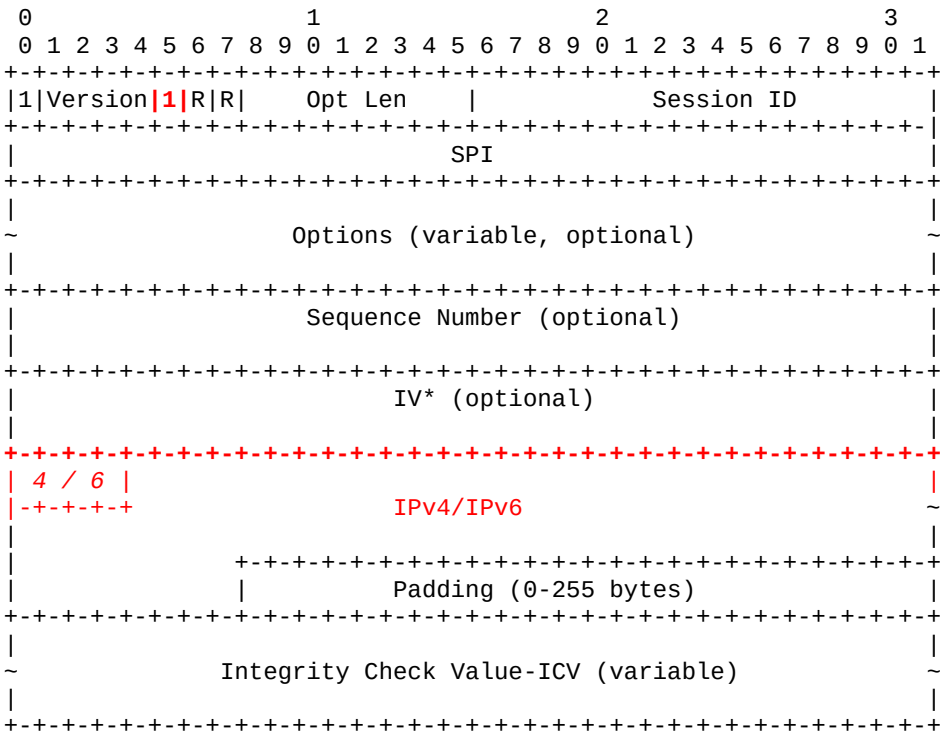
- **Payload Info Header Not Needed for IPv4/IPv6 Tunnel Mode**
- First Nibble defines payload (0x4 or 0x6 IP version value)
- “Pad Length” computed using outer length and inner length fields
- “Next Header” not needed in tunnel mode

Format Comparison

Full Packet Format

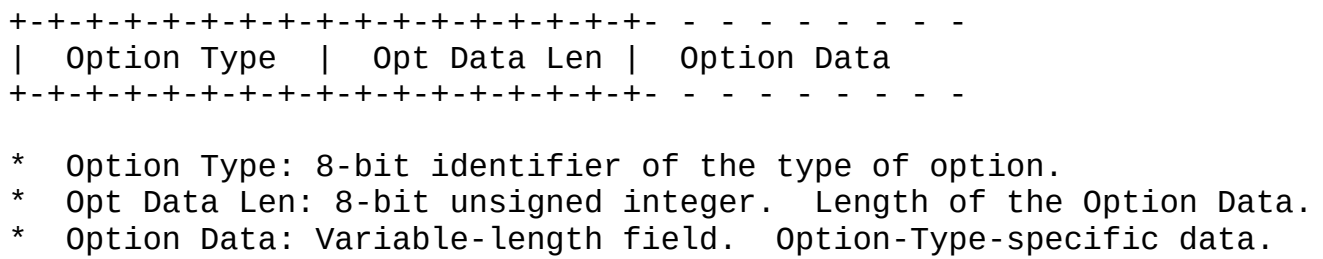


Optimized Packet Format



EESP Options

EESP Options

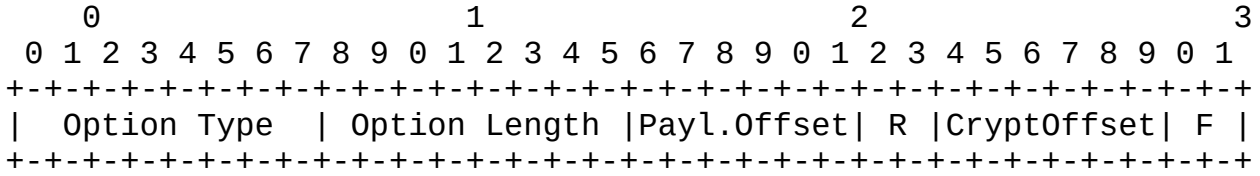


- Options are Type Length Values (TLV)
- Adapted from IPv6 Extension Header Options (RFC 8200 Section 4.2)
- Multiple Options can follow the header
- Future documents can define new Options
- Reserved space for private Options

Option Types

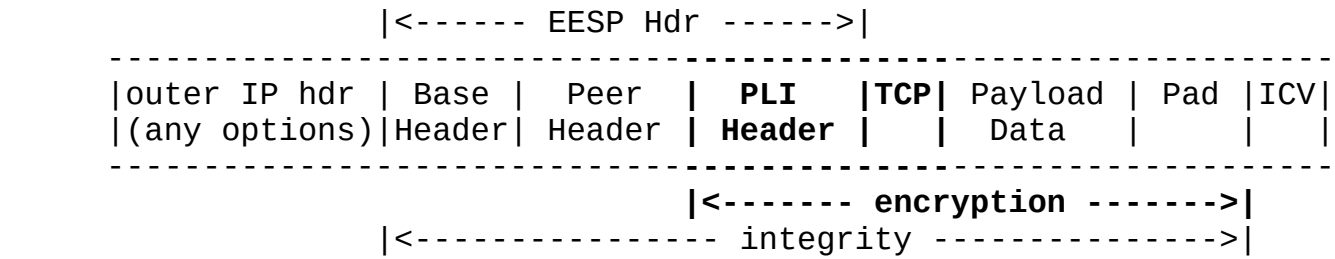
- Crypto Offset
- Flow Identifier
- Padding
- Future documents can define new Option Types

Crypto Offset Option

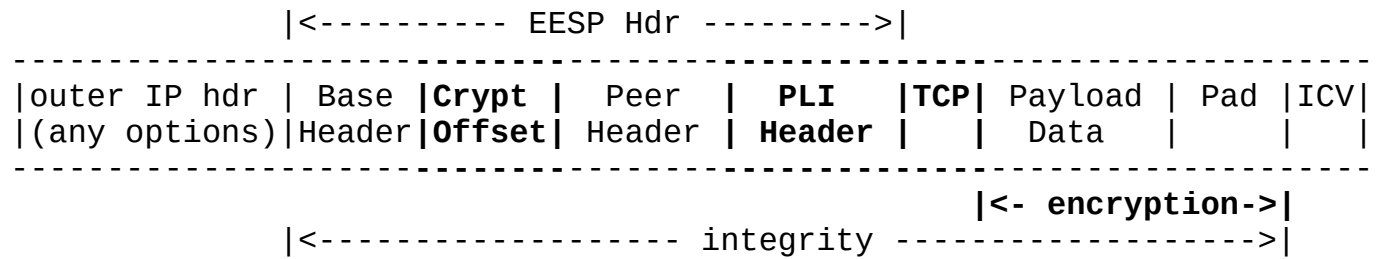


- Payload Offset – Start of the inner packet
- R – Reserved
- CryptOffset – Offset left unencrypted at beginning of the inner packet
- F – Flags (e.g. for PSP usecase)
- For telemetry usecase present just on some (not all) packets (PSP)!
- UEC TSS has crypto offset not on the packet (just negotiated)!
  - Why?
  - Use case?

Crypto Offset (Full Packet Format)

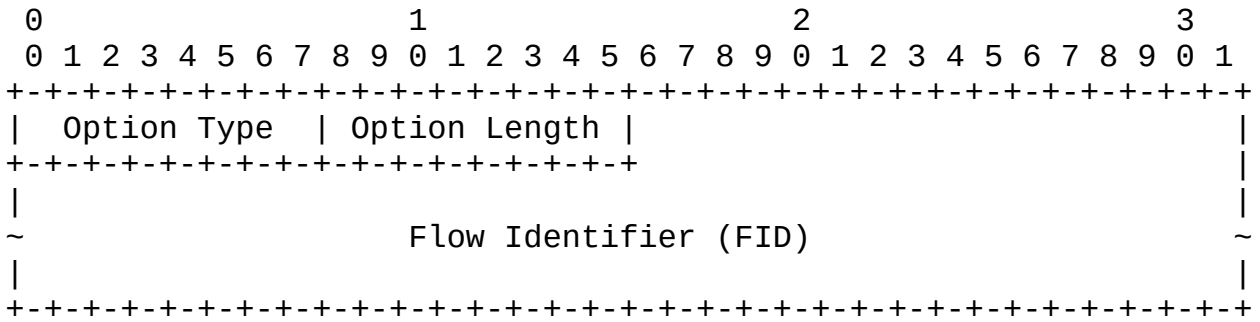


NO CryptOffset Option (Full Packet Format)



CryptOffset Option present (Full Packet Format)

Flow Identifier Options



- Carries characteristic information of the inner flow
- Initial idea: Can be used by intermediate devives (ECMP, RSS)
  - But: Session ID does that now (ECMP, RSS)!
  - FID for ECMP, RSS does not work with replay protection
- Can still be use as a VNI (Geneve like)
-
  -

Padding Options

- Pad1, PadN adapted from RFC 8200
- Used to align following header (4 byte IPv4, 8 byte IPv6)
- Future usecase: Ciphertext alignment (for SIMD, AVX)

Major changes to ESP

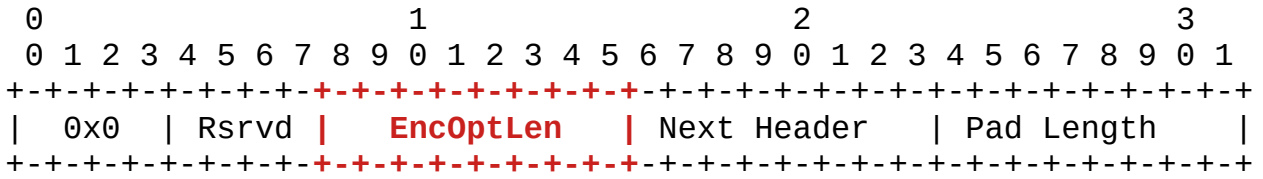
- Adds Version Number (no need for new IP protocol numbers)
- 8 byte base header
- Session ID
- Variable Header-Options possible
- Transmit 64 bit SeqNo (always use ESN)
- Make SeqNo optional
- Two packet formats (full/optimized)
- Remove implicit header parts

Major changes to ESP

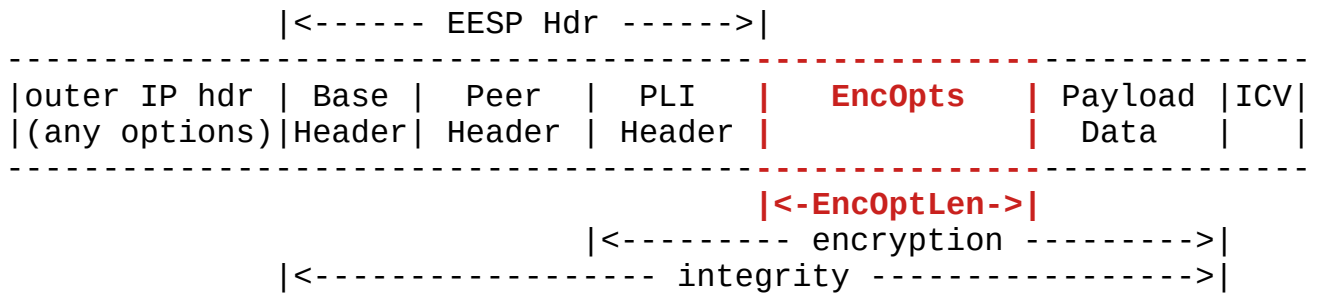
- Remove the trailer
- Remove TFC padding, use IPTFS instead
- AEAD algorithms only

Some further Ideas

Additional Encrypted Payload Info

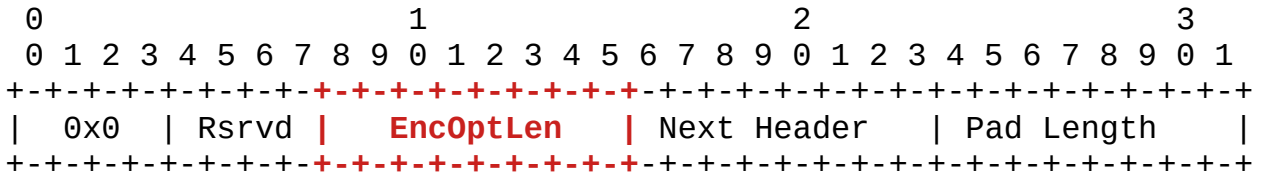


Payload Info Header with Encrytped Option Length Field



Packet with Encrypted Options

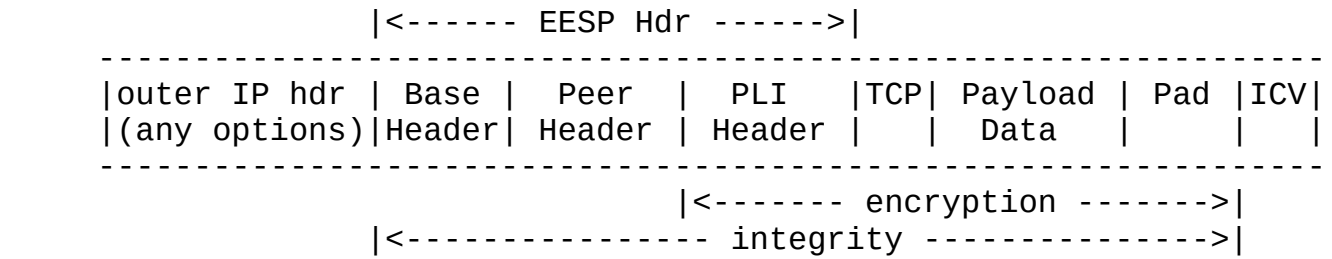
Aditional Encrypted Payload Info



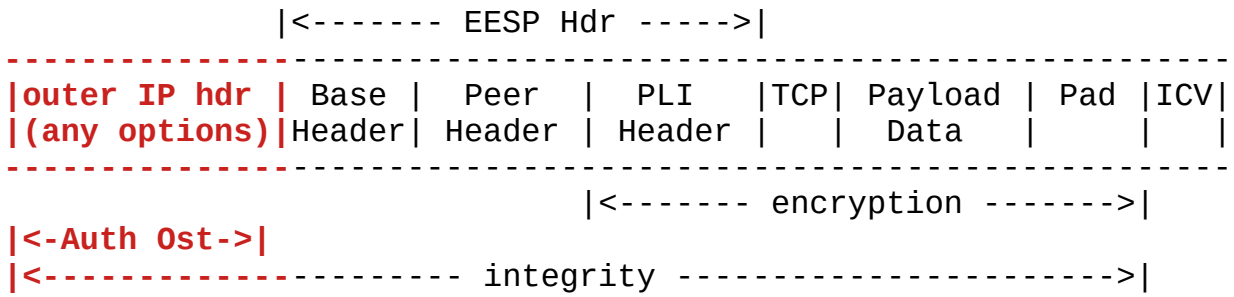
Payload Info Header with Encrytped Option Length Field

- IKEv2 Negotiated Encrypted Options
- Need a registry for Encrypted Option identification?!
  - Packet engine should not need to consider IKEv2
- If wanted: Intergrate into core EESP or new draft?

Authentication Offset (Full Packet Format)

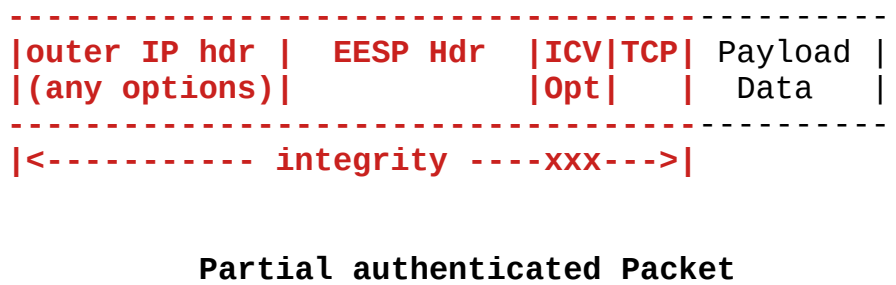


NO Authentication Offset Option (Full Packet Format)



Authentication Offset Option present (Full Packet Format)

Dynamic Authentication (AH like)



- Inspired by: draft-dunbar-ipsecme-lightweight-authenticate

Questions?

Answers?