IKEv2 Negotiation for EESP

IPsec Workshop 2025

Tobias Brunner

18.07.2025

codelabs

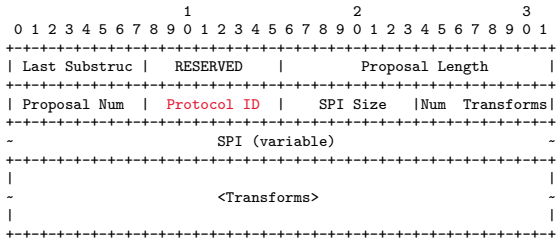
High Assurance Systems

Overview

- Basic Negotiation
- Sub SAs
- Sequence Numbers

Basic Negotiation

- EESPv0 is negotiated like ESP or AH
- New Security Protocol Identifier (<TBD>) in proposals of SA payload



- Initiator may propose EESPv0 and ESP in same SA payload

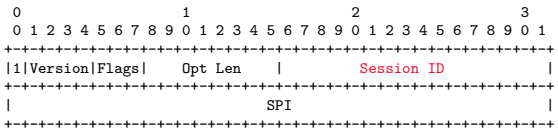
codelabs

Transforms Overview

- Two mandatory transform types: ENCR, SN
- Two optional types: KE, SSKDF
- Only AEAD algorithms allowed for ENCR transform type

Sub SAs

- Distinguished via Session ID in EESP Base Header



- Enabled by negotiating a Sub SA Key Derivation Function (SSKDF)
- Currently defined via HKDF-Expand, possibly via 256-bit AES-CMAC

Key Derivation for Sub SAs

- Every Sub SA (including Session ID 0) uses separate keys
- A "root" key is derived by IKEv2 for each EESP SA of the Child SA:

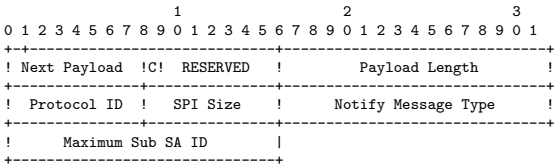
```
KEYMAT_root = prf+(SK_d, [g^ir (new) |] Ni | Nr)
KEYMAT_root_i | KEYMAT_root_r = KEYMAT_root
```

  Its size depends on the key size of the SSKDF
- Root key and selected SSKDF is passed to EESP
- Keys for individual Sub SAs are then derived by EESP:

```
KEYMAT_sub = SSKDF(KEY_root_i/r, Session ID, L)
```

Notify for Maximum Sub SA ID

- Notify indicates the maximum Sub SA ID the sender accepts



- Peer MUST select IDs for outbound Sub SAs from 0 to that limit (inclusive)
- Not negotiated, each side specifies their own limit
- Omitting the notify = no upper limit

Sequence Numbers

- Two new values for Sequence Numbers transform type
  - "64-bit Sequential Numbers": Full SN encoded in EESP header
  - "None": No SN encoded in EESP header
- Existing values (32-bit and partial 64-bit SN) unspecified for EESPv0
- These new values MUST NOT be used with ESP
- "None" is not allowed with AEAD algorithms that use implicit IVs

References I

- Latest Draft Version.
  https://klassert.github.io/eesp-ikev2/draft-ietf-ipsecme-eesp-ikev2-latest.html.
- Repository.
  https://github.com/klassert/eesp-ikev2.