# Stateless Encryption and Key Management

**Valery Smyslov**

IPsec Workshop, Madrid, July 2025

# Stateless Encryption

- Popular term, but somewhat misleading
- To process a packet you need:
  - packet encryption/authentication key(s)
  - encryption/authentication algorithm(s) and mode(s) of operation
  - some other information that may depend on mode(s) (like IV)
  - some additional information for packets processing (like SN)
- Let us call this information a state
- For an outgoing packet the state must be either retrieved from a storage or created as prescribed by a policy; in both cases it is based on characteristics of the packet
- For an incoming packet the state must be retrieved from a storage or created (or both); in both cases action is based on information from the security protocol header in the packet

# ESP

- Encapsulating Security Payload (RFC 4303)
  - SA (Security Association) is a state
  - packets contain only SA identifier (SPI)
  - the scope of the state is identical for outgoing and for incoming traffic and covers only packets with some particular characteristics, as dictated by the policy
  - if get lost, state can be re-created (with some restrictions) for any outgoing packet
  - if get lost, state cannot be re-created on receiving side based only on information from the incoming packet (SPI)
- Properties
  - statefulness
  - properties mostly depend on key-management protocol; with IKEv2
    - algorithm negotiation and agility
    - PFS
    - full replay protection

# SKIP

- Simple Key-Management for Internet Protocol by Sun Microsystems (draft-ietf-ipsec-skip-06)
  - SKIP allows re-creating of a full state from any incoming packet, since SKIP header contains
    - algorithm identifiers
    - peer's identity
- Properties
  - the only peer-related information that the host must have is a certificate that certifies peer's DH key. This is usually long lived (months or years)
  - state creating is slow, actually the protocol was implemented as stateful with some heuristic state management
  - no algorithm negotiation
  - no PFS in base protocol
  - limited replay protection
  - problems with counter-mode encryption (including modern AEAD ciphers)

# PSP

- PSP Security Protocol by Google
  - uses concert of ESP SA, but with some differences
  - for outgoing traffic the scope of a crypto state is identical to ESP
  - for incoming traffic a single master key is used to derive encryption keys for each active SA
    - algorithm identifiers are transmitted in each packet
    - for any incoming packet state can be obtained based on some global meta-state (current muster key) and information from the packet - SPI and algorithms
  - master key can be treated as meta-state for incoming traffic; this provides some “statelessness”, actual crypto state is computed for every incoming packet
  - mandatory UDP encapsulation
- Properties
  - some statelessness for incoming traffic (no SA lookup before decryption)
  - no replay protection
  - PFS is limited (master key compromise gives access to all SAs with this host for a mater key lifetime)
    - master keys must be changed frequently enough, but this requirement is in conflict with statelessness of the protocol
  - no secret salt in AEAD nonce (perhaps does not matter)
  - performance penalty for small (or forged) packets
  - limited algorithm agility
  - IKEv2 cannot be used for key management w/o modification

# TSS

- Transport Security Sublayer by Ultra Ethernet Consortium (UEC)
  - complex and feature-reach
  - uses concept of Secure Domain as a crypto state
  - the scope of a crypto state varies depending on the used KDF mode (direct, cluster, server)
  - supports various ways of re-keying (implicit & explicit) as well as key rotation
  - may support group communications
  - UDP encapsulation is optional, load balancing is supported even w/o it
- Properties
  - some statelessness for incoming traffic, but less than in PSS (state lookup before decryption is needed)
  - limited replay protection (based on epoch)
  - PFS depends on KDF mode and on key management
  - no secret salt in AEAD nonce (perhaps does not matter)
  - IKEv2 cannot be used for key management w/o modification
  - Group Key management may be needed for Secure Domains

# EESP

- Support for “stateless” encryption is planned
- Details are not yet defined

# What “Stateless” Encryption Is

- State remains, but its scope is changed
- State is more coarse-grained (up to the host level) and contains some kind of “master” key
  - keys for individual packet flows (e.g. SAs) are derived from the master key via KDF
  - some per-packet-flow information (e.g. concerned with replay protection) is sacrificed
- Scopes may differ for incoming and outgoing traffic
  - PSP: global scope for incoming, fine-grained for outgoing
  - TSS with server KDF: fine-grained on client, coarse-grained on server

# Security Issues

- It seems that replay protection is not possible (or is limited)
  - increased surface for DoS attacks
  - may affect security of upper protocols that rely on replay protection
- Using “master” keys means that there is no strict key separation between SAs - a single (master) key compromise may result in a huge loss of confidentiality
  - PFS is limited
- No origin authentication with group keys
  - any group member can impersonate any other group member

# Key Management Issues

- In many (all?) cases “stateless” encryption requires that raw keys are transferred by KMP
- IKEv2 does not support transferring raw keys
  - no appropriate payload
  - raw keys are sensitive information, must be protected inside IKE messages
  - when initial Child SA is being created in IKE_AUTH, raw keys cannot be sent in the request message, since the responder is not yet authenticated

# Possible KMPs

- IKEv2 can be modified to support transfer of raw keys
  - use childless IKE_AUTH, create Child SAs only in CREATE_CHILD_SA when both peers are authenticated
  - define new KE Method “wrapped_raw_key”
  - use wrapping mechanism from G-IKEv2 (using new SK_w key)
  - reuse Key Wrap Algorithm transform from G-IKEv2
  - transfer wrapped raw keys in the KE payload
  - suited for PSP, may be not suited for TSS with “server” KDF mode, when a server is responsible for providing keys for both directions
- G-IKEv2 can be used for managing keys for group communications
  - while G-IKEv2 assumes that group has a multicast IP, this is not a strict requirement
  - if group has a multicast IP then it may make key management more effective for large groups
  - applicability for KDF modes other then “direct” in TSS should be evaluated for other KDF modes (looks feasible, but more complicated)

Thank you!