FIT5037: Network Security
Security at Network layer (IPsec)

Faculty of Information Technology
Monash University

Copyright Notice

Lecture 5: Security at Network layer (IPsec)

Lecture Topics:

- Symmetric key cryptography
- Asymmetric key cryptography
- Pseudorandom Number Generators and hash functions
- Authentication Methods and AAA protocols
- **Security at Network layer**
- Security at Network layer (continued)
- Security at Transport layer
- Security at Application layer
- Computer system security and malicious code
- Computer system vulnerabilities and penetration testing
- Intrusion detection
- Denial of Service Attacks and Countermeasures / Revision

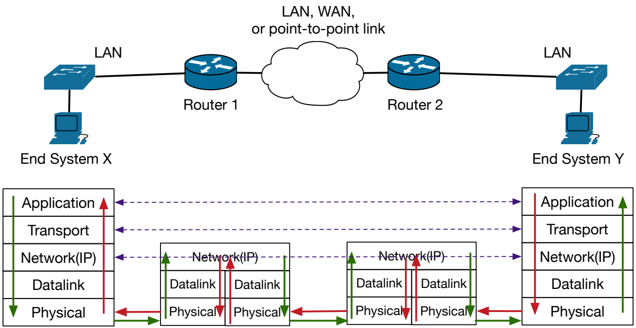
MONASH University

Outline

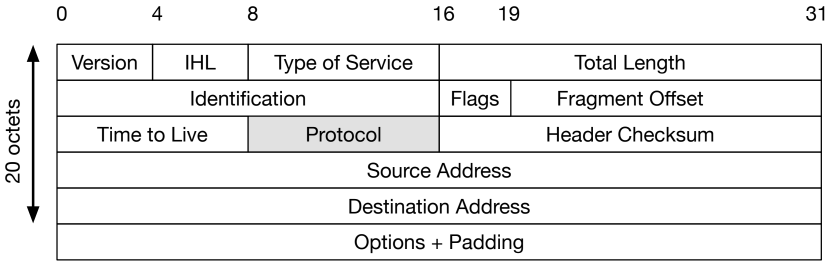
- Brief overview of TCP/IP protocol stack and IPv4 and IPv6 headers
- IPsec Architecture
- ESP Protocol
- Internet Key Exchange Protocol

TCP/IP Protocol Stack

- Internet Protocol (IP) is implemented at Network layer of TCP/IP protocol stack
- End systems and all intermediate routers implement IP
- Packets are routed based on destination IP address
- No built-in security feature in IP



IPv4 Header



0 4 8 16 19 31

20 octets

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

Time to Live | Protocol | Header Checksum

Source Address

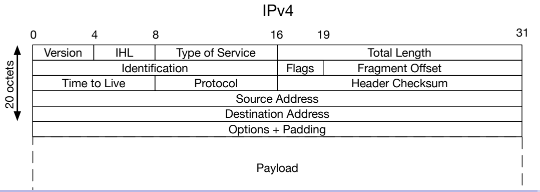
Destination Address

Options + Padding

- Header length = 20 octets
- Some fields may change on route, i.e., ‘Time to Live’, ‘Fragment Offset’
- ‘Header Checksum’ is meant to protect header from corruption
- But hackers can purposely change header fields as well as checksum
- Protocol field specify what protocol runs on top of IP

IPv6 Header

- Header length = 40 octets
- Next Header field tells the *type* of the next header
  - for an IPv6 packet without any header extension this filed indicates the protocol that runs on top of IPv6.
  - If the packet contains options, this field contains the *type* of the next option (next extension header).
  - The *Next Header* field of the last option, points to the upper-layer protocol that is carried in the packet's payload.
- IPv6 Optional Extension headers carry options that are used for special treatment of a packet in the network, e.g., for routing, fragmentation, and for security using the IPsec framework.



IPv4

0 4 8 16 19 31

20 octets

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

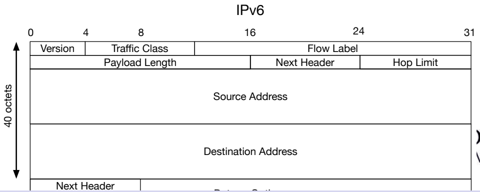
Time to Live | Protocol | Header Checksum

Source Address

Destination Address

Options + Padding

Payload



IPv6

0 4 8 16 24 31

40 octets

Version | Traffic Class | Flow Label

Payload Length | Next Header | Hop Limit

Source Address

Destination Address

Next Header

IPsec

- IPsec is a set of protocols to provide security services for traffic at the IP layer
  - the security services can be provided to all protocols that run on top of IP
- IPsec Provides
  - access control (traffic selectors)
  - authentication
  - confidentiality
  - key management (key derivation)
- key exchange (negotiation) is performed by IKEv2 protocol (outside IPsec)
- applicable wherever IP packets can be routed
  - over LANs, across public and private WANs, and Internet
- IPsec architecture is defined in RFC 4301

IPsec Modes of Operation

IPsec operates in two modes: Transfer Mode and Tunnel Mode

- **Transfer Mode**: The source and destination hosts must directly perform all cryptographic operations. Encrypted data is sent through a single tunnel that is created with L2TP (Layer 2 Tunneling Protocol). Data (ciphertext) is created by the source host and retrieved by the destination host. This mode of operation establishes end-to-end security
- **Tunnel Mode**: Special gateways are used in order to perform cryptographic processing in addition to the source and destination hosts. Many tunnels are created in series between gateways, establishing gateway-to-gateway security.
- In both modes, it's important to provide all gateways with the ability to verify that a packet is real and to authenticate the packet at both ends

IPsec Data Encodings

Two types of data packet encodings (DPE) are required in IPsec.

- **Authentication header (AH)**:
  - provides authenticity and integrity of the packet. The authentication is made available through keyed hash functions i.e. MACs (message authentication codes).
  - prohibits illegal modification and has the option of providing antireplay security.
  - can establish security between multiple hosts, multiple gateways, or multiple hosts and gateways, all implementing AH
- **Encapsulating Security Payload (ESP)**
  - The ESP header provides encryption, data encapsulation and data confidentiality. Data confidentiality is made available through symmetric key

Security Association

An association is a one-way logical connection between a sender and a receiver that affords security services to the traffic carried on it. Uniquily identified by:

- Security Parameter Index (SPI)
- IP Destination Address
- Security Protocol Identifier

Other IPsec Entities

- **Security parameter index (SPI):** The SPI specifies the algorithms and keys that were used by the last system to view the packet.
  - Any change or error in the data will be detected, causing the receiving party to drop the packet.
  - The headers are applied at the beginning of each tunnel and then verified and removed at the end of each tunnel.
- **Security association (SA):**. The SA uses the SPI number that is carried in the AH and ESP to indicate which SA was used for the packet
  - An IP destination address is also included to indicate the endpoint:
    - Firewall, Router or End point
- **Security Association Database (SAD):** stores all SAs that are used
  - A security policy is used by the SAD to indicate what the router should do with the packet. All security policies are stored in the **Security Policy Database (SPD)**

Benefits of IPsec

- When implemented in a firewall/router
  - provides strong security to all traffic crossing the perimeter
  - is resistant to any traffic bypass
- It is implemented below transport layer, hence transparent to applications
- transparent to end users (implemented in security gateways)
  - secure branch office connectivity over the Internet (network to network)
- can provide security for individual users (implemented in host systems)
  - secure remote access over the Internet (host to network, host to host)
- can shift the cost of performing cryptographic operations from servers and users' systems to dedicated devices
- secures routing architecture
  - a router or neighbour advertisement comes from an authorised router
  - a redirect message comes from the router to which the initial packet was sent
  - a routing update is not forged

IP Security Architecture

specifications are defined in RFC documents:

- RFC 4301 Security Architecture for Internet Protocol
- RFC 4302 IP Authentication Header (AH)
- RFC 4303 IP Encapsulating Security Payload (ESP)
- RFC 7296 Internet Key Exchange (IKEv2) Protocol
- RFC 8221 Cryptographic Algorithm Implementation Requirements and Usage Guidance for ESP and AH
- Other

IPsec Services

AH Services:

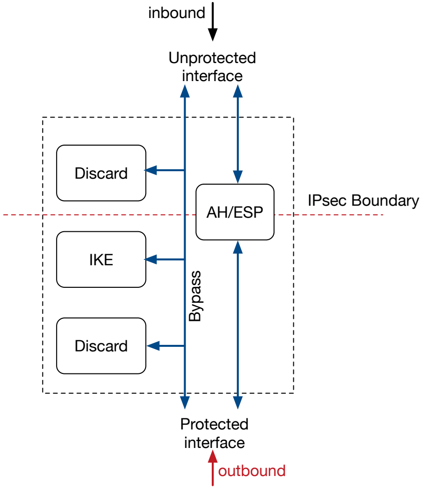
- Authentication of IP protocol header
- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets

ESP Services:

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets
- Confidentiality
- Limited traffic flow confidentiality

IPsec Traffic Processing Model

- *inbound*: traffic entering via unprotected interface directed towards the protected interface
- *outbound*: traffic entering via protected interface directed towards the unprotected interface
- *bypass*: traffic not processed by IPsec



IP Security Policy

- Security Policy Database (SPD) management allows specifying
  - which *security protocol* to use: AH or ESP
  - which *mode* to use: Tunnel or Transport
  - what cryptographic algorithms to use
  - in what combination to use specified protocols and services
  - at what granularity the protection should be applied

Security Associations (SA)

- SA is a one-way relationship between sender and receiver that affords security for traffic flow
  - for a bi-directional communication a pair of SAs (one in each direction) is required
- SA is defined by 3 parameters:
  - Security Parameter Index (SPI)
  - IP Destination Address
  - Security Protocol Identifier AH or ESP
- SA has a number of other parameters
  - Seq. no, AH and ESP info, lifetime etc.
- established SAs are kept in a database of Security Associations: SAD
- Security Policy Database (SPD) defines the security policies to be applied
  - SA identifier links the SAD and SPD
- IKE protocol establishes and maintains SAs

Security Association Database (SAD)

- **Security Parameter Index**: A 32-bit value selected by the receiving end of an SA to uniquely identify the SA.
- **Sequence Number Counter**: A 64-bit (extended) value used to generate the Sequence Number field in AH or ESP headers.
- **Sequence Counter Overflow**: A flag indicating whether overflow of the Sequence Number Counter should prevent further transmission of packets on this SA.
- **Anti-Replay Window**: A 64-bit counter used to determine whether an inbound AH or ESP packet is a replay.
- **AH Information**: Authentication algorithm, keys, key lifetimes, and related parameters being used with AH.
- **ESP Information**: Encryption and authentication algorithm, keys, initialisation values, key lifetimes, and related parameters being used with ESP.
- **Lifetime of this Security Association**: A time interval or byte count after which an SA must be replaced with a new SA (and new SPI).
- **IPsec Protocol Mode**: Tunnel or transport.
- **Path MTU**: Maximum size of a packet that can be transmitted without fragmentation.

Security Policy Database (SPD)

- specifies what services are to be offered to IP packets and in what fashion
- SPD is an ordered database consistent with the use of Access Control Lists (e.g. in packet filter firewalls)
- processing choices are:
  - DISCARD: not allowed to traverse IPsec boundary
  - BYPASS: is allowed to traverse without IPsec protection
  - PROTECT: is afforded IPsec protection
- SPD is logically divided into three parts:
  - SPD-I: applies to inbound traffic that will be bypassed or discarded
  - SPD-O: applies to outbound traffic that will be bypassed or discarded
  - SPD-S: secure traffic which contains entries for all traffic subject to IPsec protection

SPD Selectors

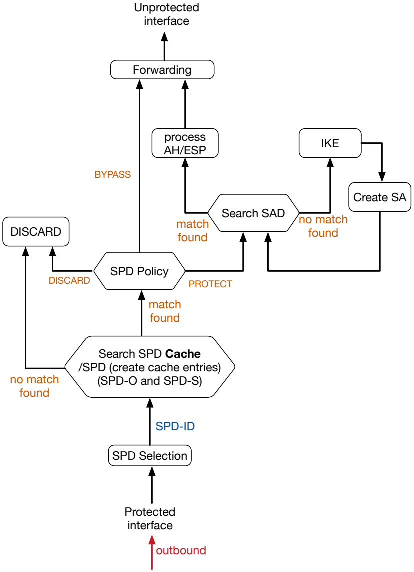
- Selectors are used to define the set of traffic for an SA
- Remote IP Address(es)
  - a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one destination system sharing the same SA (e.g., behind a firewall).
- Local IP Address(es)
  - a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one source system sharing the same SA (e.g., behind a firewall).
- Next Layer Protocol
  - designates the protocol operating over IP
    - Protocol field for IPv4
    - Next Header for IPv6 or IPv6 Extension
  - an individual protocol number, ANY, or OPAQUE for IPv6 only
- Name
  - different from the other selectors (not acquired from a packet)
  - used by a responder when IP address is not appropriate (e.g. road warriors)
    - matched against a filed in IKE negotiation in ID payload --> may be used by initiator to identify a user

Peer Authorization Database (PAD)

- PAD provides the link between SPD and a SA management protocol such as IKE
- identifies peer(s) allowed to communicate with this entity
- specifies protocols and method of peer authentication
  - e.g. IKEv1, IKEv2
- provides authentication data for each peer
  - pre-shared secrets, or certificates
  - for certificate-based authentication may provide information to assist in verifying the revocation status of the peer
- constrains the types and values of IDs that can be asserted by a peer with regard to Child SA creation
  - whether the IKE ID payload will be used for SPD lookups or
  - remote IP address provided in traffic selector payloads be used
- peer gateway location info e.g. IP address(es) or DNS names

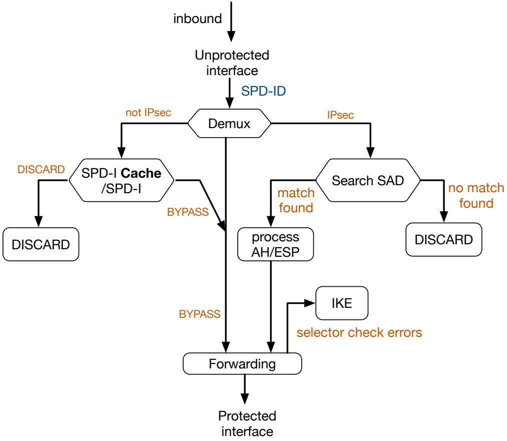
IPsec Outbound Traffic Processing

1. SPD selection function obtains the SPD-ID
2. Search SPD cache using SPD-ID and match packet header against found entries
3. matched vs no match
   - 3a. if matched: process the packet according to found policy
   - 3b. if no match: search SPD
     - match found: create cache entries, launch IKE if protect policy
     - no match: discard
4. packet is passed to outbound forwarding function (protected or bypassed)



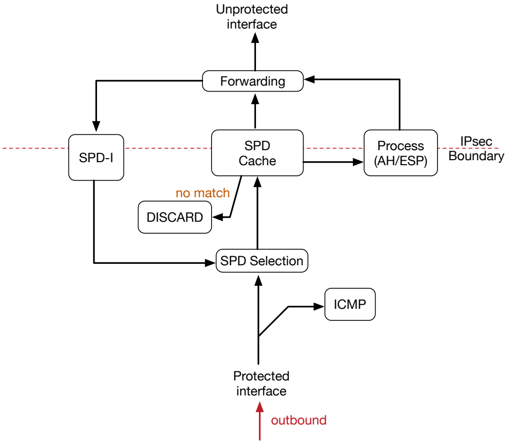
IP Inbound Traffic Processing

1. packet is mapped to an SPD-ID (tagged with interface ID)
2. demuxed into:
   - IPsec protected
   - not addressed to this device, or not IPsec
     - SPD-I: bypass or discard
3. if addressed to this device and IPsec
   - match found in SAD: process AH/ESP
   - match not found: discard
4. send IKE selectors error message if inconsistent selectors are received

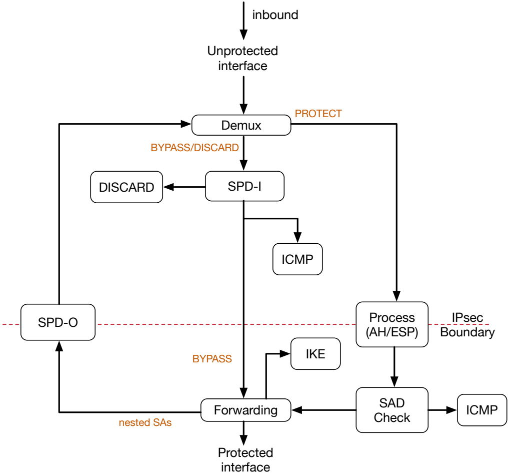


IPsec Traffic Processing Model

Outbound Traffic (RFC 4301)



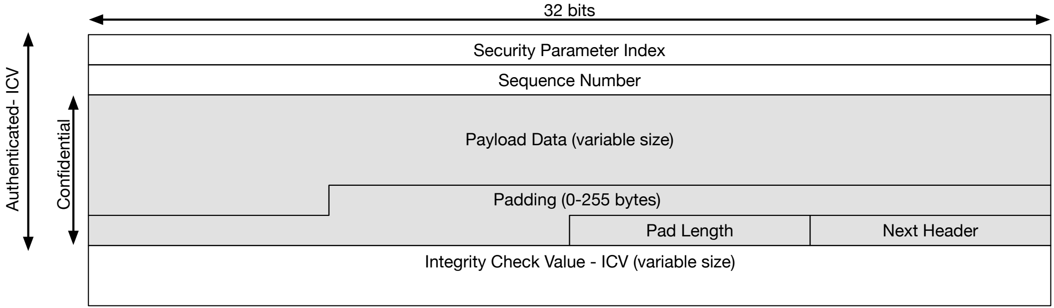
Inbound Traffic (RFC 4301)



Encapsulating Security Payload (ESP)

- Defined in RFC 4303
- provides
  - message content confidentiality,
  - data origin authentication,
  - connectionless integrity,
  - anti-replay service,
  - limited traffic flow confidentiality
- services depend on options selected during Security Association (SA)
- can use a variety of encryption and authentication algorithms (RFC 8221)
  - RFC 8221 specifies Encryption must be authenticated
  - three approaches are proposed
    - AEAD cipher (most efficient)
    - non-AEAD cipher + authentication (e.g. HMAC)
    - non-AEAD cipher + AH with authentication (not recommended, the slowest)

ESP Header, Payload, and Trailer



32 bits

Security Parameter Index

Sequence Number

Payload Data (variable size)

Padding (0-255 bytes)

Pad Length

Next Header

Integrity Check Value - ICV (variable size)

Authenticated- ICV

Confidential

Encryption and Authentication Algorithms and Padding

- ESP can encrypt payload data, padding, pad length, and next header fields
  - if needed have IV at start of payload data
- ESP can have optional ICV (Integrity Check Value) for integrity
  - is computed after encryption is performed
- ESP uses padding
  - to expand plaintext to required length
  - to align pad length and next header fields
  - to provide partial traffic flow confidentiality

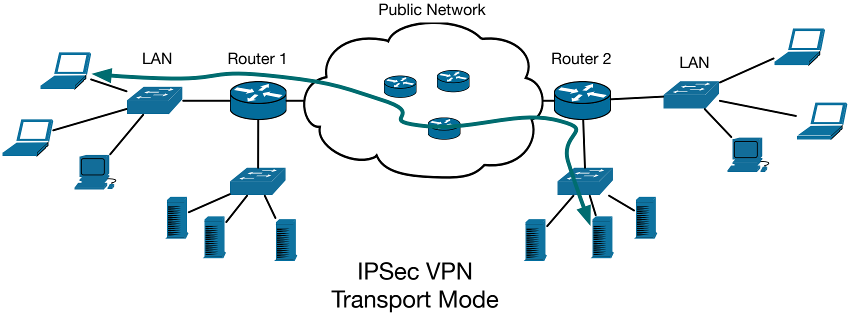
Anti-Replay Service

Replay is when attacker resends a copy of an authenticated packet

- use sequence number (32 bits) to thwart this attack
- Extended Sequence Number (ESN): 64-bit
  - can be negotiated between peers
  - is the default in IKEv2
- sender initialises sequence number to 0 when a new SA is established
  - increment for each packet
  - must not exceed the limit of 2^{32}-1 (2^{64}-1 ESN)
  - If this limit is reached, the sender terminates SA and renegotiates a new SA with a new key
    - Sequence Counter Overflow flag defines the behaviour (whether roll-over is permitted)
- receiver then accepts packets with sequence numbers within the window range of (N-W+1) to N
  - where W=window size; N = 2^{32} (2^{64} ESN)
  - Anti-Replay Window: limits how far out of order a packet can be, relative to the packet with the highest sequence number that has been authenticated so far

Transport Mode (AH/ESP)

- AH and ESP both can be configured in Transport or Tunnel mode
- to only encrypt IP payload but not IP header
- adversary can do traffic analysis (ESP)
    - source and destination IP addresses, frequency, limited size of original messages
- suitable in host to host VPN
    - since tunnel mode in this case does not provide any advantage

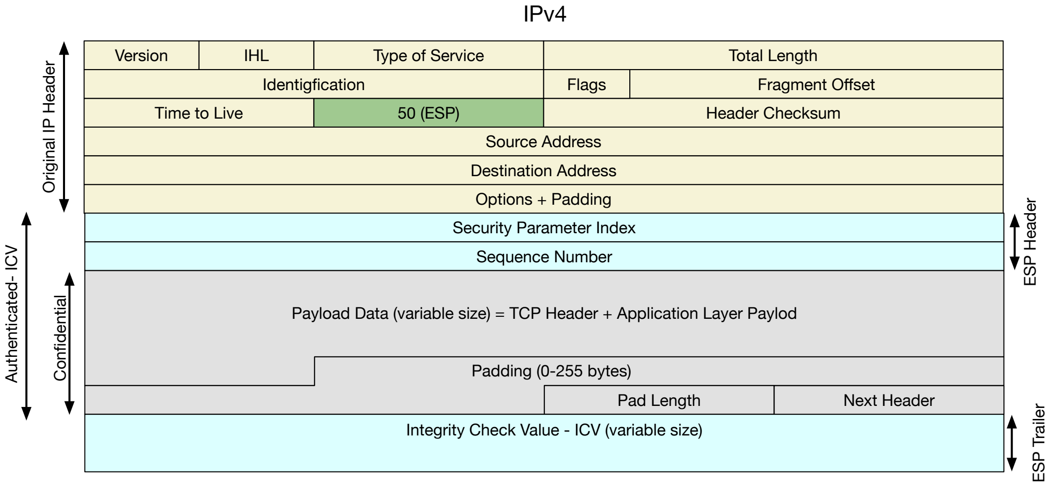


ESP Transport Mode: IPv4 Before ESP Encapsulation

IPv4

<table>
<tr><td colspan="1">Version</td><td colspan="1">IHL</td><td colspan="2">Type of Service</td><td colspan="4">Total Length</td></tr>
<tr><td colspan="4">Identigfication</td><td colspan="1">Flags</td><td colspan="3">Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td colspan="2">6 (TCP)</td><td colspan="4">Header Checksum</td></tr>
<tr><td colspan="8">Source Address</td></tr>
<tr><td colspan="8">Destination Address</td></tr>
<tr><td colspan="8">Options + Padding</td></tr>
<tr><td colspan="8">Payload = TCP Header + Application Layer Paylod</td></tr>
</table>

ESP Transport Mode Encapsulation



IPv4

Version | IHL | Type of Service | Total Length

Identigfication | Flags | Fragment Offset

Time to Live | 50 (ESP) | Header Checksum

Source Address

Destination Address

Options + Padding

Security Parameter Index

Sequence Number

Payload Data (variable size) = TCP Header + Application Layer Paylod

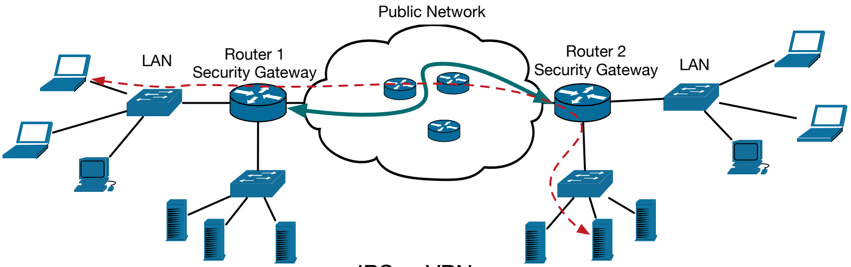
Padding (0-255 bytes)

Pad Length | Next Header

Integrity Check Value - ICV (variable size)

Tunnel Mode

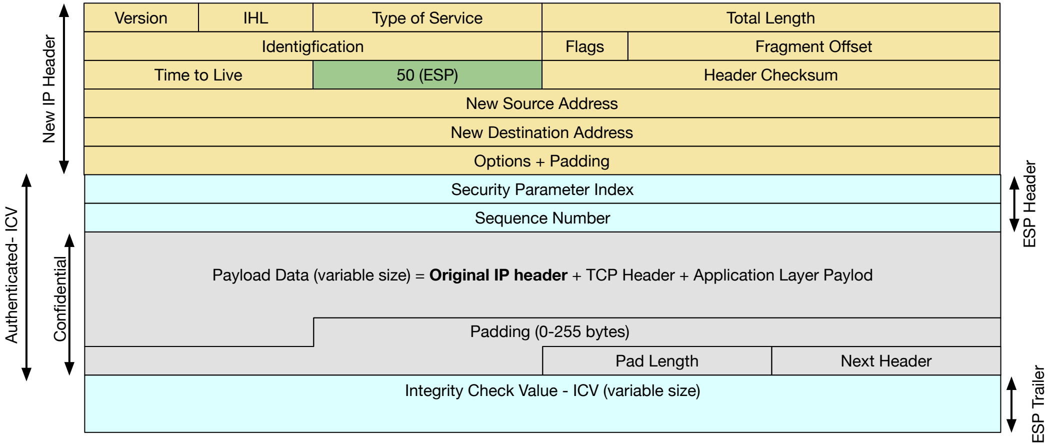
- encrypts entire original IP packet (ESP) and adds a new IP header
  - new IP header has the IP addresses of VPN end points as its source and destination addresses
- no router in the path can examine inner IP header
- suitable in gateway to gateway VPN (network to network)
- AH in this mode does not provide any advantage as the original IP header is still visible



IPSec VPN
Tunnel Mode

ESP Tunnel Mode Encapsulation

IPv4



New IP Header

Version | IHL | Type of Service | Total Length

Identigfication | Flags | Fragment Offset

Time to Live | 50 (ESP) | Header Checksum

New Source Address

New Destination Address

Options + Padding

ESP Header

Security Parameter Index

Sequence Number

Authenticated- ICV

Confidential

Payload Data (variable size) = **Original IP header** + TCP Header + Application Layer Paylod

Padding (0-255 bytes)

Pad Length | Next Header

ESP Trailer

Integrity Check Value - ICV (variable size)

Security association bundle

- Traffic flow between hosts may require IPsec but separate services maybe required between security gateways
- Security association bundle refers to a sequence of SAs through which traffic must be processed to provide desired set of IPsec services
    - combination of ESP and AH for the same traffic
- SA bundle methods
    - *Transport adjacency* (same end point pair)
        - Applying more than one security protocol (AH, ESP) to the same IP packet without invoking tunnelling
    - *Iterated tunnelling* (multiple pairs of end points)
        - Application of multiple layers of security protocols (AH, ESP) achieved through IP tunnelling
    - The two above approaches can be combined (not in the same end points)
        - Example: transport SA between hosts travel part of the way through a tunnel SA between security gateways

Authentication Plus Confidentiality

ESP with authentication option

- Option-1: Authenticated Encryption with Associated Data
  - most efficient method
- Option-2: ESP to protect data followed by AH on ciphertext which also protects IP header
  - Transport adjacency
- Option-3: Transport-Tunnel Bundle on multiple end points
  - Authentication prior to encryption in Transport mode between the actual end points
  - ESP in Tunnel mode between two security gateways

ESP and AH Cryptographic Suites: Authentication

- Authentication algorithms recommended by RFC 8221 (ESP and AH)

Name	Status	Comment
AUTH_NONE	MUST / MUST NOT	[RFC7296][RFC5282] AEAD-only
AUTH_HMAC_MD5_96	MUST NOT	[RFC2403][RFC7296]
AUTH_HMAC_SHA1_96	MUST-	[RFC2404][RFC7296]
AUTH_DES_MAC	MUST NOT	UNSPECIFIED
AUTH_KPDK_MD5	MUST NOT	UNSPECIFIED
AUTH_AES_XCBC_96	SHOULD / MAY	[RFC3566][RFC7296] (IoT)
AUTH_AES_128_GMAC	MAY	[RFC4543]
AUTH_AES_256_GMAC	MAY	[RFC4543]
AUTH_HMAC_SHA2_256_128	MUST	[RFC4868]
AUTH_HMAC_SHA2_512_256	SHOULD	[RFC4868]

ESP Cryptographic Suites: Encryption

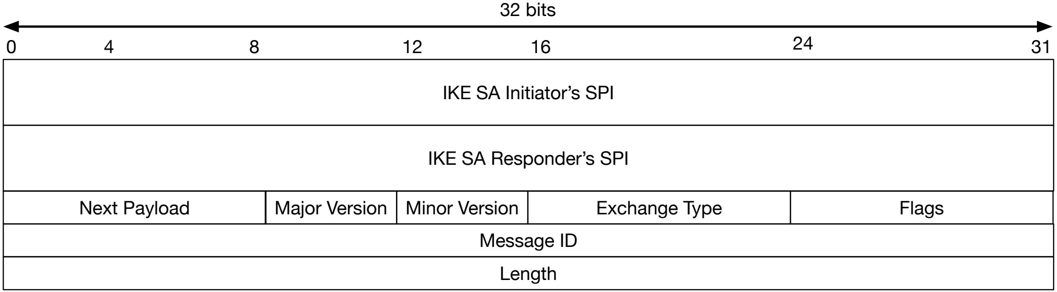
- Encryption algorithms recommended by RFC 8221 (ESP)

Name	Status	AEAD	Comment
ENCR_DES_IV64	MUST NOT	No	UNSPECIFIED
ENCR_DES	MUST NOT	No	[RFC2405]
ENCR_3DES	SHOULD NOT	No	[RFC2451]
ENCR_BLOWFISH	MUST NOT	No	[RFC2451]
ENCR_3IDEA	MUST NOT	No	UNSPECIFIED
ENCR_DES_IV32	MUST NOT	No	UNSPECIFIED
ENCR_NULL	MUST	No	[RFC2410]
ENCR_AES_CBC	MUST	No	[RFC3602][1]
ENCR_AES_CCM_8	SHOULD	Yes	[RFC4309](IoT)
ENCR_AES_GCM_16	MUST	Yes	[RFC4106][1]
ENCR_CHACHA20_POLY1305	SHOULD	Yes	[RFC7634]

Internet Key Exchange Protocol

- Version 1 (for historical references) was defined in RFC 2409 (IKEv1) and with relation to
  - RFC 2407 Domain Of Interpretation (DOI)
  - RFC 2408 Internet Security Association and Key Management Protocol (ISAKMP)
    - provides framework for key management
  - RFC 2412 OAKLEY Key Determination Protocol
    - defines DH key exchange
- Version 2 RFC 7296 replaces all previous documents (we will only study IKEv2)
- IKEv2 performs:
  - mutual authentication between two parties
  - establishes an IKE Security Association (SA) which includes
    - shared secret to establish ESP or AH SAs (called Child SAs)
    - set of cryptographic tools to be used by ESP or AH SAs
  - optionally can negotiate use of compression
- IKEv2 communication:
  - comprised of pair of messages: request and response
    - the pair is called exchange

IKE Header



32 bits

0 4 8 12 16 24 31

IKE SA Initiator's SPI

IKE SA Responder's SPI

Next Payload | Major Version | Minor Version | Exchange Type | Flags

Message ID

Length

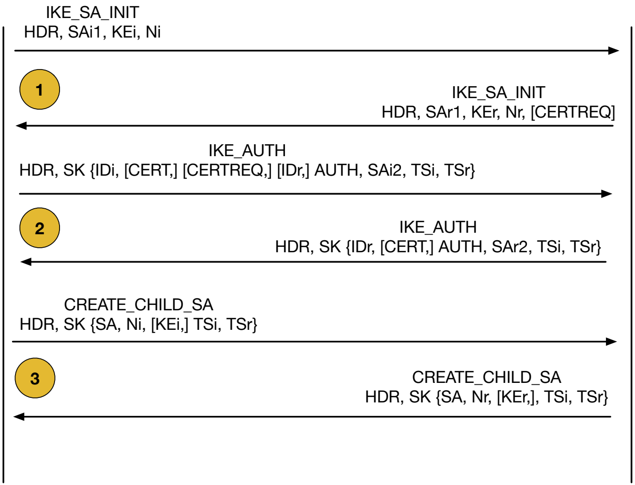
IKE Header Fields

IKE messages header fields:

- **Initiator's SPI (64 bits)**: chosen by the initiator to identify a unique IKE SA
- **Responder's SPI (64 bits)**: chosen by responder to identify unique IKE SA
- **Next Payload (8 bits)**: type of the first payload in the message (next slide).
- **Major/Minor Version (4 bits)**: Indicates major/minor version of IKE in use (v1, v2)
- **Exchange Type (8 bits)**: type of exchange.
- **Flags (8 bits)**: indicates specific options set for the message.
  - `XXRVIXXX`:
  - `X` bits are cleared when sending and ignored when received
  - `R`: Response, `I`: Initiator
  - `V`: Version, if set means transmitter is capable of speaking a higher major version number than specified in Major version field
- **Message ID (32 bits)**: control retransmission of lost packets, matching of requests/responses.
- **Length (32 bits)**: Total message (header plus all payloads) in octets

IKE Payloads and Overall Exchanges

- Security Association
- Key Exchange
- Identification
- Certificate
- Certificate Request
- Authentication
- Nonce
- Notify
- Delete
- Vendor ID
- Traffic Selector
- Encrypted
- Configuration
- Extensible Authentication Protocol



IKE_SA_INIT
HDR, SAi1, KEi, Ni

IKE_SA_INIT
HDR, SAr1, KEr, Nr, [CERTREQ]

IKE_AUTH
HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr}

IKE_AUTH
HDR, SK {IDr, [CERT,] AUTH, SAr2, TSi, TSr}

CREATE_CHILD_SA
HDR, SK {SA, Ni, [KEi,] TSi, TSr}

CREATE_CHILD_SA
HDR, SK {SA, Nr, [KEr,], TSi, TSr}

IKEv2 IKE_SA_INIT Exchange

- two parties: Initiator(i) and Responder(r)
- first message exchange:
    - IKE_SA_INIT: to establish an IKE SA
        - must be completed before any other exchange
        - negotiates security parameters for IKE SA
        - exchange nonces (Ni and Nr) and Diffie-Hellman values (g^i and g^r)
        - each party then derives keys from the nonces exchanged and DH shared secret
- initiator message: HDR, SAi1, KEi, Ni
    - HDR: contains SPI, version numbers, Exchange Type, Message ID, and flags
    - SAi1: states the cryptographic algorithms the initiator supports (for IKE SA)
    - KEi: initiator's DH public key (guesses which DH algorithm and group will be chosen)
    - Ni: initiators nonce
- responder message: HDR, SAr1, KEr, Nr, [CERTREQ]
    - HDR: similar to initiator message
    - SAr1: chosen cryptographic suite from the list of supported algorithms offered by initiator
    - [CERTREQ]: responder can optionally request certificate from initiator
- responder will send INVALID_KE_PAYLOAD if DH group was guessed incorrectly and

IKEv2 Key Derivation after IKE_SA_INIT Exchange

- both sides derive

```
SKEYSEED = prf(Ni || Nr, g^ir)
```

- seven keys are derived for this IKE SA

```
Key Material = prf+ (SKEYSEED, Ni || Nr || SPIi || SPIr)
```

- Key Material
  - SK_d: for deriving new keys for Child SAs established with this IKE_SA_INIT
  - SK_ai and SK_ar: used as keys to the integrity protection algorithm for the subsequent messages
    - integrity keys are different in each direction
  - SK_ei and SKer: for encrypting and decrypting all subsequent exchanges
    - encryption keys are different in each direction
  - SK_pi and SK_pr: are used when generating AUTH payload
- prf is a pseudo-random function (HMAC)
- prf and prf+ are used as key derivation functions (see Appendix)

IKEv2 IKE_AUTH Exchange

- IKE_AUTH: to authenticate the peer (mutual)
  - must be completed after IKE_SA_INIT and before any other exchanges
  - transmits identities
  - proves knowledge of secrets corresponding to identities
  - authenticates previous messages
  - sets up an SA for ESP or AH Child SA
  - parts of these messages are encrypted and integrity protected
  - also known as Phase 1 of IKEv1
- initiator: HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr}
  - the notation SK {} means the enclosed values are encrypted with SK_ei and SK_er and integrity protected using SK_ai and SK_ar
  - IDi: initiators identity
  - [CERT,] [CERTREQ,]: optionally sends its certificate and requests responder's certificate
  - [IDr,]: optionally chooses which of the responder's identities it wants to talk to
  - AUTH: proves knowledge of secret information
  - SAi2: begins negotiation of a Child SA using this payload

IKEv2 CREATE_CHILD_SA Exchange

- CREATE_CHILD_SA exchange: creates a Child SA
  - this exchange rekeys both IKE SAs and Child SAs
  - may be initiated by either peers
  - an SA is rekeyed by creating a new SA and deleting the old one
- initiator: `HDR, SK {SA, Ni, [KEi,] TSi, TSr}`
  - `SA`: SA offer(s)
  - `Ni`: a (new) nonce
  - `[KEi,]`: optionally a (new) DH public key
  - `TSi, TSr`: proposed traffic selectors for proposed Child SA
- responder: `HDR, SK {SA, Nr, [KEr,], TSi, TSr}`
- these exchanges are protected using cryptographic algorithms and keys negotiated by `IKE_SA_INIT`

IKEv2 CREATE_CHILD_SA Exchange to Rekey IKE or Child SAs

- to rekey IKE SA
  - initiator: `HDR, SK {SA, Ni, KEi}`
    - `SA`: SA offer(s), will include a new SPI in SPI field of SA offer
    - `Ni`: a nonce
    - `KEi`: DH public key
  - responder: `HDR, SK {SA, Nr, KEr}`
    - `SA`: accepted SA offer(s), will include a new SPI in SPI field
    - `Nr`: a nonce
    - `KEr`: DH public key
  - message counter for the new IKE SA will be set to 0
- to rekey Child SA
  - initiator: `HDR, SK {N(REEKEY_SA), SA, Ni, [KEi,], TSi, TSr}`
    - `N(REEKEY_SA)`: Child SA rekey notification
    - `SA`: SA offer(s), will include a new SPI in SPI field of SA offer
    - `Ni`: a nonce
    - `[KEi,]`: optionally a DH public key
    - `TSi, TSr`: proposed traffic selectors for proposed Child SA

IKEv2 INFORMATIONAL Exchange

- INFORMATIONAL Exchange
  - to delete SAs
  - report errors and conditions
  - an empty INFORMATIONAL exchange could be used to check liveness
  - these exchanges can occur in any order (after the first two)
- initiator: `HDR, SK {[N,] [D,] [CP,] ...}`
- responder: `HDR, SK {[N,] [D,] [CP,] ...}`
- INFORMATIONAL messages outside IKE SA
  - an ESP or AH packet with an unrecognised SPI
  - an encrypted IKE request on port 500 or 4500 with an unrecognised IKE SPI
  - an IKE request packet with a higher major version number than supported (by this node)

IKEv2 Cryptographic Suites: Encryption Methods

- Supported Encryption Methods (RFC 8247)

Name	Status	AEAD	Comment
ENCR_DES_IV64	MUST NOT	No	UNSPECIFIED
ENCR_DES	MUST NOT	No	[RFC2405]
ENCR_3DES	SHOULD NOT	No	[RFC2451]
ENCR_BLOWFISH	MUST NOT	No	[RFC2451]
ENCR_3IDEA	MUST NOT	No	UNSPECIFIED
ENCR_DES_IV32	MUST NOT	No	UNSPECIFIED
ENCR_NULL	MUST	No	[RFC2410]
ENCR_AES_CBC	MUST	No	[RFC3602][1]
ENCR_AES_CCM_8	SHOULD	Yes	[RFC4309](IoT)
ENCR_AES_GCM_16	MUST	Yes	[RFC4106][1]
ENCR_CHACHA20_POLY1305	SHOULD	Yes	[RFC7634]

IKEv2 Cryptographic Suites: Symmetric Authentication Methods

- Supported Symmetric Authentication Methods (RFC 8247)

Name	Status	Comment
AUTH_NONE	MUST / MUST NOT	[RFC7296][RFC5282] AEAD-only
AUTH_HMAC_MD5_96	MUST NOT	[RFC2403][RFC7296]
AUTH_HMAC_SHA1_96	MUST-	[RFC2404][RFC7296]
AUTH_DES_MAC	MUST NOT	UNSPECIFIED
AUTH_KPDK_MD5	MUST NOT	UNSPECIFIED
AUTH_AES_XCBC_96	SHOULD / MAY	[RFC3566][RFC7296] (IoT)
AUTH_AES_128_GMAC	MAY	[RFC4543]
AUTH_AES_256_GMAC	MAY	[RFC4543]
AUTH_HMAC_SHA2_256_128	MUST	[RFC4868]
AUTH_HMAC_SHA2_512_256	SHOULD	[RFC4868]

IKEv2 Key Exchange Methods

- Key Exchange methods recommended by RFC 8247

Number	Description	Status
14	2048-bit MODP Group	MUST
19	256-bit random ECP group	SHOULD
5	1536-bit MODP Group	SHOULD NOT
2	1024-bit MODP Group	SHOULD NOT
1	768-bit MODP Group	MUST NOT
22	1024-bit MODP Group with 160-bit Prime Order Subgroup	MUST NOT
23	2048-bit MODP Group with 224-bit Prime Order Subgroup	SHOULD NOT
24	2048-bit MODP Group with 256-bit Prime Order Subgroup	SHOULD NOT

DH 3072-bit MODP Group Specified in RFC 3526

```
4.  3072-bit MODP Group

   This group is assigned id 15.

   This prime is: 2^3072 - 2^3008 - 1 + 2^64 * { [2^2942 pi] + 1690314 }

   Its hexadecimal value is:

      FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
      29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
      EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
      E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
      EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
      C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
      83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
      670C354E 4ABC9804 F1746C08 CA18217C 32905E46 2E36CE3B
      E39E772C 180E8603 9B2783A2 EC07A28F B5C55DF0 6F4C52C9
      DE2BCBF6 95581718 3995497C EA956AE5 15D22618 98FA0510
      15728E5A 8AAAC42D AD33170D 04507A33 A85521AB DF1CBA64
      ECFB8504 58DBEF0A 8AEA7157 5D060C7D B3970F85 A6E1E4C7
      ABF5AE8C DB0933D7 1E8C94E0 4A25619D CEE3D226 1AD2EE6B
      F12FFA06 D98A0864 D8760273 3EC86A64 521F2B18 177B200C
      BBE11757 7A615D6C 770988C0 BAD946E2 08E24FA0 74E5AB31
      43DB5BFC E0FD108E 4B82D120 A93AD2CA FFFFFFFF FFFFFFFF

   The generator is: 2.
```

IKEv2 Authentication Methods

- Authentication methods recommended by RFC 8247

Number	Description	Status
1	RSA Digital Signature	MUST
2	Shared Key Message Integrity Code	MUST
3	DSS Digital Signature	SHOULD NOT
9	ECDSA with SHA-256 on the P-256 curve	SHOULD
10	ECDSA with SHA-384 on the P-384 curve	SHOULD
11	ECDSA with SHA-512 on the P-521 curve	SHOULD
14	Digital Signature	SHOULD

References

Materials are reproduced, at times without any modification, from the following sources:

- RFC 4301: Security Architecture for Internet Protocol
- RFC 4302: IP Authentication Header (AH)
- RFC 4303: IP Encapsulating Security Payload (ESP)
- RFC 7296: Internet Key Exchange (IKEv2) Protocol
- RFC 8221: Cryptographic Algorithm Implementation Requirements and Usage Guidance for ESP and AH
- RFC 7296: Internet Key Exchange Protocol Version 2
- RFC 3526: More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)

Appendix

```
prf+ (K,S) = T1 || T2 || T3 || || T4 ...
```

where:

```
T1 = prf (K, S || 0x01)
T2 = prf (K, T1 || 0x02)
T3 = prf (K, T2 || 0x03)
T4 = prf (K, T3 || 0x04)
...
```

- Recommended PRFs in RFC 8247: Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)

Name	Status	Comment
PRF_HMAC_SHA2_256	MUST	
PRF_HMAC_SHA2_512	SHOULD+	
PRF_HMAC_SHA1	MUST-	
PRF_AES128_XCBC	SHOULD	(IoT)
PRF_HMAC_MD5	MUST NOT	