



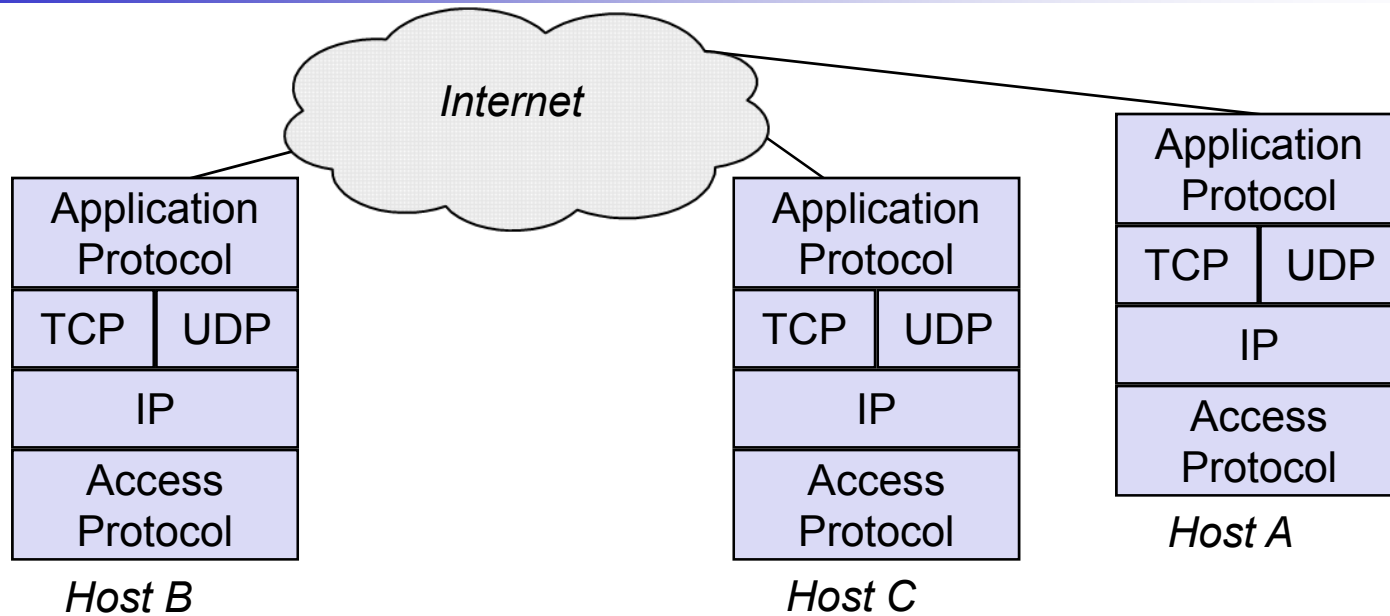
---

# Chapter 11

## The *IPSec* Security Architecture for the Internet Protocol

- ❑ IPSec Architecture
- ❑ Security Associations
- ❑ AH / ESP
- ❑ IKE

# The TCP/IP Protocol Suite



- ❑ *IP (Internet Protocol)*: unreliable, connectionless network protocol
- ❑ *TCP (Transmission Control Protocol)*: reliable, connection-oriented transport protocol, realized over IP
- ❑ *UDP (User Datagram Protocol)*: unreliable, connectionless transport protocol, offers an application interface to IP
- ❑ Examples for *application protocols*:
  - HTTP: Hypertext Transfer Protocol
  - SMTP: Simple Mail Transfer Protocol

# The IP Packet Format



Ver.	IHL	TOS	Length	
IP Identification			Flags	Fragment Offset
TTL		Protocol	IP Checksum	
Source Address				
Destination Address				
IP Options (if any)				
TCP / UDP / ... Payload				

- ❑ **Version (Ver.):** 4 bit
  - ❑ Currently version 4 is widely deployed
  - ❑ Version 6 is in deployed sparsely
- ❑ **Internet header length (IHL):** 4 bit
  - ❑ Length of the IP header in 32-bit words
- ❑ **Type of service (TOS):** 8 bit
  - ❑ This field is used to indicate the traffic requirements of a packet
  - ❑ It is currently under review at the IETF
- ❑ **Length:** 16 bit
  - ❑ The length of the packet including the header in octets
  - ❑ This field is, like all other fields in the IP suite, in “big endian” representation
- ❑ **Identification:** 16 bit
  - ❑ Used to “uniquely” identify an IP datagram
  - ❑ Important for re-assembly of fragmented IP datagrams
- ❑ **Flags:** 3 bit
  - ❑ Fragmentation
- ❑ **Fragmentation offset:** 13 bit
  - ❑ The position of this packet in the corresponding IP datagram
- ❑ **Time to live (TTL):** 8 bit
  - ❑ At every processing network node, this field is decremented by one
  - ❑ When TTL reaches 0 the packet is discarded to avoid packet looping
- ❑ **Protocol:** 8 bit
  - ❑ Indicates the (transport) protocol of the payload
  - ❑ Used by the receiving end system to de-multiplex packets among various transport protocols like TCP, UDP, ...
- ❑ **Checksum:** 16 bit
  - ❑ Protection against transmission errors
  - ❑ As this is not a cryptographic checksum, it can easily be forged
- ❑ **Source address:** 32 bit
  - ❑ The IP address of sender of this packet
- ❑ **Destination address:** 32 bit
  - ❑ The IP address of the intended receiver of this packet
- ❑ **IP Options:** variable length
  - ❑ An IP header can optionally carry additional information

# Security Problems of the Internet Protocol

---



- ❑ When an entity receives an IP packet, it has no assurance of:
  - ❑ *Data origin authentication / data integrity:*
    - The packet has actually been send by the entity which is referenced by the source address of the packet
    - The packet contains the original content the sender placed into it, so that it has not been modified during transport
    - The receiving entity is in fact the entity to which the sender wanted to send the packet
  - ❑ *Confidentiality:*
    - The original data was not inspected by a third party while the packet was sent from the sender to the receiver

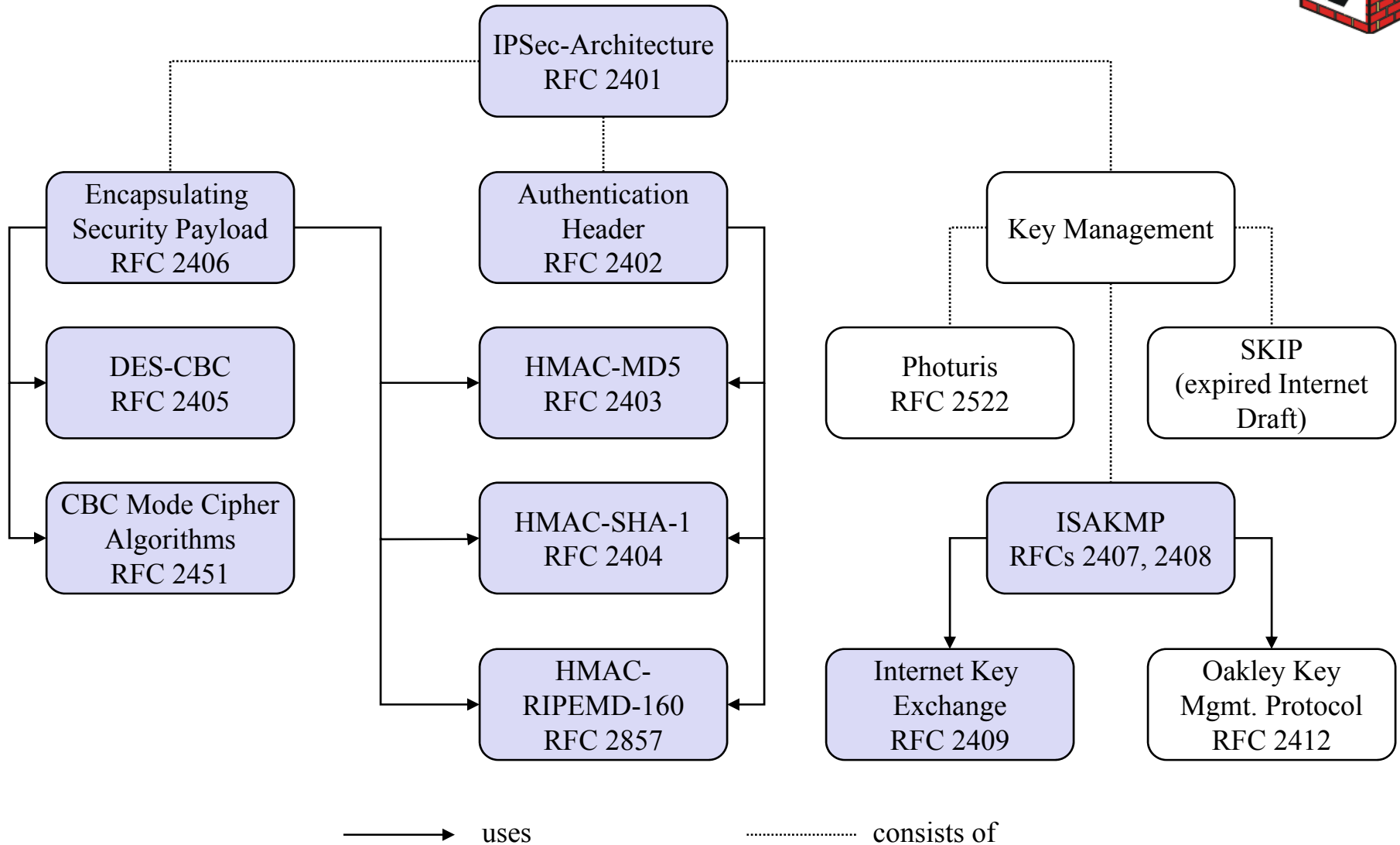
# Security Objectives of IPSec

---



- ❑ IPSec aims to ensure the following security objectives:
  - ❑ *Data origin authentication / connectionless data integrity:*
    - It is not possible to send an IP datagram with neither a masqueraded IP source nor destination address without the receiver being able to detect this
    - It is not possible to modify an IP datagram in transit without the receiver being able to detect the modification
  - ❑ *Replay protection:* it is not possible to later replay a recorded IP packet without the receiver being able to detect this
  - ❑ *Confidentiality:* it is not possible to eavesdrop on the content of IP datagrams (and limited traffic flow confidentiality)
- ❑ Security policy:
  - ❑ Sender, receiver and intermediate nodes can determine the required protection for an IP packet according to a local security policy
  - ❑ Intermediate nodes and the receiver will drop IP packets that do not meet these requirements

# Overview of the IPSec Standardization



# Overview of the IPSec Architecture



- ❑ RFC 2401 defines the basic architecture of IPSec:
  - ❑ Concepts:
    - Security association (SA), security association database (SADB)
    - Security policy, security policy database (SPD)
  - ❑ Fundamental IPSec Protocols:
    - Authentication Header (AH)
    - Encapsulating Security Payload (ESP)
  - ❑ Protocol Modes:
    - Transport Mode
    - Tunnel Mode
  - ❑ Use of various cryptographic primitives with AH and ESP:
    - Encryption: DES-CBC, CBC mode cipher algorithms
    - Integrity: HMAC-MD5, HMAC-SHA-1, HMAC-RIPEND-160
  - ❑ Key Management Procedures:
    - ISAKMP, IKE

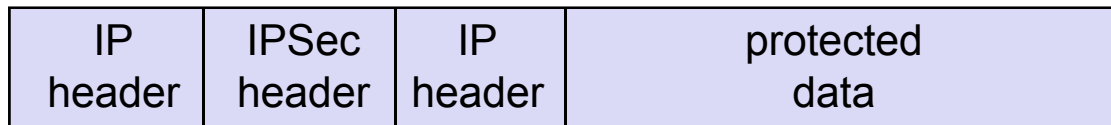
# IPSec Protocol Modes



- ❑ **Transport mode** can only be used between end-points of a communication
- ❑ **Tunnel mode** can be used with arbitrary peers
- ❑ The difference between the two modes is, that:
  - ❑ Transport mode just adds a security specific header (+ trailer):



- ❑ Tunnel mode encapsulates IP packets:

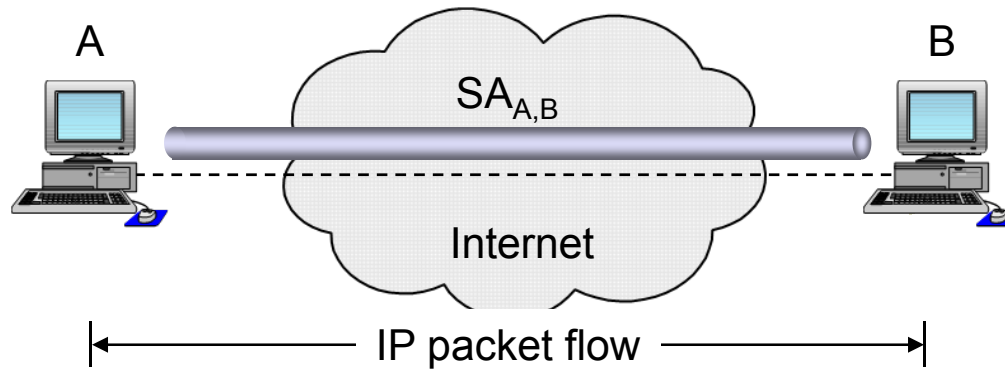


Encapsulation of IP packets allows for a gateway protecting traffic on behalf of other entities (e.g. hosts of a subnetwork, etc.)

# When to use which IPSec Mode?



- ❑ Transport mode is used when the “cryptographic endpoints” are also the “communication endpoints” of the secured IP packets
  - ❑ Cryptographic endpoints: the entities that generate or process an IPSec header
  - ❑ Communication endpoints: source and destination of an IP packet

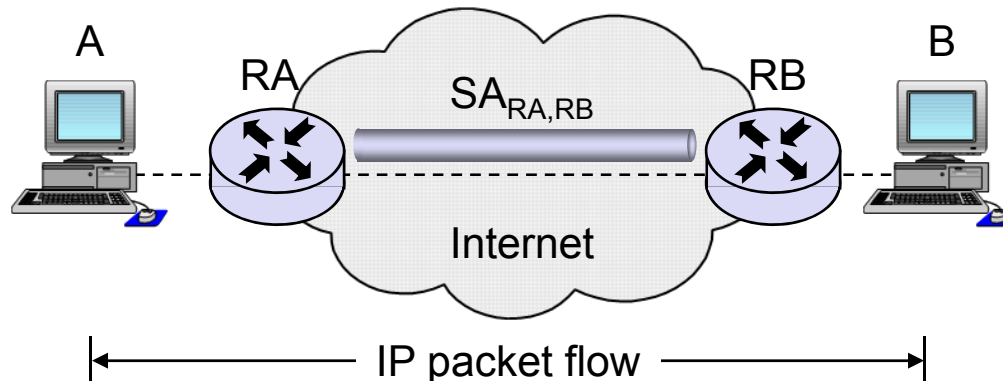


- ❑ In most cases, communication endpoints are hosts (workstations, servers), but this is not necessarily the case:
  - ❑ Example: a gateway being managed via SNMP by a workstation

# When to use which IPSec Mode?



- ❑ Tunnel mode is used when at least one “cryptographic endpoint” is not a “communication endpoint” of the secured IP packets
  - ❑ This allows for gateways securing IP traffic on behalf of other entities



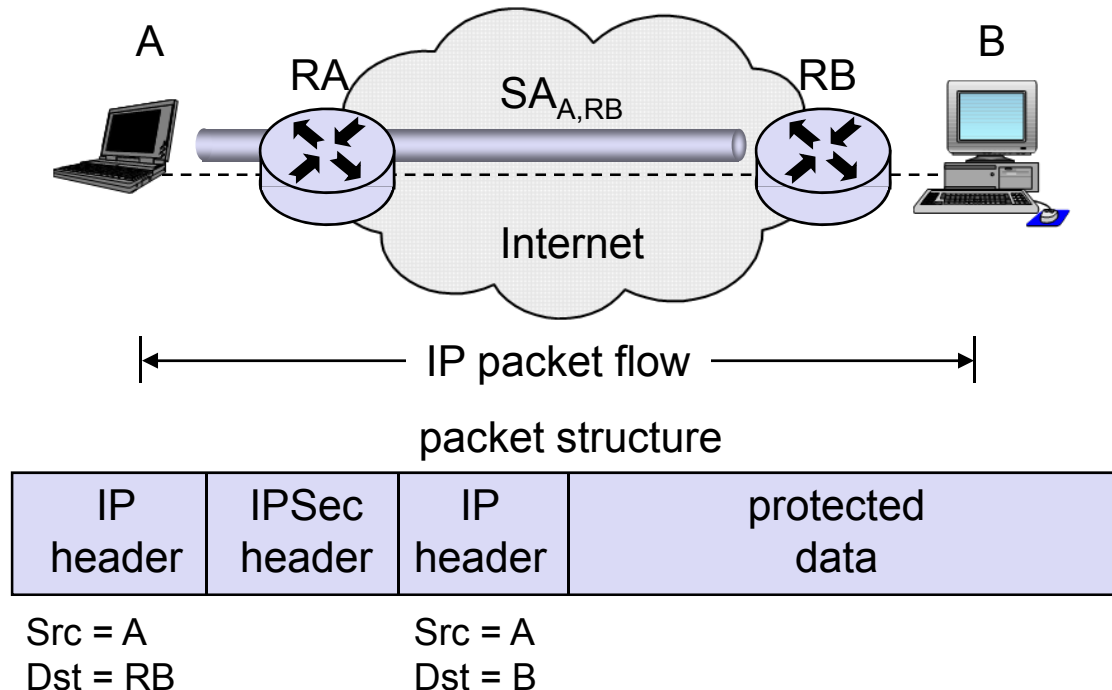
packet structure

IP header	IPSec header	IP header	protected data
Src = RA Dst = RB		Src = A Dst = B	

# When to use which IPSec Mode?



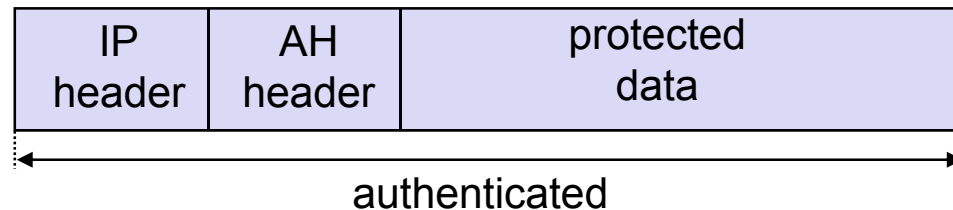
- The above description of application scenarios for tunnel mode includes the case in which only one cryptographic endpoint is not a communication endpoint:
  - Example: a security gateway ensuring authentication and / or confidentiality of IP traffic between a local subnetwork and a host connected via the Internet (“road warrior scenario”)



# Authentication Header (AH) Protocol

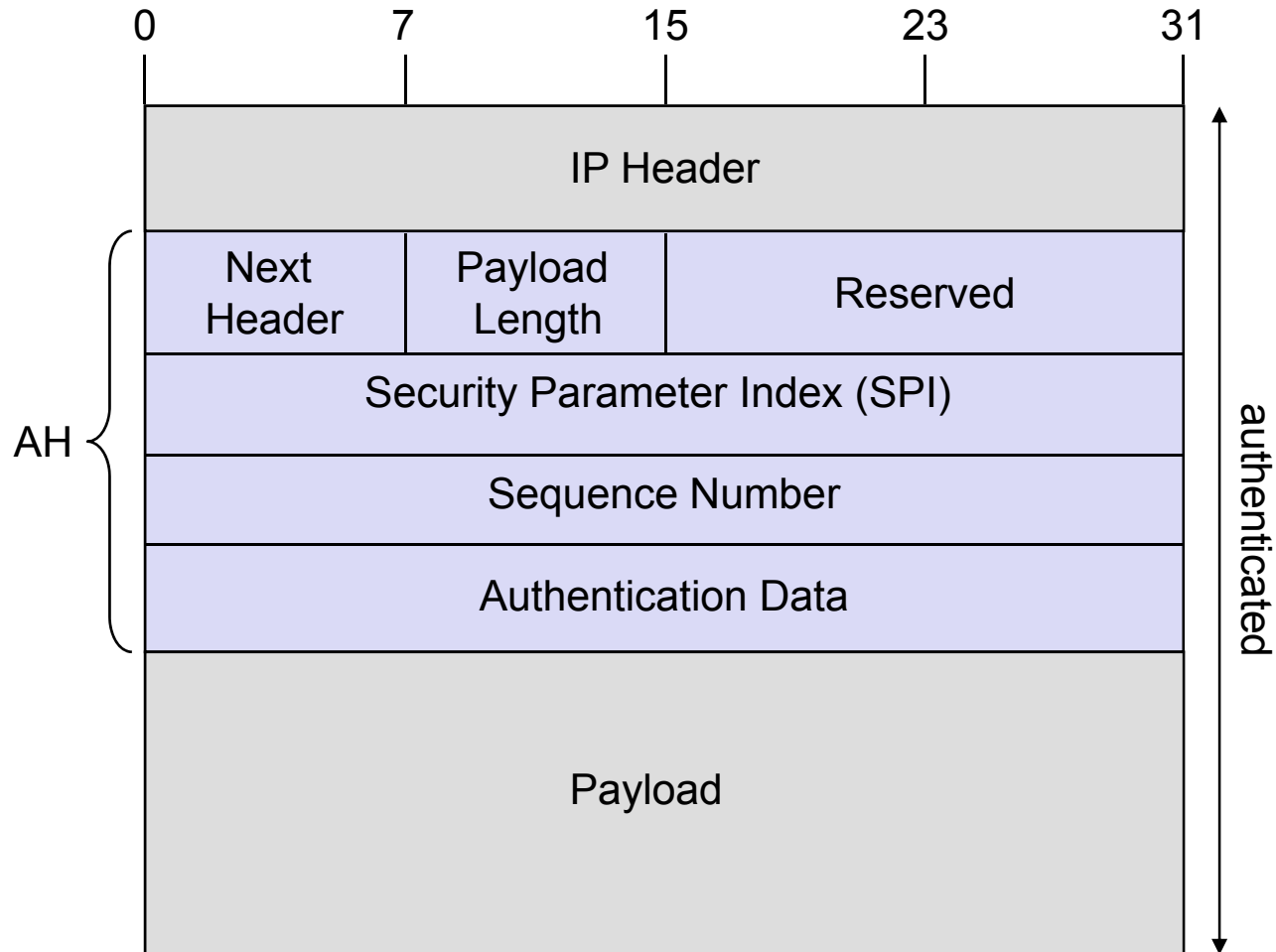


- ❑ Provides data origin authentication and replay protection
- ❑ Is realized as a header which is inserted between the IP header and the data to be protected



- ❑ AH constitutes a generic security protocol that provides to IP packets:
  - ❑ Data origin authentication, by creating and adding MACs to packets
- ❑ The AH definition is divided up into two parts:
  - ❑ The definition of the base protocol [RFC2402]: definition of the header format, basic protocol processing, tunnel and transport mode operation
  - ❑ The use of specific cryptographic algorithms with AH (for authentication):
    - HMAC-MD5-96 [RFC2403], HMAC-SHA-96 [RFC2404]

# Authentication Header (AH) Protocol

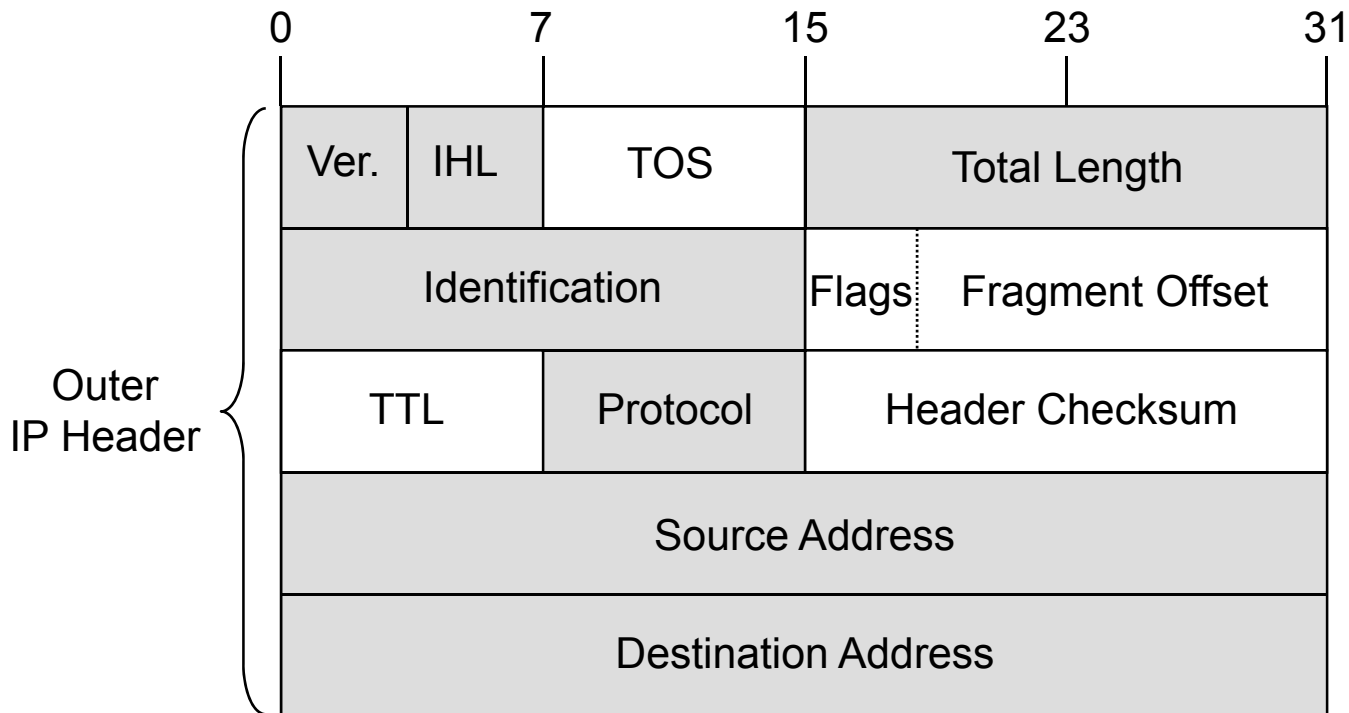


- ❑ In tunnel mode the payload constitutes a complete IP packet

# Authentication Header (AH) Protocol

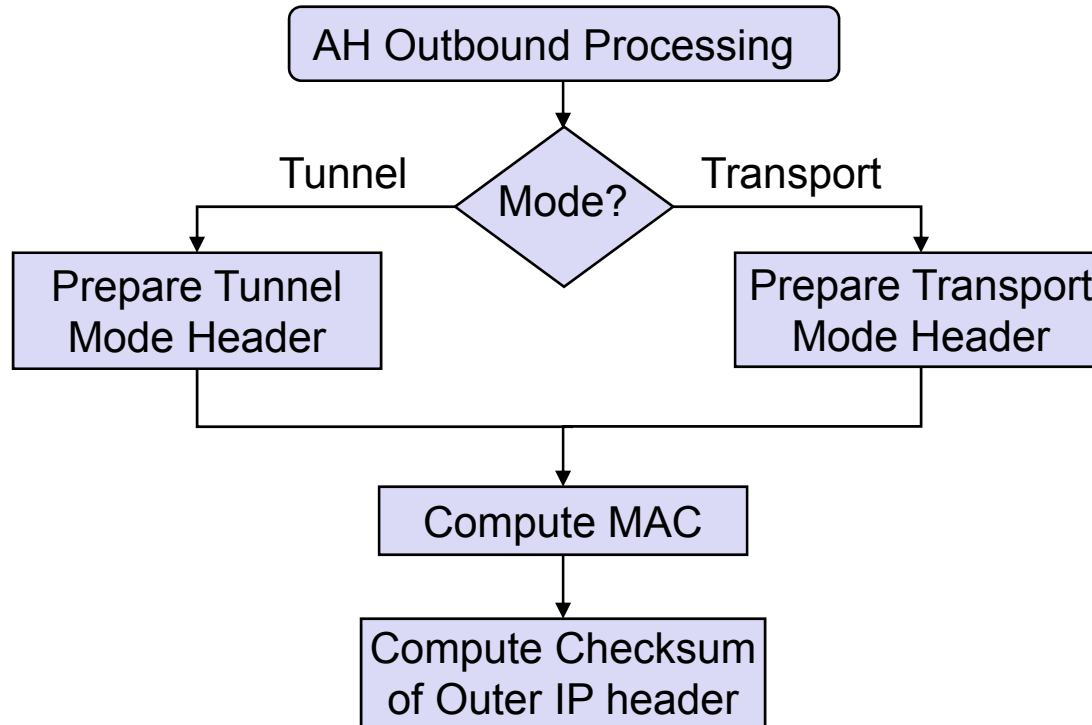


- Although AH also protects the outer IP header, some of its' fields must not be protected as they are subject to change during transit:
  - This also applies to mutable IPv4 options or IPv6 extensions
  - Such fields are assumed being zero when computing the MAC

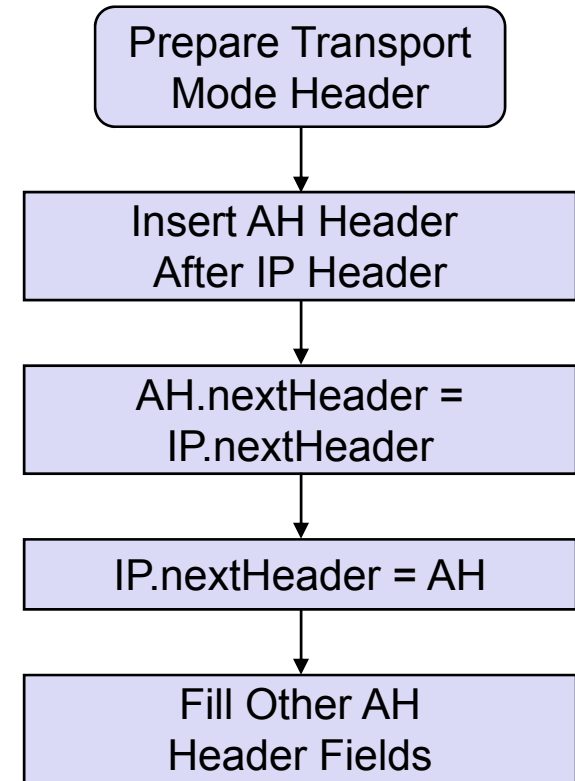
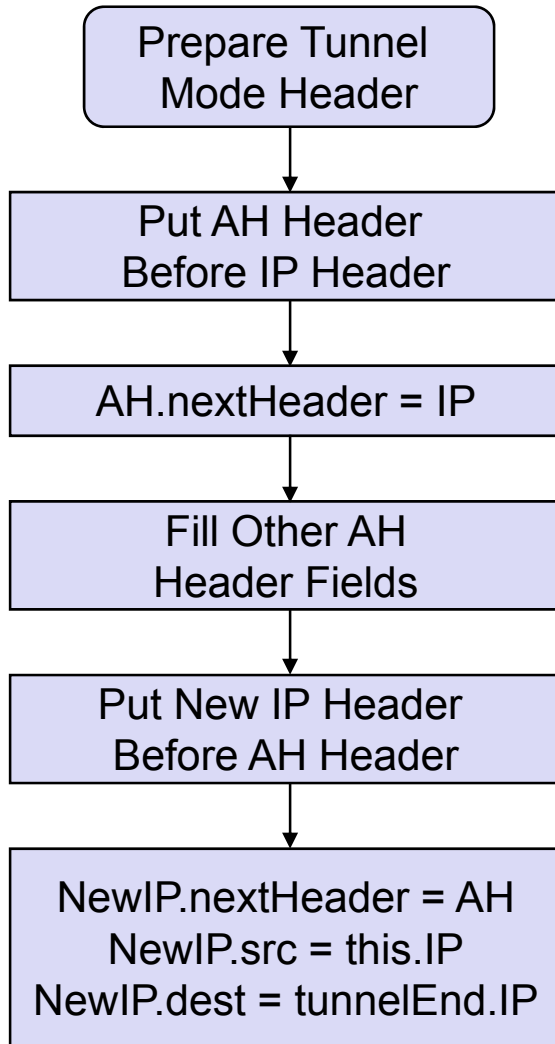


- All immutable fields, options and extensions (gray) are protected

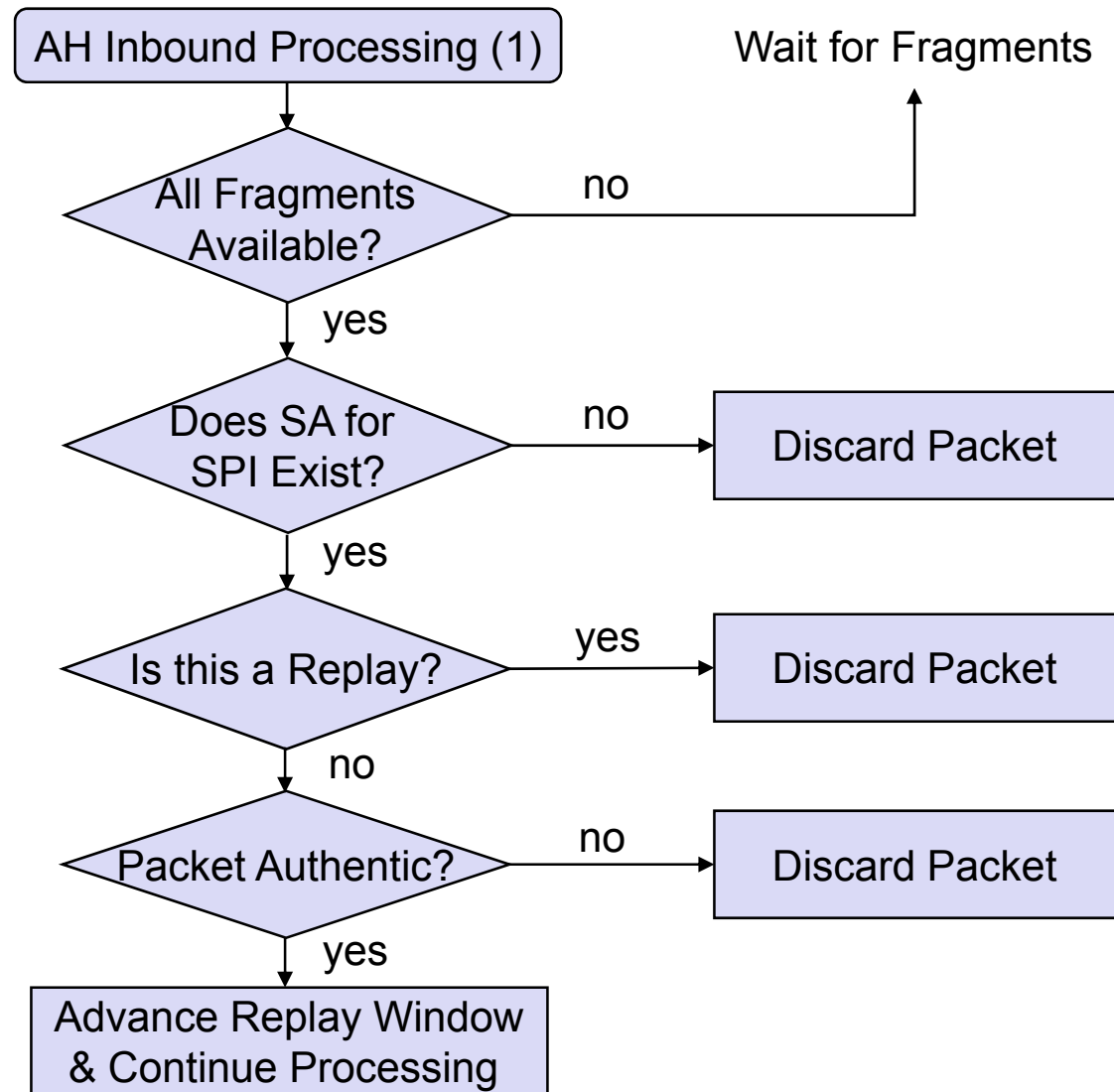
# Authentication Header (AH) Protocol



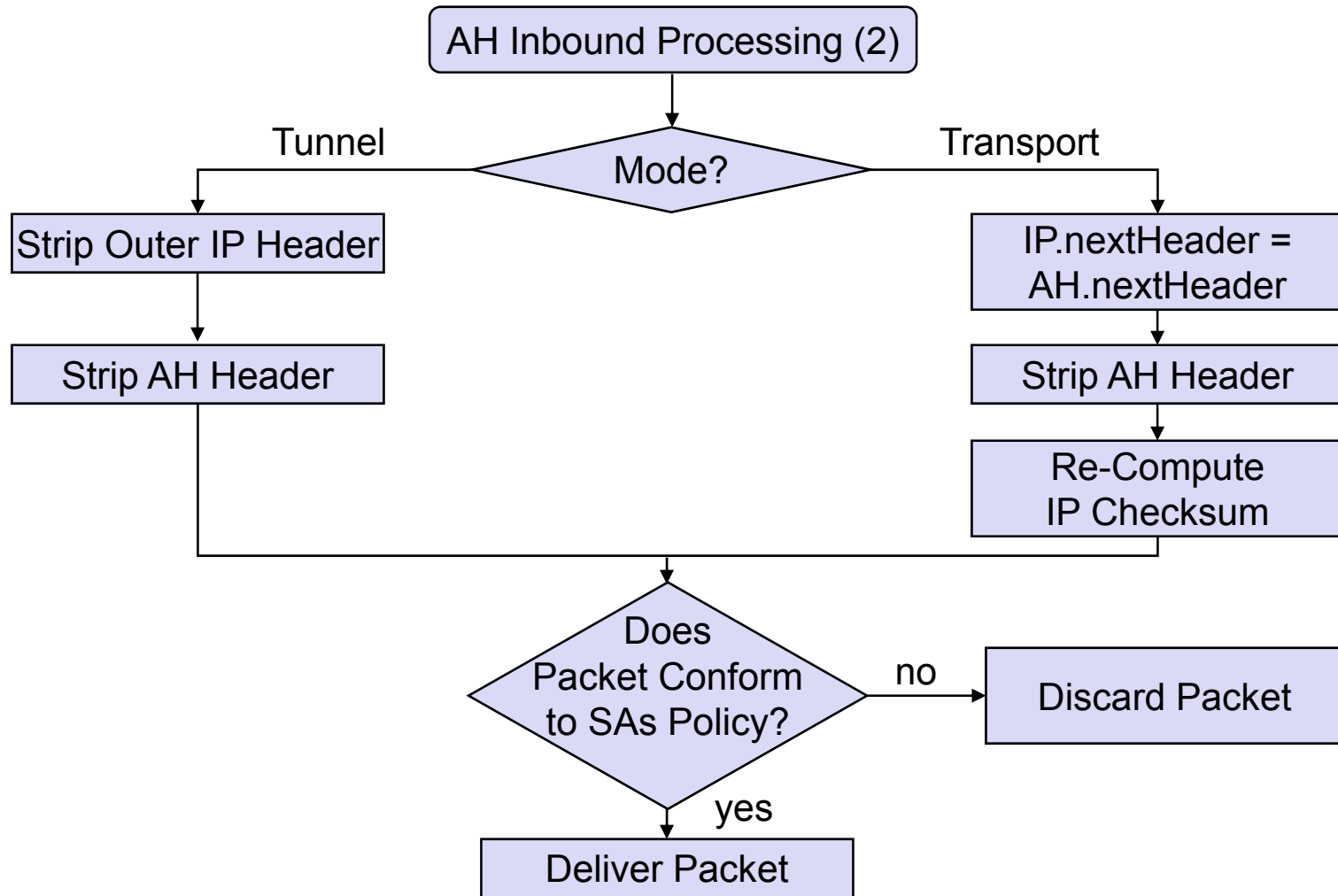
# Authentication Header (AH) Protocol



# Authentication Header (AH) Protocol



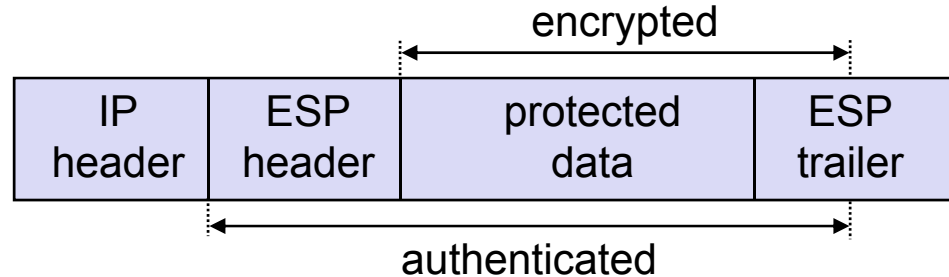
# Authentication Header (AH) Protocol



# Encapsulating Security Payload (ESP) Protocol

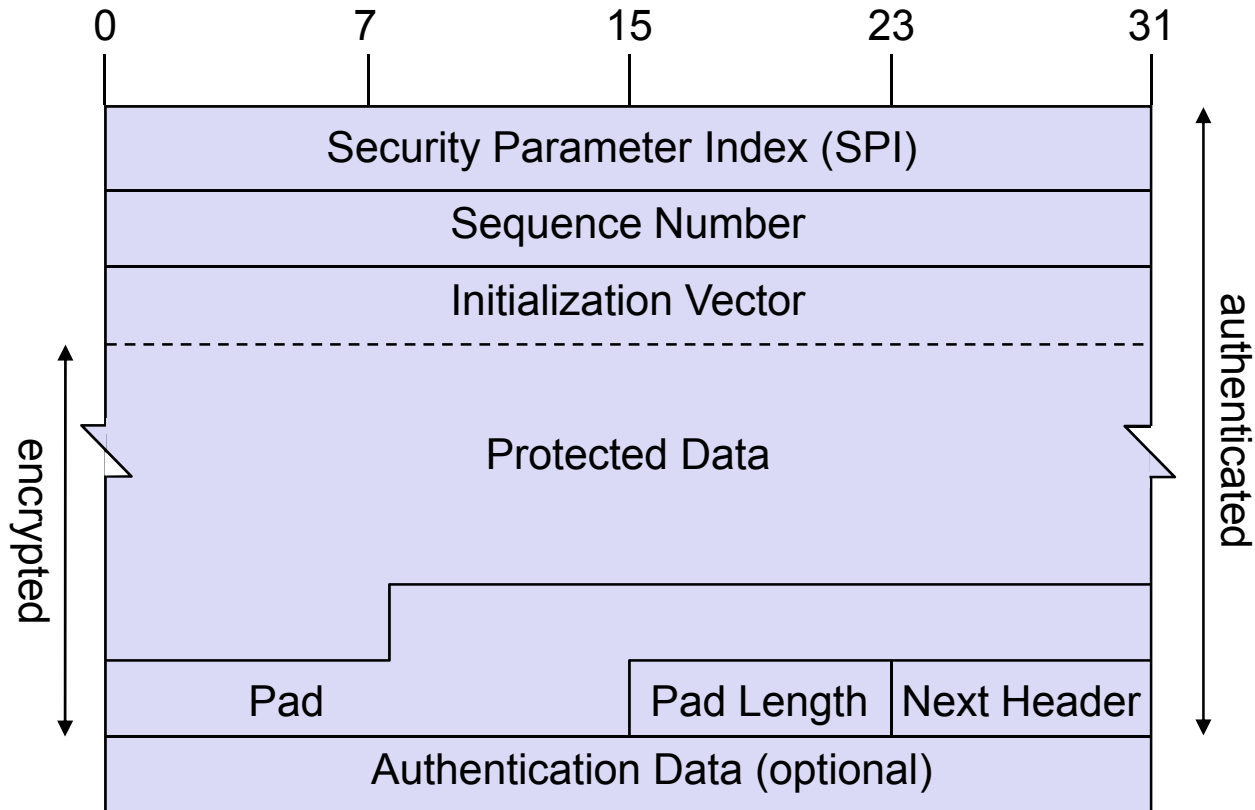


- ❑ Provides data origin authentication, confidentiality and replay protection
- ❑ Is realized with a header and a trailer encapsulating the data to be protected



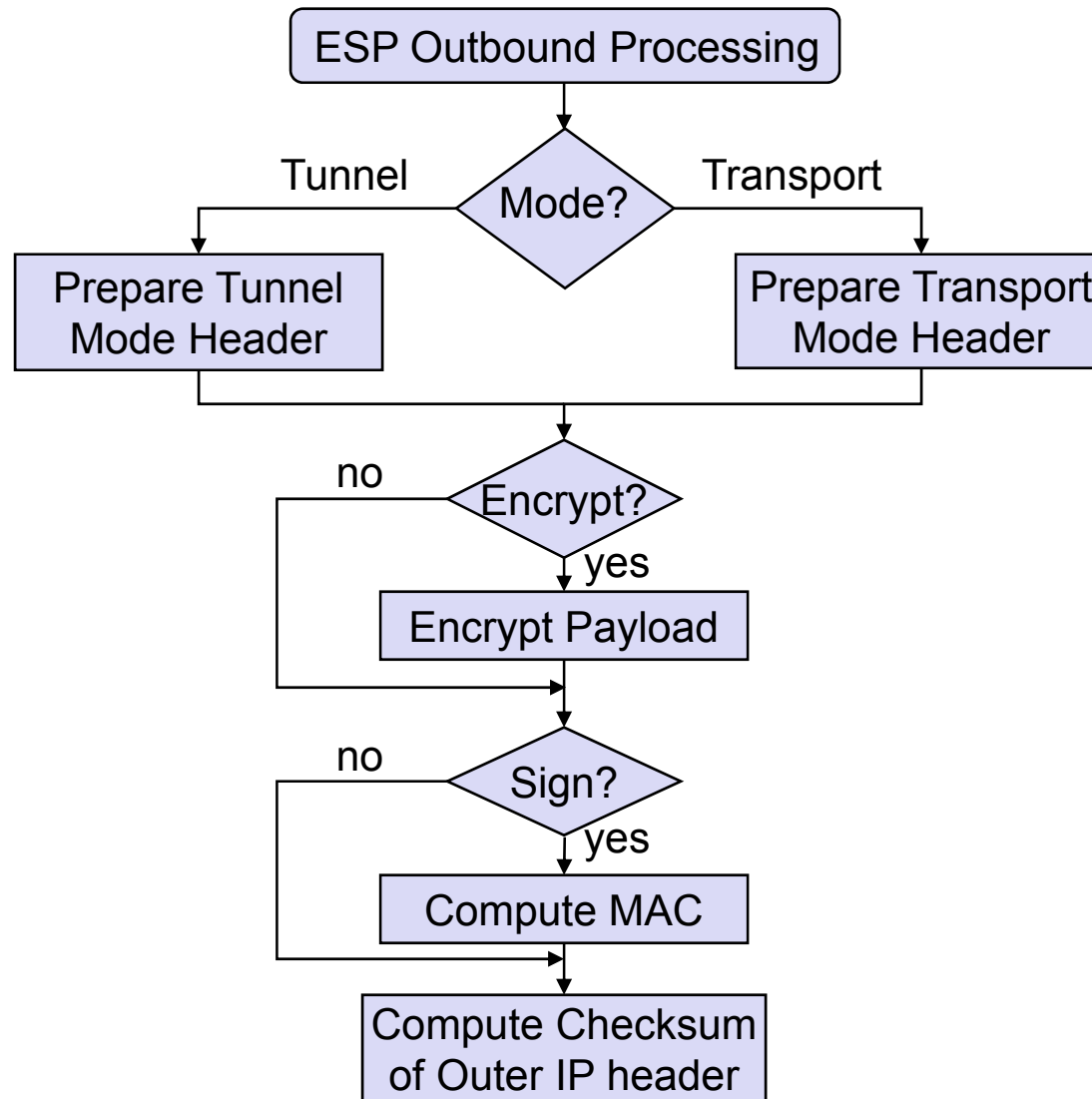
- ❑ ESP constitutes a generic security protocol that provides to IP packets replay protection and one or both of the following security services:
  - ❑ Confidentiality, by encrypting encapsulated packets or just their payload
  - ❑ Data origin authentication, by creating and adding MACs to packets
- ❑ The ESP definition is divided up into two parts:
  - ❑ The definition of the base protocol [RFC2406]: definition of the header and trailer format, basic protocol processing, tunnel and transport mode operation
  - ❑ The use of specific cryptographic algorithms with ESP:
    - Encryption: DES-CBC [RFC2405], use of other ciphers [RFC2451]
    - Authentication: HMAC-MD5-96 [RFC2403], HMAC-SHA-96 [RFC2404]

# Encapsulating Security Payload (ESP) Protocol

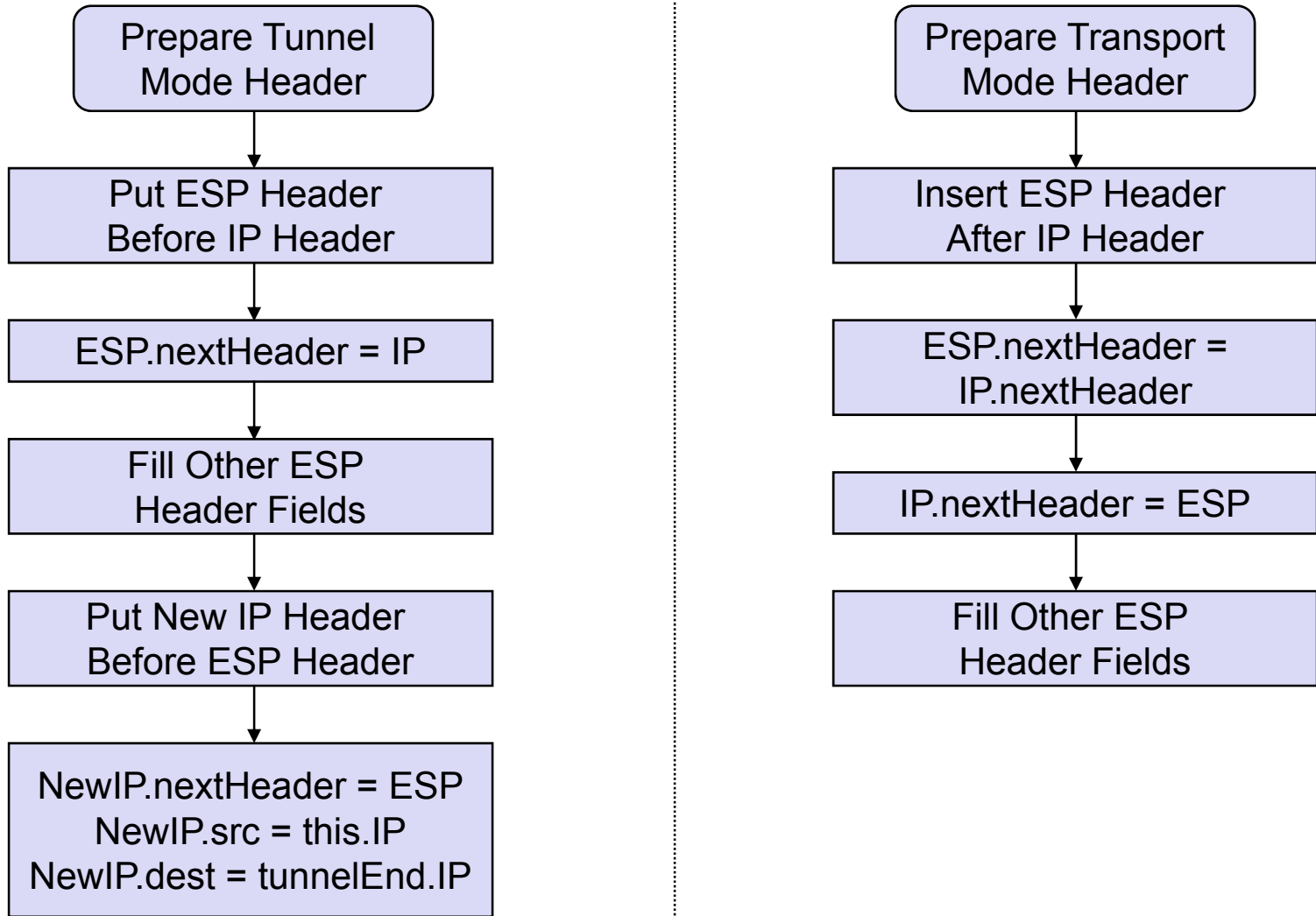


- ❑ The ESP header immediately follows an IP header or an AH header
- ❑ The next-header field of the preceding header indicates “50” for ESP

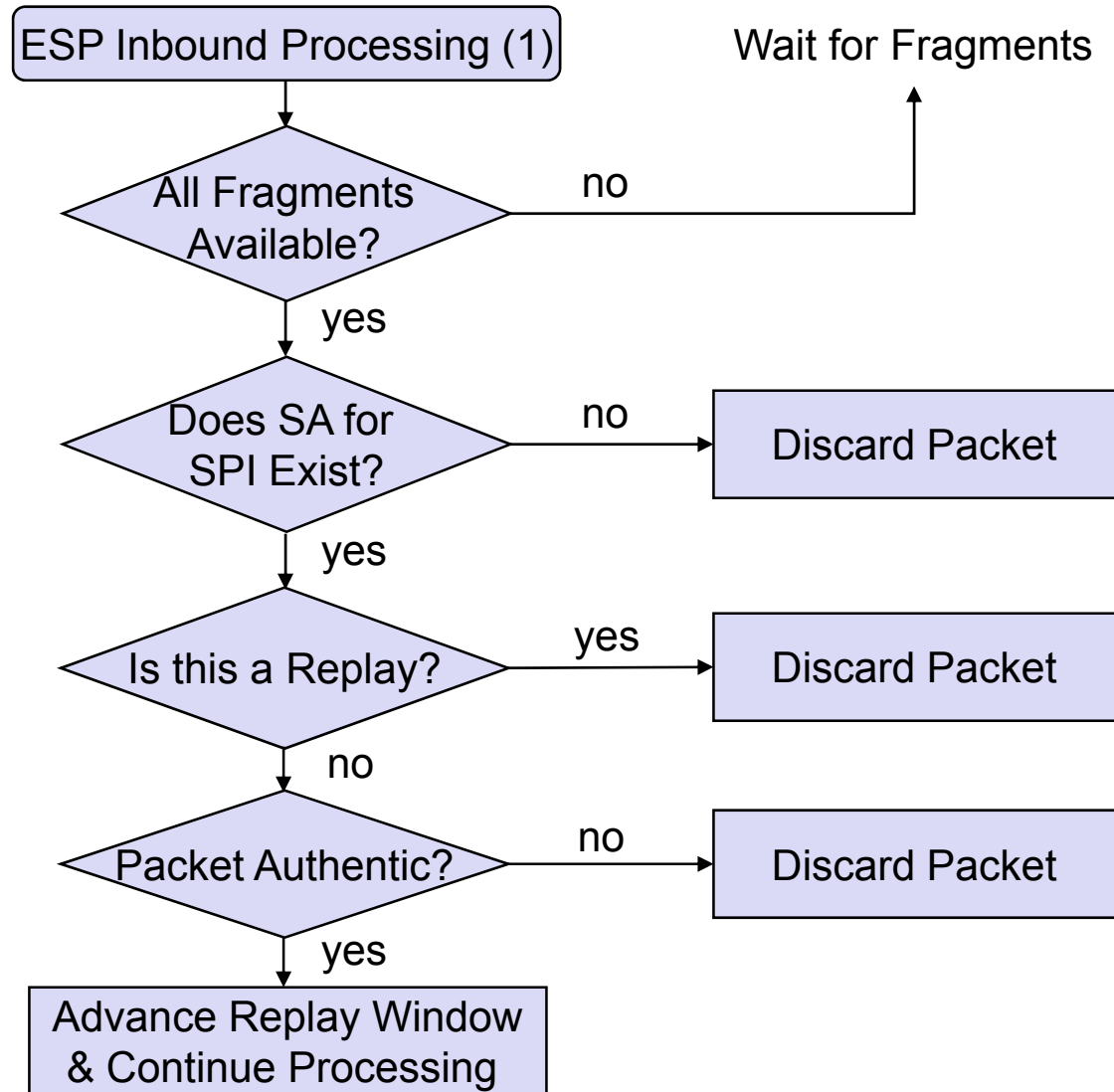
# Encapsulating Security Payload (ESP) Protocol



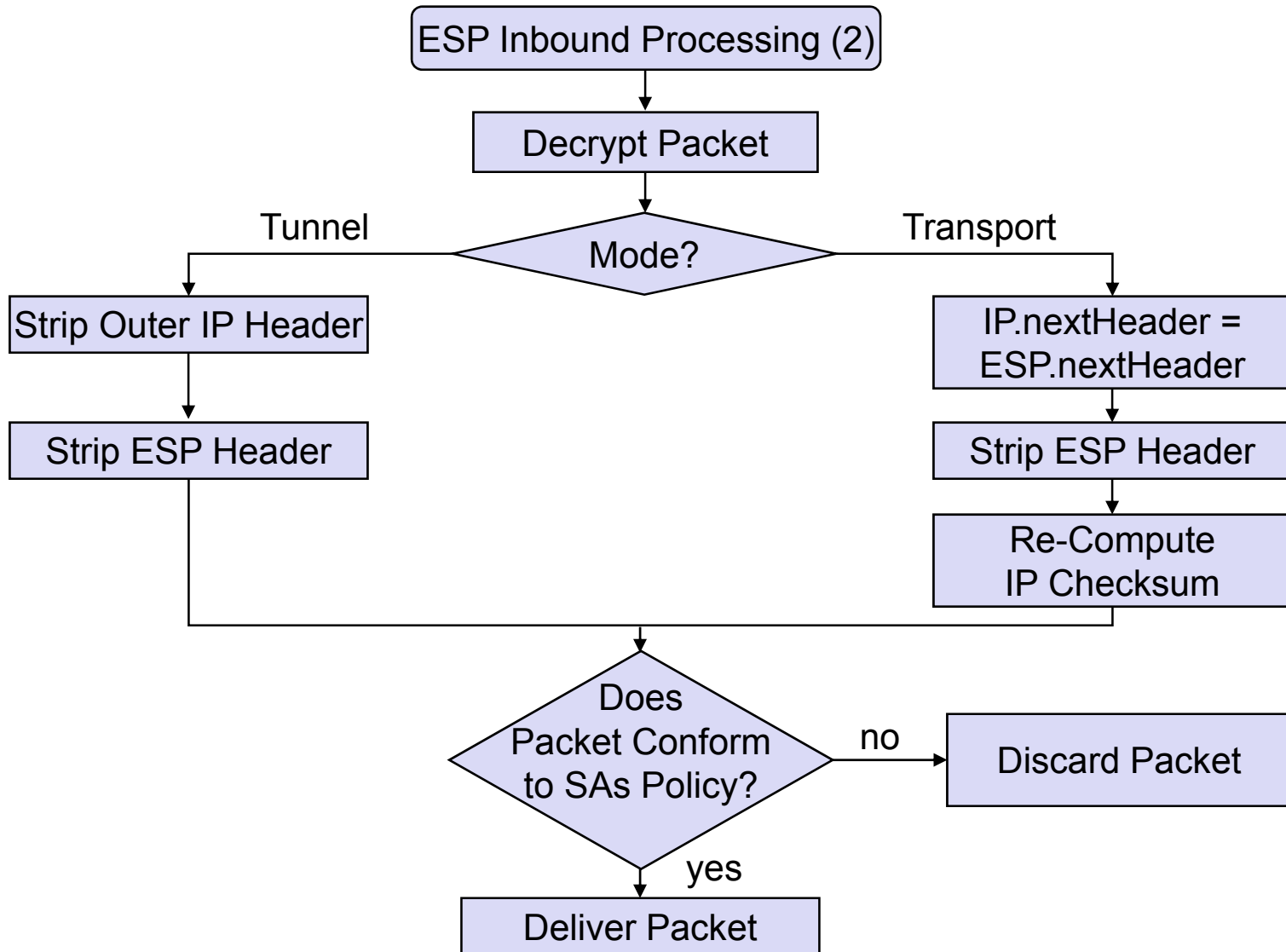
# Encapsulating Security Payload (ESP) Protocol



# Encapsulating Security Payload (ESP) Protocol



# Encapsulating Security Payload (ESP) Protocol



# Combination of AH and ESP

---

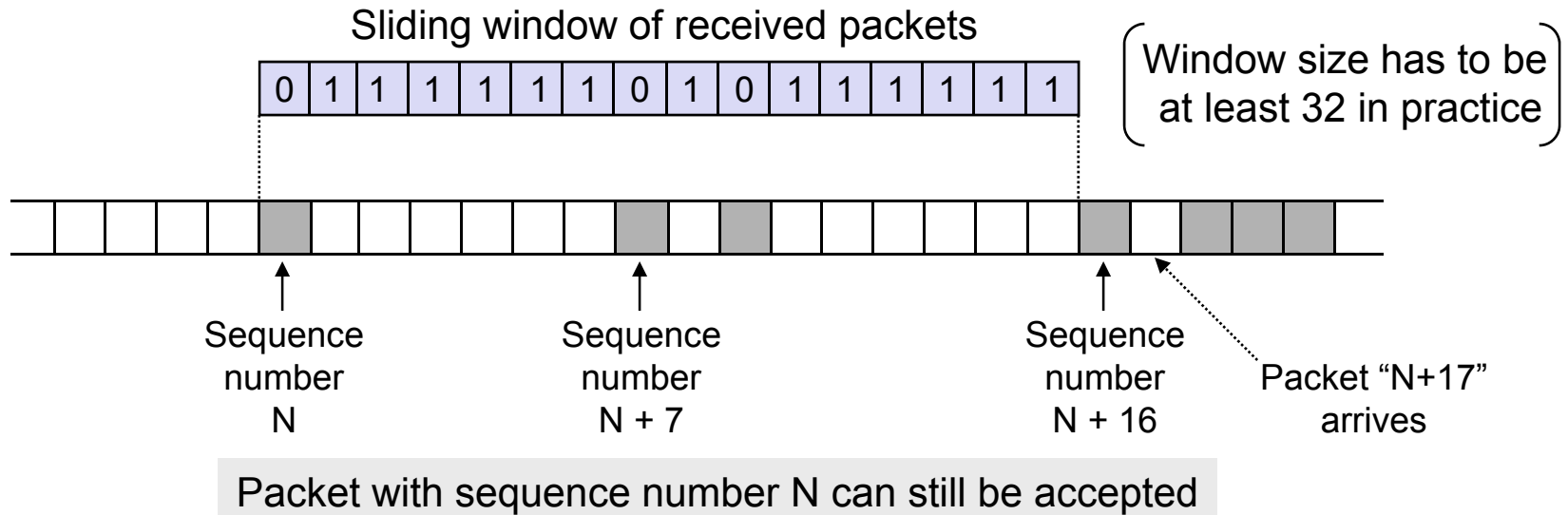


- ❑ If both ESP and AH are to be applied *by one entity*, then ESP is always applied first:
  - ❑ This results in AH being the outer header
  - ❑ Advantage: the ESP header can also be protected by AH
  - ❑ Remark: two SAs (one for each AH, ESP) are needed for each direction

# IPSec Replay Protection



- ❑ Both AH- and ESP-protected IP packets carry a sequence number which realizes a replay protection:
  - ❑ When setting up an SA this sequence number is initialized to zero
  - ❑ The sequence number is increased with every IP packet sent
  - ❑ A new session key is needed before a wrap-around occurs
  - ❑ The receiver of an IP packet checks, if the sequence number is contained in a window of acceptable numbers



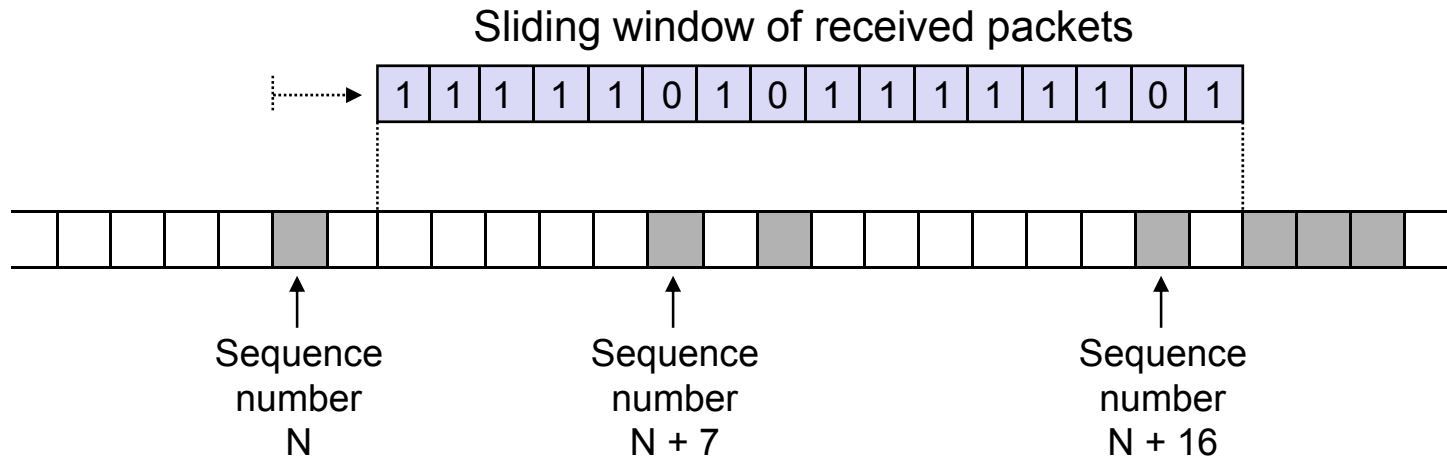
# IPSec Replay Protection



- ❑ If a received packet has a sequence number which:
  - ❑ is to the left of the current window  $\Rightarrow$  the receiver rejects the packet
  - ❑ is inside the current window  $\Rightarrow$  the receiver accepts the packet
  - ❑ is to the right of the current window  $\Rightarrow$  the receiver accepts the packet and advances the window

Of course IP packets are only accepted if they pass the authentication verification and the window is never advanced before this verification

- ❑ The minimum window size is 32 packets (64 packets is recommended)



Packet with sequence number N can no longer be accepted

# Security Associations

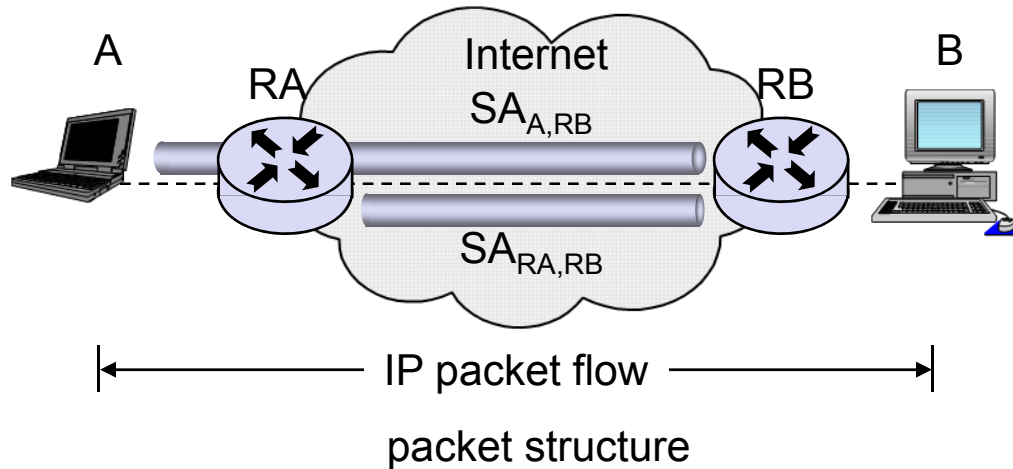


- ❑ A **security association (SA)** is a simplex “connection” that provides security services to the traffic carried by it
  - ❑ Security services are provided to one SA by the use of either AH or ESP, but not both
  - ❑ For bi-directional communication two security associations are needed
  - ❑ An SA is uniquely identified by a triple consisting of a **security parameter index (SPI)**, an IP destination address, and a security protocol identifier (AH / ESP)
  - ❑ An SA can be set up between the following peers:
    - Host ↔ Host
    - Host ↔ Gateway (or vice versa)
    - Gateway ↔ Gateway
  - ❑ There are two conceptual databases associated with SAs:
    - The **security policy database (SPD)** specifies, what security services are to be provided to which IP packets and in what fashion
    - The **security association database (SADB)**

# Nesting of Security Associations



- Security associations may be nested:
  - Example: Host A and gateway RB perform data origin authentication and gateways RA and RB perform subnetwork-to-subnetwork confidentiality

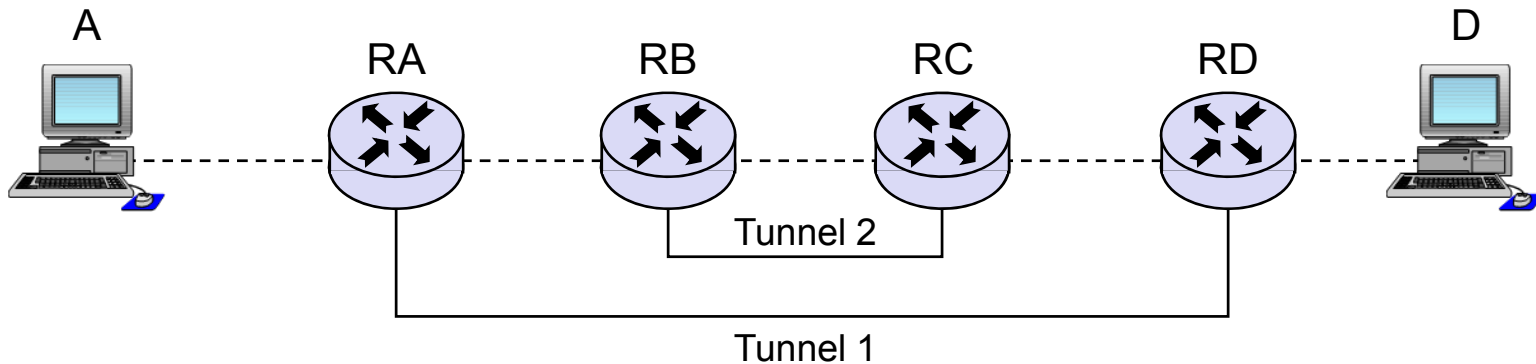


IP header	IPSec header	IP header	IPSec header	IP header	protected data
Src = RA Dst = RB		Src = A Dst = RB		Src = A Dst = B	

# Nesting of Security Associations



- ❑ However, one has to take care when nesting SAs that there occurs no “incorrect bracketing” of SAs, like “[([)]”
- ❑ One example of valid SA nesting:



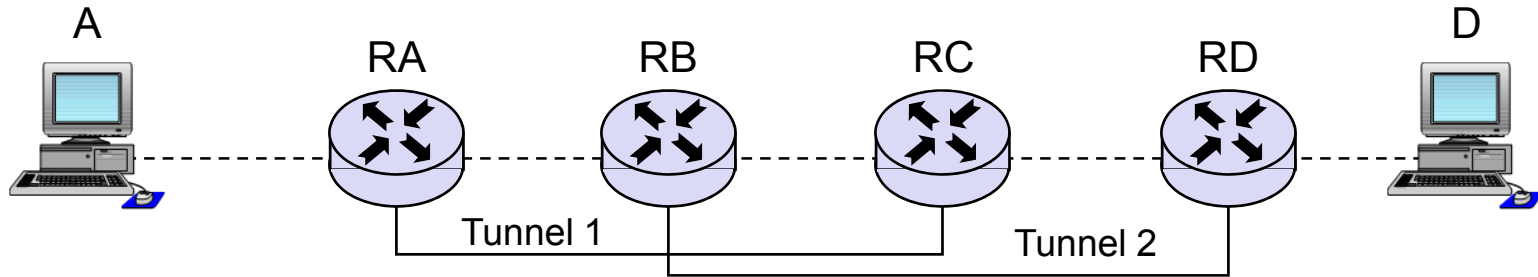
Packet Structure

IP header	IPSec header	IP header	IPSec header	IP header	protected data
Src = RB Dst = RC		Src = RA Dst = RD		Src = A Dst = D	

# Nesting of Security Associations



- ❑ One example of invalid SA nesting:



Packet Structure

IP header	IPSec header	IP header	IPSec header	IP header	protected data
Src = RB Dst = RD		Src = RA Dst = RC		Src = A Dst = D	

- ❑ As the packet is tunneled from RB to RD the gateway RC can not process the inner IPSec header
- ❑ A possible result of this faulty configuration could be that the packet is routed back to RC

# IPSec Security Policy Selection

---



- ❑ The following selectors to be extracted from the network and transport layer headers allow to select a specific policy in the SPD:
  - ❑ *IP source address:*
    - Specific host , network prefix, address range, or wildcard
  - ❑ *IP destination address:*
    - Specific host , network prefix, address range, or wildcard
    - In case of incoming tunneled packets the inner header is evaluated
  - ❑ *Name:*
    - DNS name, X.500 name or other name types as defined in IPSec domain of interpretation of a protocol for setting up SAs
    - Only used during SA negotiation
  - ❑ *Protocol:*
    - The protocol identifier of the transport protocol for this packet
    - This may not be accessible when a packet is secured with ESP
  - ❑ *Upper layer ports:*
    - If accessible, the upper layer ports for session oriented policy selection

# IPSec Security Policy Definition



- ❑ Policy selectors are used to select specific policy definitions
- ❑ A policy definition specifies:
  1. How to perform setup of an IKE SA between two nodes:
    - *Phase I mode*: main mode or aggressive mode
    - *Protection suite(s)*: specify how IKE authentication is performed
  2. Which and how security services should be provided to IP packets:
    - *Selectors*, that identify specific flows
    - *Security attributes* for each flow:
      - *Security protocol*: AH or ESP
      - *Protocol mode*: transport or tunnel mode
      - *Security transforms*: cryptographic algorithms and parameters
      - *Other parameters*: SA lifetime, replay window
    - *Action*: discard, secure, bypass
- ❑ If an SA is already established with a corresponding security endpoint, it is referenced in the SPD

# Establishment of Security Associations

---



- ❑ Prior to any packet being protected by IPSec, an SA has to be established between the two “cryptographic endpoints” providing the protection
- ❑ SA establishment can be realized:
  - ❑ Manually, by proprietary methods of systems management
  - ❑ Dynamically, by a standardized authentication & key management protocol
  - ❑ Manual establishment is supposed to be used only in very restricted configurations (e.g. between two encrypting firewalls of a VPN) and during a transition phase
- ❑ IPSec defines a standardized method for SA establishment:
  - ❑ *Internet Security Association and Key Management Protocol (ISAKMP)*
    - Defines protocol formats and procedures for security negotiation
  - ❑ *Internet Key Exchange (IKE)*
    - Defines IPSec’s standard authentication and key exchange protocol

# ISAKMP – Introduction

---



- ❑ The IETF has adopted two RFCs on ISAKMP for IPsec:
  - ❑ RFC 2408, which defines the ISAKMP base protocol
  - ❑ RFC 2407, which defines IPsec's “*domain of interpretation*” (*DOI*) for ISAKMP further detailing message formats specific for IPsec
- ❑ The ISAKMP base protocol is a generic protocol, that can be used for various purposes:
  - ❑ The procedures specific for one application of ISAKMP are detailed in a *DOI document*
  - ❑ Other DOI documents have been produced:
    - Group DOI for secure group communication (Internet Draft, Sep. 2000)
    - MAP DOI for use of ISAKMP to establish SAs for securing the *Mobile Application Protocol (MAP)* of GSM (Internet Draft, Nov. 2000)
- ❑ ISAKMP defines two fundamental categories of exchanges:
  - ❑ Phase 1 exchanges, which negotiate some kind of “Master SA”
  - ❑ Phase 2 exchanges, which use the “Master SA” to establish other SAs

# ISAKMP – Limited Denial of Service Protection



- ❑ The initiator and responder cookies also serve as a protection against simple denial of service attacks:
  - ❑ Authentication and key exchange often requires “expensive” computations, e.g. exponentiation (for Diffie-Hellman key exchange)
  - ❑ In order to avoid, that an attacker can easily flood an ISAKMP entity with bogus messages from forged source addresses and cause these expensive operations, the following scheme is used:
    - The initiating ISAKMP entity generates an initiator cookie:  
 $CKY-I = H(\text{Secret}_{\text{Initiator}}, \text{Address}_{\text{Responder}}, t_{\text{Initiator}})$
    - The responder generates his own cookie:  
 $CKY-R = H(\text{Secret}_{\text{Responder}}, \text{Address}_{\text{Initiator}}, t_{\text{Responder}})$
    - Both entities always include both cookies, and always check *their own cookie* before performing any expensive operation
    - The attack mentioned above will, therefore, not be successful as the attacker needs to receive a response from the attacked system in order to obtain a cookie from it
  - ❑ ISAKMP does not specify the exact cookie generation method

# ISAKMP – SA Negotiation

---



- ❑ The proposal payload provides the initiating entity with the capability to present to the responding entity the security protocols and associated security mechanisms for use with the security association being negotiated
- ❑ If the SA establishment negotiation is for a combined *protection suite* consisting of multiple protocols, then there must be multiple proposal payloads each with the same proposal number
- ❑ These proposals must be considered as a unit and must not be separated by a proposal with a different proposal number

# ISAKMP – SA Negotiation



- ❑ This example shows an ESP AND AH protection suite:
  - ❑ The first protocol is presented with two transforms supported by the proposing entity, ESP with:
    - Transform 1 as 3DES; Transform 2 as DES
    - The responder must select from the two transforms proposed for ESP
  - ❑ The second protocol is AH and is presented with a single transform:
    - Transform 1 as SHA
  - ❑ The resulting protection suite will be either:
    - 3DES and SHA, or
    - DES and SHA
  - ❑ In this case, the SA payload will be followed by the following payloads:
    - [Proposal 1, ESP, (Transform 1, 3DES, ...), (Transform 2, DES)]
    - [Proposal 1, AH, (Transform 1, SHA)]
  
- ❑ Please remark, that this will result in two SAs per direction!

# ISAKMP – SA Negotiation



- ❑ This example shows a proposal for two different protection suites:
  - ❑ The first protection suite is presented with:
    - one transform (MD5) for the first protocol (AH), and
    - one transform (3DES) for the second protocol (ESP)
  - ❑ The second protection suite is presented with two transforms for a single protocol (ESP):
    - 3DES, or
    - DES
  - ❑ Please note, that it is not possible to specify that transform 1 and transform 2 have to be used for one instance of a protocol specification
  - ❑ In this case, the SA payload will be followed by the following payloads:
    - [Proposal 1, AH, (Transform 1, MD5, ...)]
    - [Proposal 1, ESP, (Transform 1, 3DES, ...)]
    - [Proposal 2, ESP, (Transform1, 3DES, ...), (Transform 2, DES, ...)]
  
- ❑ Please note, that proposal 1 results in two SAs per direction.

# IKE – Introduction

---



- ❑ Whereas ISAKMP defines the basic data formats and procedures to negotiate arbitrary SAs, the *Internet Key Exchange* specifies the standardized protocol to negotiate IPSec SAs
- ❑ IKE defines five exchanges:
  - ❑ Phase 1 exchanges for establishment of an IKE SA :
    - *Main mode exchange* which is realized by 6 exchanged messages
    - *Aggressive mode exchange* which needs only 3 messages
  - ❑ Phase 2 exchange for establishment of IPSec SAs:
    - *Quick mode exchange* which is realized with 3 messages
  - ❑ Other exchanges:
    - *Informational exchange* to communicate status and error messages
    - *New group exchange* to agree upon private Diffie-Hellman groups

# IKE – Computation of IKE Session Keys



- ❑ IKE establishes four different keys with an authentication exchange:
  - ❑ *SKEYID* is a string derived from secret material known only to the active players in the exchange and it serves as a “master key”
    - The computation of *SKEYID* depends on the authentication method
  - ❑ *SKEYID\_d* is the keying material used to derive keys for non-IKE SAs
    - $SKEYID_d = H(SKEYID, g^{xy}, CKY-I, CKY-R, 0)$   
with  $g^{xy}$  denoting the shared Diffie-Hellman secret
  - ❑ *SKEYID\_a* is the keying material used by the IKE SA to authenticate its messages
    - $SKEYID_a = H(SKEYID, SKEYID_d, g^{xy}, CKY-I, CKY-R, 1)$
  - ❑ *SKEYID\_e* is the keying material used by the IKE SA to protect the confidentiality of its messages
    - $SKEYID_e = H(SKEYID, SKEYID_a, g^{xy}, CKY-I, CKY-R, 2)$
- ❑ If required, keys are expanded by the following method:
  - ❑  $K = (K_1, K_2, \dots)$  with  $K_i = H(SKEYID, K_{i-1})$  and  $K_0 = 0$

# IKE – Authentication Methods

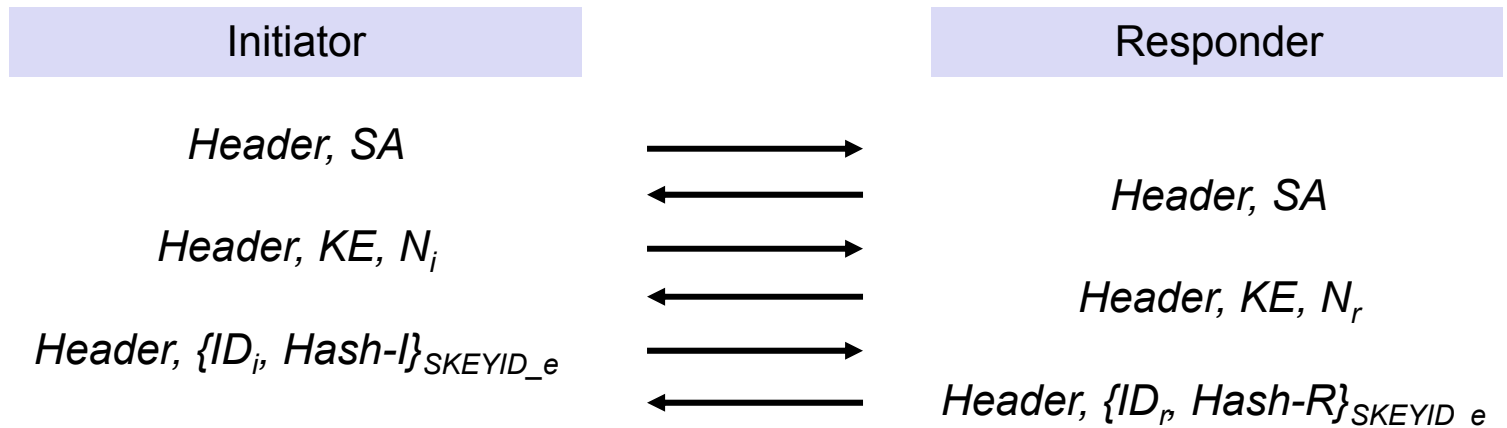


- ❑ Phase 1 IKE exchanges are authenticated with the help of two hash values *Hash-I* and *Hash-R*, created by the initiator and responder:
  - ❑  $Hash-I = H(SKEYID, g^x, g^y, CKY-I, CKY-R, SA-offer, ID-I)$
  - ❑  $Hash-R = H(SKEYID, g^y, g^x, CKY-R, CKY-I, SA-offer, ID-R)$where  $g^x, g^y$  denote the exchanged public Diffie-Hellman values  
*ID-I, ID-R* denote the identity of the initiator and the responder  
*SA-offer* denotes the payloads concerning SA negotiation
- ❑ IKE supports four different methods of authentication:
  - ❑ Pre-shared key:
    - $SKEYID = H(K_{Initiator, Responder}, r_{Initiator}, r_{Responder})$
  - ❑ Two different forms of authentication with public-key encryption:
    - $SKEYID = H(H(r_{Initiator}, r_{Responder}), CKY-I, CKY-R)$
  - ❑ Digital Signature:
    - $SKEYID = H(r_{Initiator}, r_{Responder}, g^{xy})$
    - As in this case *SKEYID* itself provides no authentication, the values *Hash-I* and *Hash-R* are signed by the initiator / responder

# IKE – Main Mode Exchange with Pre-Shared Key

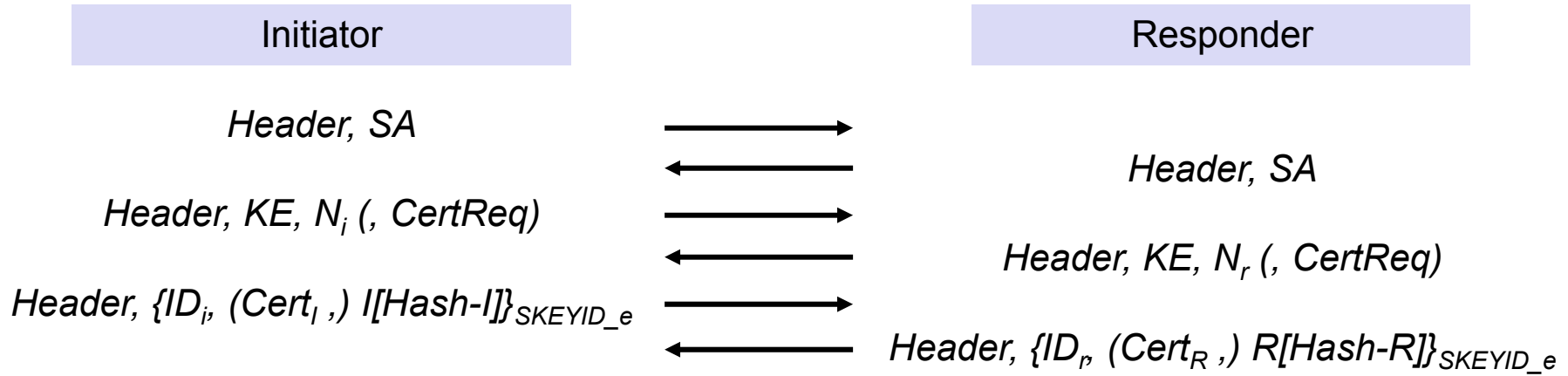


- The following descriptions list the exchanged ISAKMP- and IKE-payloads when performing different “flavors” of IKE authentication:



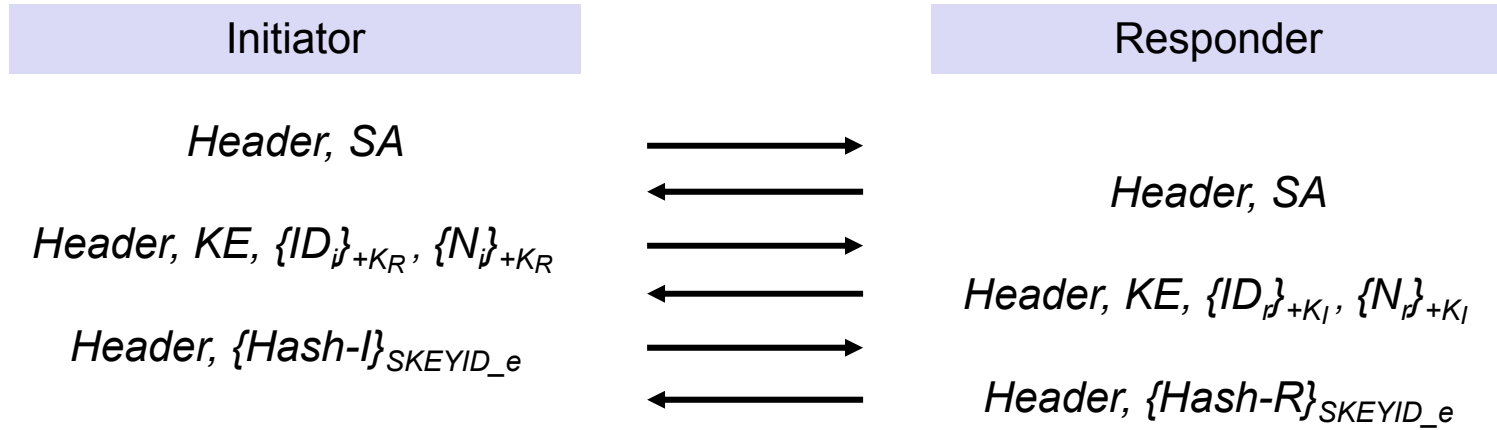
- where:  $N_i, N_r$  denote  $r_{\text{Initiator}}, r_{\text{Responder}}$  (IKE notation)  
 $ID_i, ID_r$  denote the identity of the initiator and the responder  
 $KE$  denotes the public values of a DH-exchange
- Please note that *Hash-I* and *Hash-R* need not to be signed, as they already “contain an authentic secret” (pre-shared key)

# IKE – Main Mode Exchange with Signatures



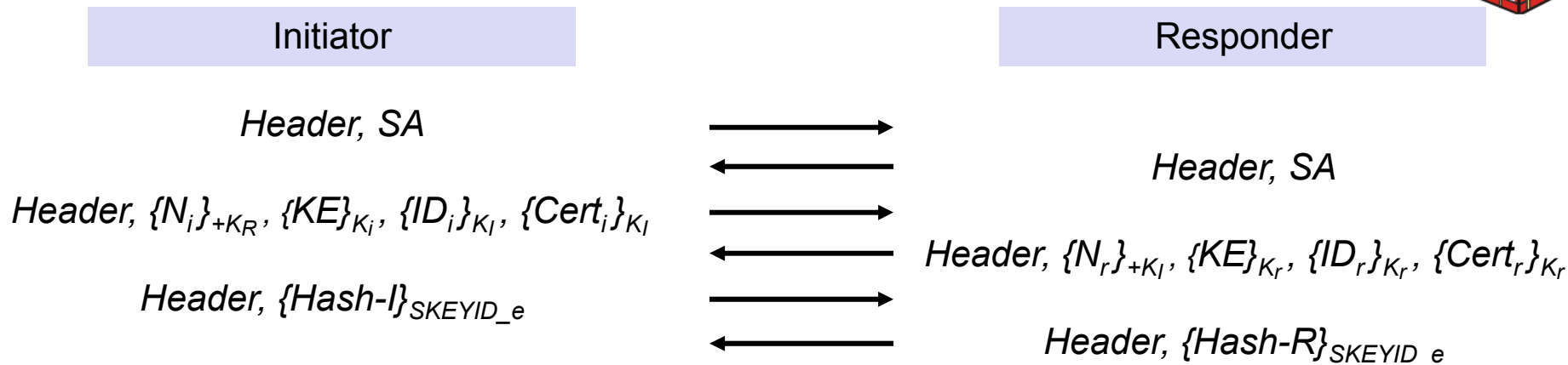
- where:  $(m)$  denotes that  $m$  is optional  
 $I[m]$  denotes that  $I$  signs  $m$
- Please note that  $Hash-I$  and  $Hash-R$  need to be signed, as they do not contain anything which is known to be authentic

# IKE – Main Mode Exchange with Public Key Encryption



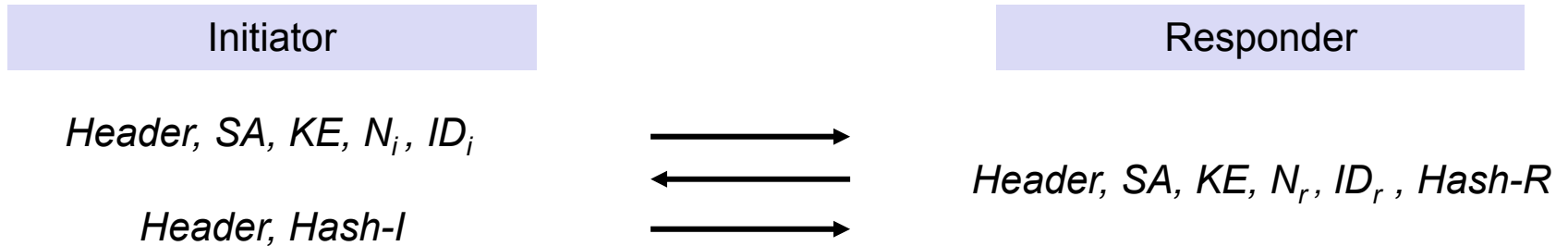
- ❑ where:  $\{m\}_{+K_I}$  denotes that  $m$  is encrypted with the public key  $+K_I$
- ❑ Please note that *Hash-I* and *Hash-R* need not to be signed, as they “contain” the exchanged random numbers  $N_i$  or  $N_r$ , respectively
  - ❑ So, every entity proves his authenticity by decrypting the received random number ( $N_i$  or  $N_r$ ) with its’ private key

# IKE – Main Mode Exchange with Public Key Encryption



- where:  $\{m\}_{+K_i}$  denotes that  $m$  is encrypted with the public key  $+K_i$   
 $\{m\}_{K_i}$  denotes that  $m$  is encrypted with the symmetric key  $K_i$   
with  $K_i = H(N_i, CKY-I)$  and  $K_r = H(N_r, CKY-R)$
- Please note that all schemes described so far provide protection of identity against eavesdroppers in the Internet, as the IDs and certificates are not send in the clear:
  - However, the IP addresses of exchanged packets are always readable...

# IKE – Aggressive Mode Exchange with Pre-Shared Key



- ❑ As the identity of the initiator and the responder have to be sent before a session key can be established, the aggressive mode exchange can not provide identity protection against eavesdroppers
- ❑ There are similar aggressive mode variants for authentication with:
  - ❑ Digital signature
  - ❑ Public key encryption

# Basic Scheme of IPSec Processing: Outgoing Packets



- In order to support IPSec it has to perform the following steps:
  1. Determine if and how the outgoing packet needs to be secured:
    - This is realized by performing a lookup in the SPD
    - If the policy specifies “discard” then drop the packet ⇒ done
    - If the packet does not need to be secured, then send it ⇒ done
  2. Determine which SA should be applied to the packet:
    - If there is not yet an appropriate SA established, then ask the key management demon to perform an IKE
  3. Look up the determined (and eventually freshly created) SA in the SADB
  4. Perform the security transform determined by the SA by using the algorithm, its’ parameters and the key as specified in the SA
    - This results in the construction of an AH or an ESP header
    - Eventually also a new (outer) IP header will be created (tunnel mode)
  5. Send the resulting IP packet ⇒ done

# Basic Scheme of IPSec Processing: Incoming Packets

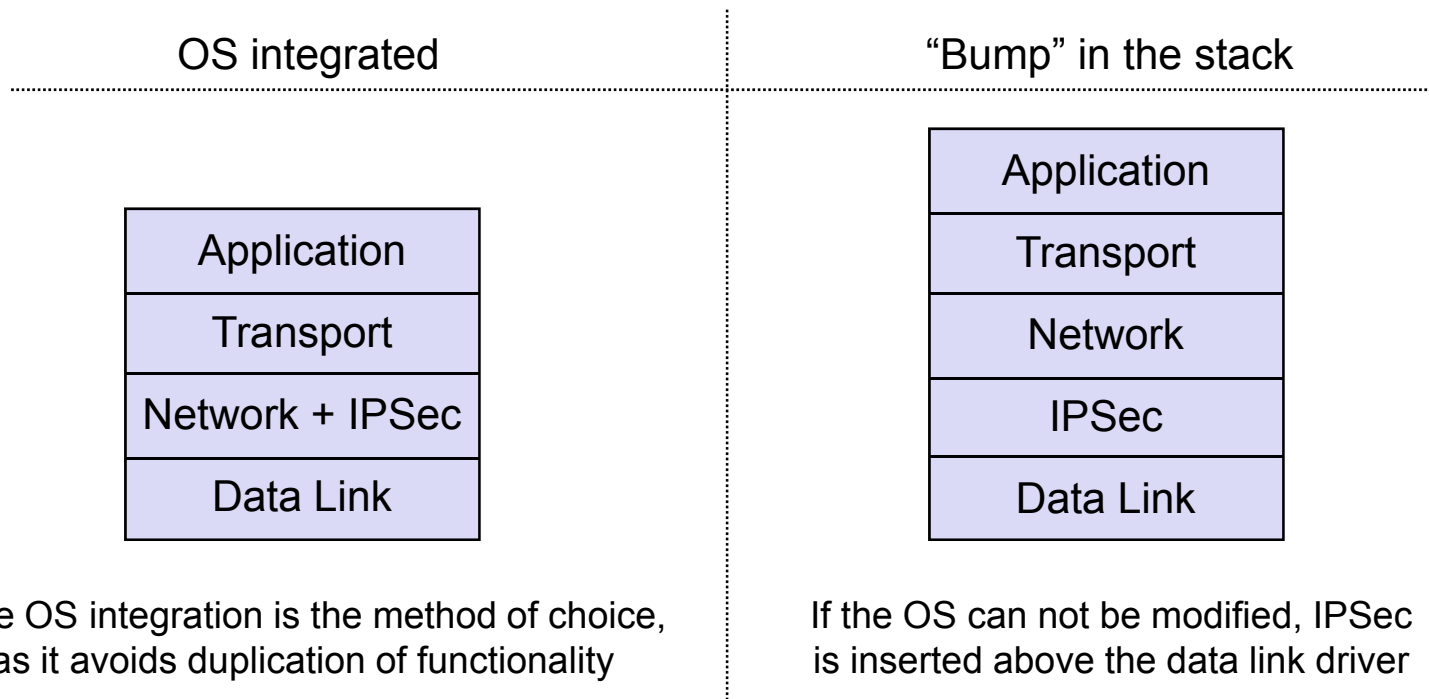


- ❑ In order to support IPSec it has to perform the following steps:
  1. Determine if the packet contains an IPSec header this entity is supposed to process:
    - If there is such an IPSec header then look up the SA in the SADB which is specified by the SPI of the IPSec header and perform the appropriate IPSec processing
    - If the SA referenced by the SPI does not (yet) exist, drop the packet
  2. Determine if and how the packet should have been protected:
    - This is again realized by performing a lookup in the SPD, with the lookup being performed by evaluating the inner IP header in case of tunneled packets
    - If the policy specifies “discard” then drop the packet
    - If the protection of the packet did not match the policy, drop the packet
  3. If the packet had been properly secured, then deliver it to the appropriate protocol entity (network / transport layer)

# IPSec Implementation Alternatives: Host Implementation



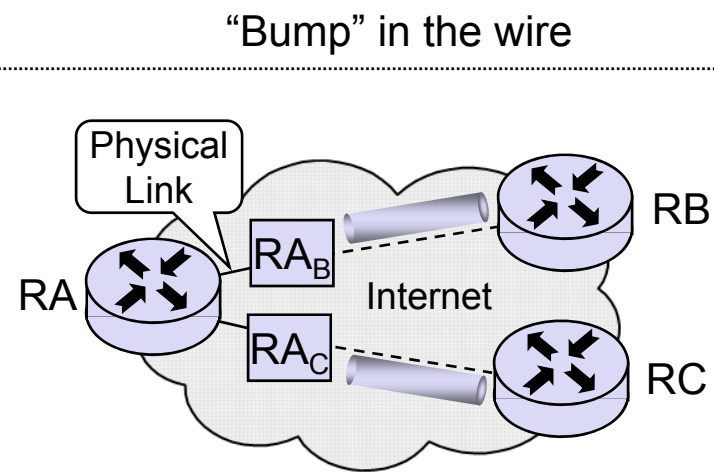
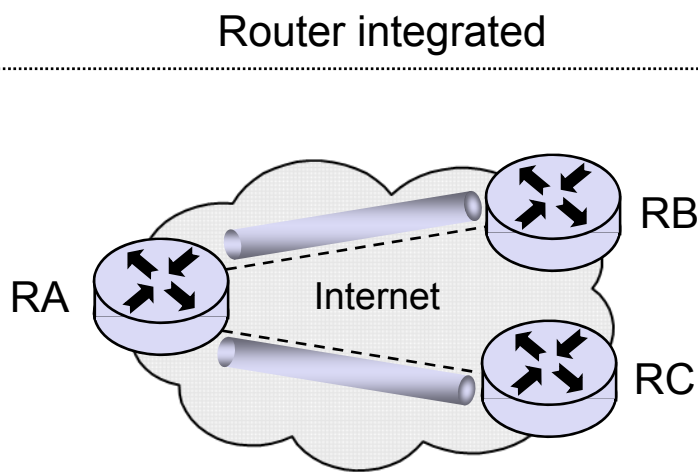
- ❑ Advantages of IPSec implementation in end systems:
  - ❑ Provision of end-to-end security services
  - ❑ Provision of security services on a per-flow basis
  - ❑ Ability to implement all modes of IPSec
- ❑ Two main integration alternatives:



# IPSec Implementation Alternatives: Router Implementation



- ❑ Advantages of IPSec implementation in routers:
  - ❑ Ability to secure IP packets flowing between two networks over a public network such as the Internet:
    - Allows to create *virtual private networks (VPNs)*
    - No need to integrate IPSec in every end system
  - ❑ Ability to authenticate and authorize IP traffic coming in from remote users
- ❑ Two main implementation alternatives:



# Further Issues with IPSec

---



- ❑ Compression:
  - ❑ If encryption is used, then the resulting IP packets can not be compressed in the link layer, e.g. when connecting to an ISP via Modem
  - ❑ Therefore, the *IP payload compression protocol (PCP)* has been defined
  - ❑ PCP can be used with IPSec:
    - IPSec policy definition allows to specify PCP
    - IKE SA negotiation allows to include PCP in proposals
- ❑ Interoperability problems of end-to-end security with header processing in intermediate nodes:
  - ❑ Interoperability with firewalls:
    - End-to-end encryption conflicts with the firewalls' need to inspect upper layers protocol headers in IP packets
  - ❑ Interoperability with network address translation (NAT):
    - Encrypted packets do neither permit analysis nor change of addresses
    - Authenticated packets will be discarded if source or destination address is changed

# Summary (what do I need to know)

---



- ❑ IPSec provides the following security services to IP packets
  - ❑ Data origin authentication, confidentiality, replay protection
  
- ❑ Two operation modes
  - ❑ Tunnel mode
  - ❑ Transport mode
  
- ❑ Two fundamental security protocols have been defined
  - ❑ Authentication header (AH)
  - ❑ Encapsulating security payload (ESP)
  
- ❑ SA negotiation and key management (overview)
  - ❑ Internet security association key management protocol (ISAKMP)
  - ❑ Internet key exchange (IKE)

# Additional References

---



- [RFC2401] R. Atkinson, S. Kent. *Security Architecture for the Internet Protocol*. RFC 2401, Internet Engineering Taskforce (IETF), 1998.
- [RFC2402] R. Atkinson, S. Kent. *IP Authentication Header (AH)*. RFC 2402, IETF, 1998.
- [RFC2403] C. Madson, R. Glenn. *The Use of HMAC-MD5-96 within ESP and AH*. RFC 2403, IETF, 1998.
- [RFC2404] C. Madson, R. Glenn. *The Use of HMAC-SHA-1-96 within ESP and AH*. RFC 2404, IETF, 1998.
- [RFC2405] C. Madson, N. Doraswami. *The ESP DES-CBC Cipher Algorithm With Explicit IV*. RFC 2405, IETF, 1998.
- [RFC2406] R. Atkinson, S. Kent. *IP Encapsulating Security Payload (ESP)*. RFC 2406, IETF, 1998.
- [RFC2407] D. Piper. *The Internet IP Security Domain of Interpretation for ISAKMP*. RFC 2407, IETF, 1998.
- [RFC2408] D. Maughan, M. Schertler, M. Schneider, J. Turner. *Internet Security Association and Key Management Protocol (ISAKMP)*. RFC 2408, IETF, 1998.
- [RFC2409] D. Harkins, D. Carrel. *The Internet Key Exchange (IKE)*. RFC 2409, IETF, 1998.
- [RFC2857] A. Keromytis, N. Provos. *The Use of HMAC-RIPEND-160-96 within ESP and AH*. RFC 2857, IETF, 2000.