Resilient Networking

Thorsten Strufe

Module 4: IPsec

Disclaimer: Parts of these slides are taken from the lecture „Network Security“ at TU Ilmenau (Schäfer, Rossberg)

Dresden, SS 16

Overview

Brief introduction to the Internet Protocol (IP) suite

Security problems of IP and objectives of IPsec

The IPsec architecture

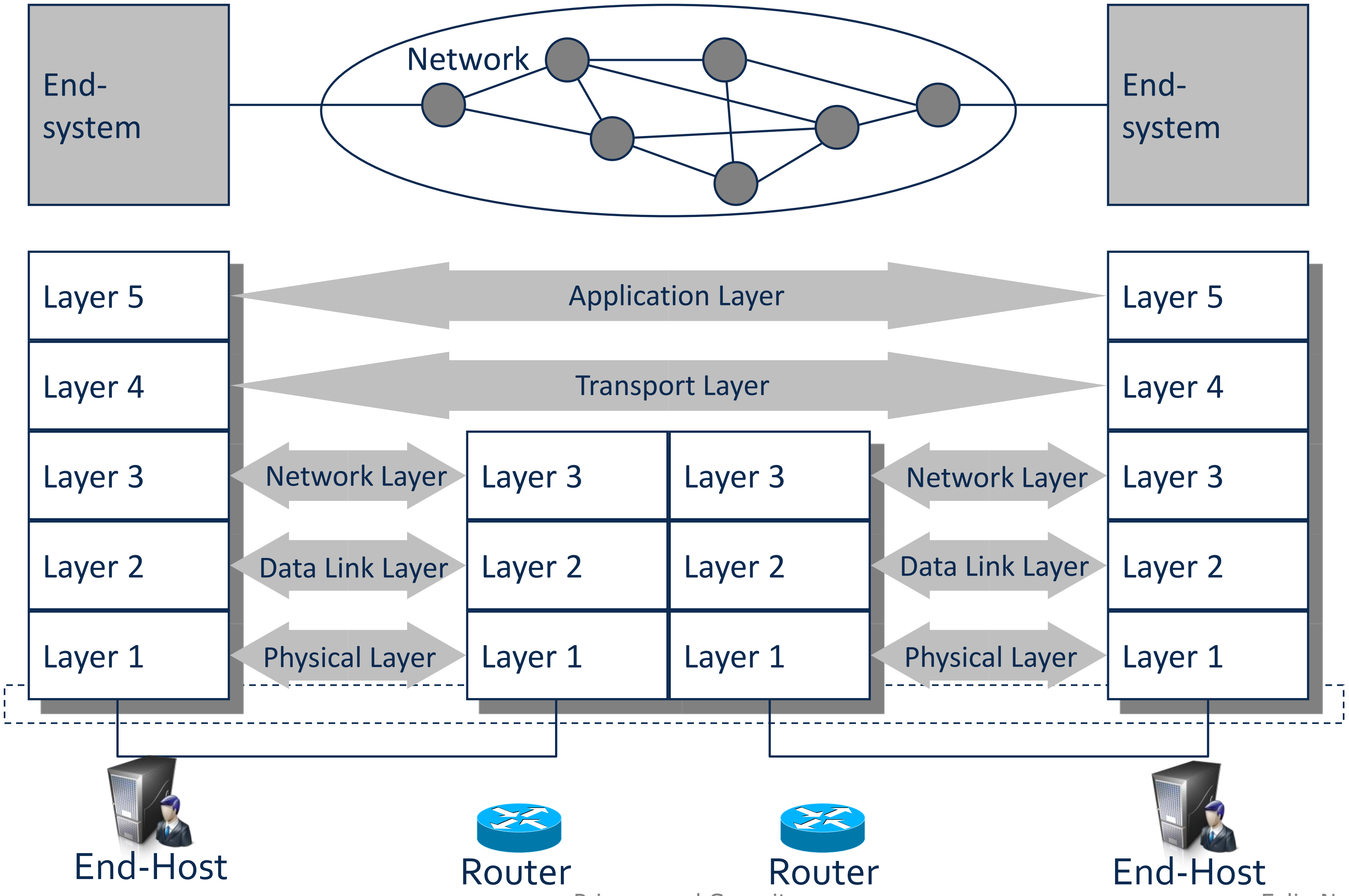
IPsec security protocols

- Authentication Header (AH)
- Encapsulating Security Payload (ESP)

Entity Authentication and the Internet Key Exchange (IKE)

Architectural View of Threatened "Object"

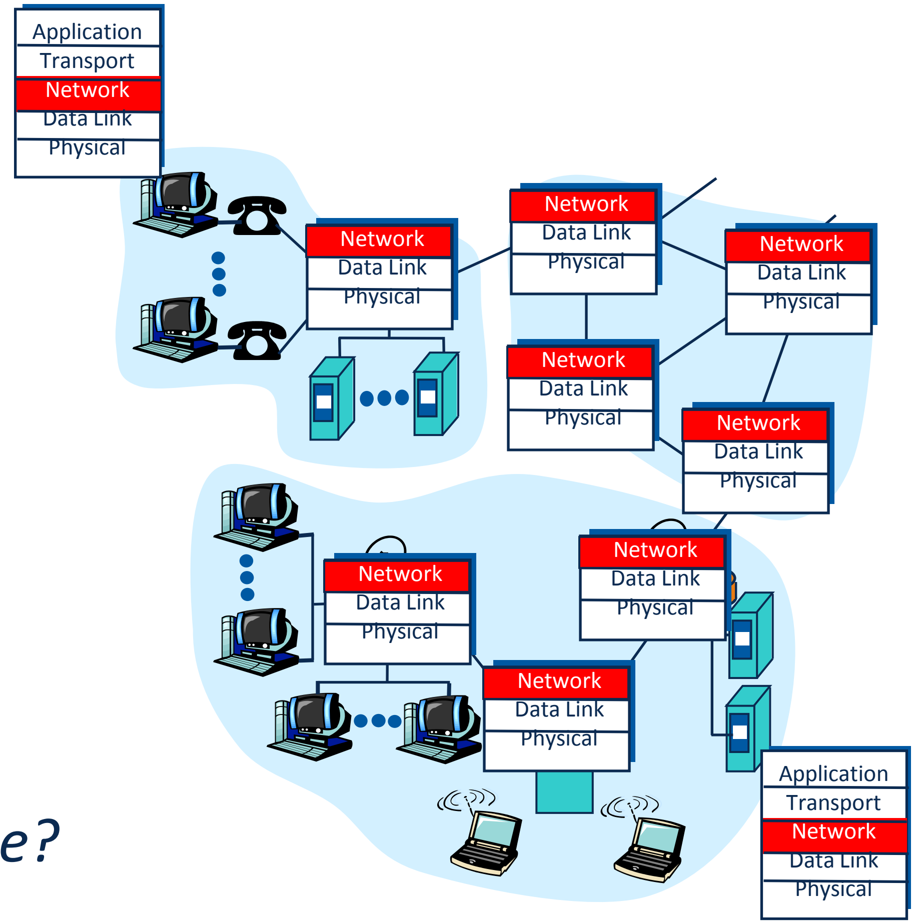
Communication in Layered Protocol Architectures



The General Internet Structure

Communication End-Points

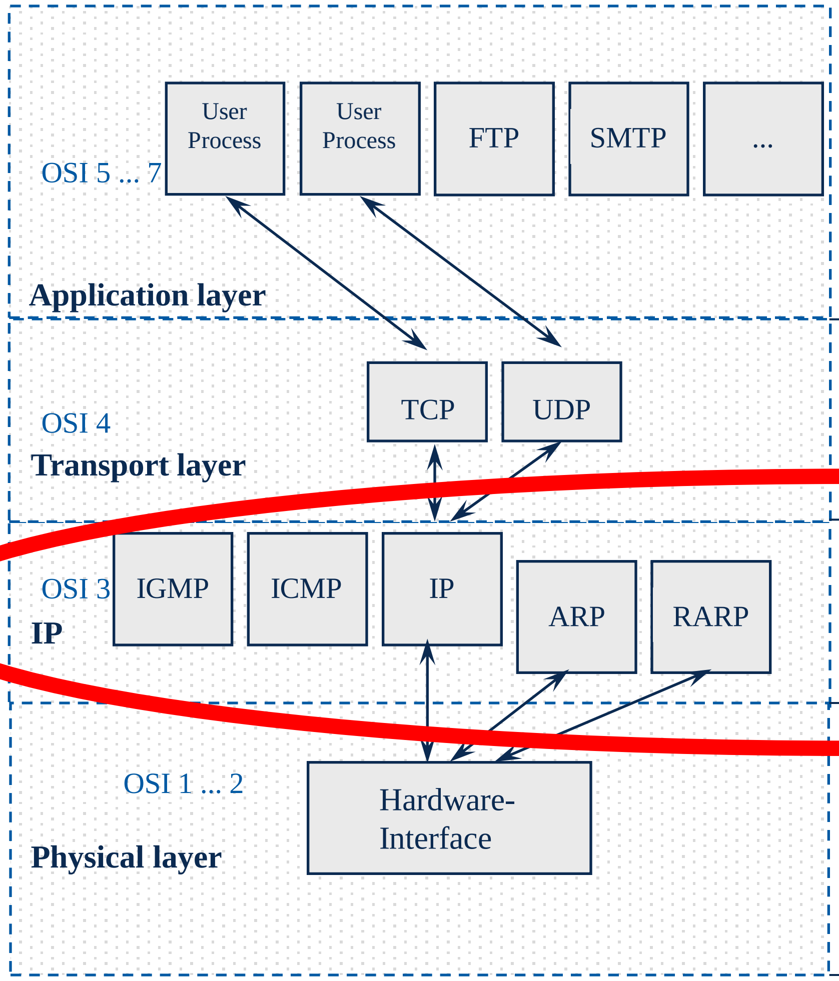
- AL: Application
- TL: Socket
- NL: End Host
- DL/NW: Point to Point



From whom do we protect, where?

Layered Models and Internet - Protocols

Layered Models vs. Internet Protocols



Responsability (ISO/OSI)

Application Layer
- Dist Apps and their protocols

Presentation Layer
- Data representation

Session Layer
- Dialogues and sessions
- Synchronisation of data transfers

Transport Layer
- Connection oriented vs. connectionless services
- De-/multiplexing pakets/connections
- Flow control between sender and receiver

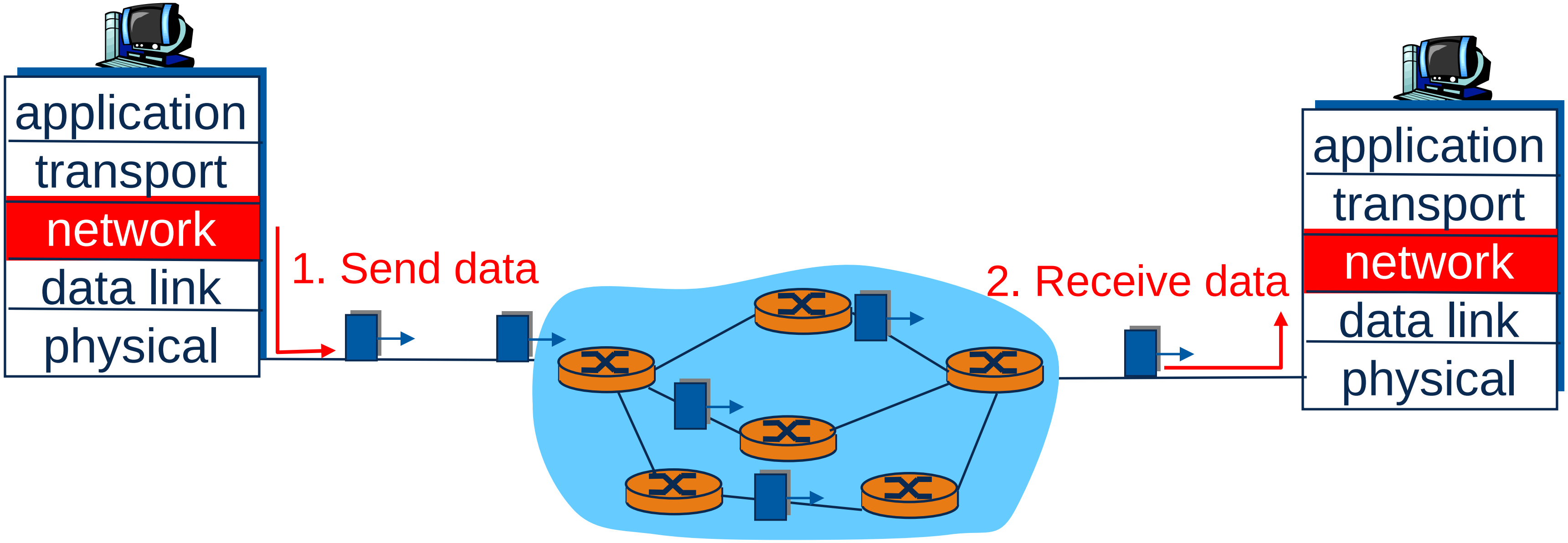
Network Layer
- Global addresses
- Routing
- Connection oriented vs. connectionless networks

Data Link Layer
- Encapsulating raw data in transmission frames
- Frame detection and acknowledgement
- Error control

Physical Layer
- Transmission of bits (signals)

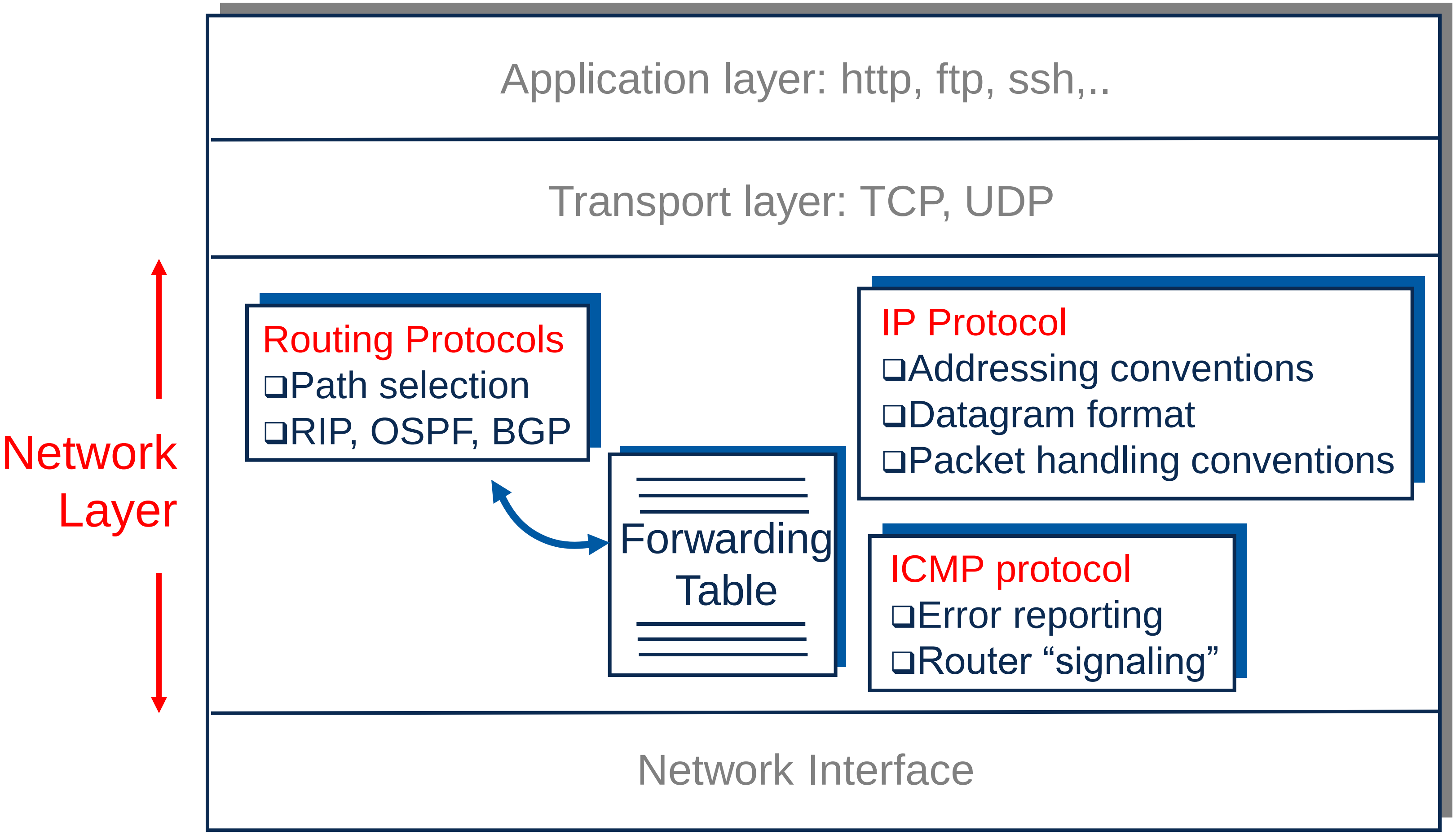
The Internet – Datagrams

- Source sends packets to some receiver
- Connectionless: no call setup at network layer
- Routers: no notion of the end-to-end connections
- Packets forwarded using destination host address at each hop



The Internet Network Layer

Host & router network layer functions:



Application layer: http, ftp, ssh,..

Transport layer: TCP, UDP

Network Layer

Routing Protocols
- ❑Path selection
- ❑RIP, OSPF, BGP

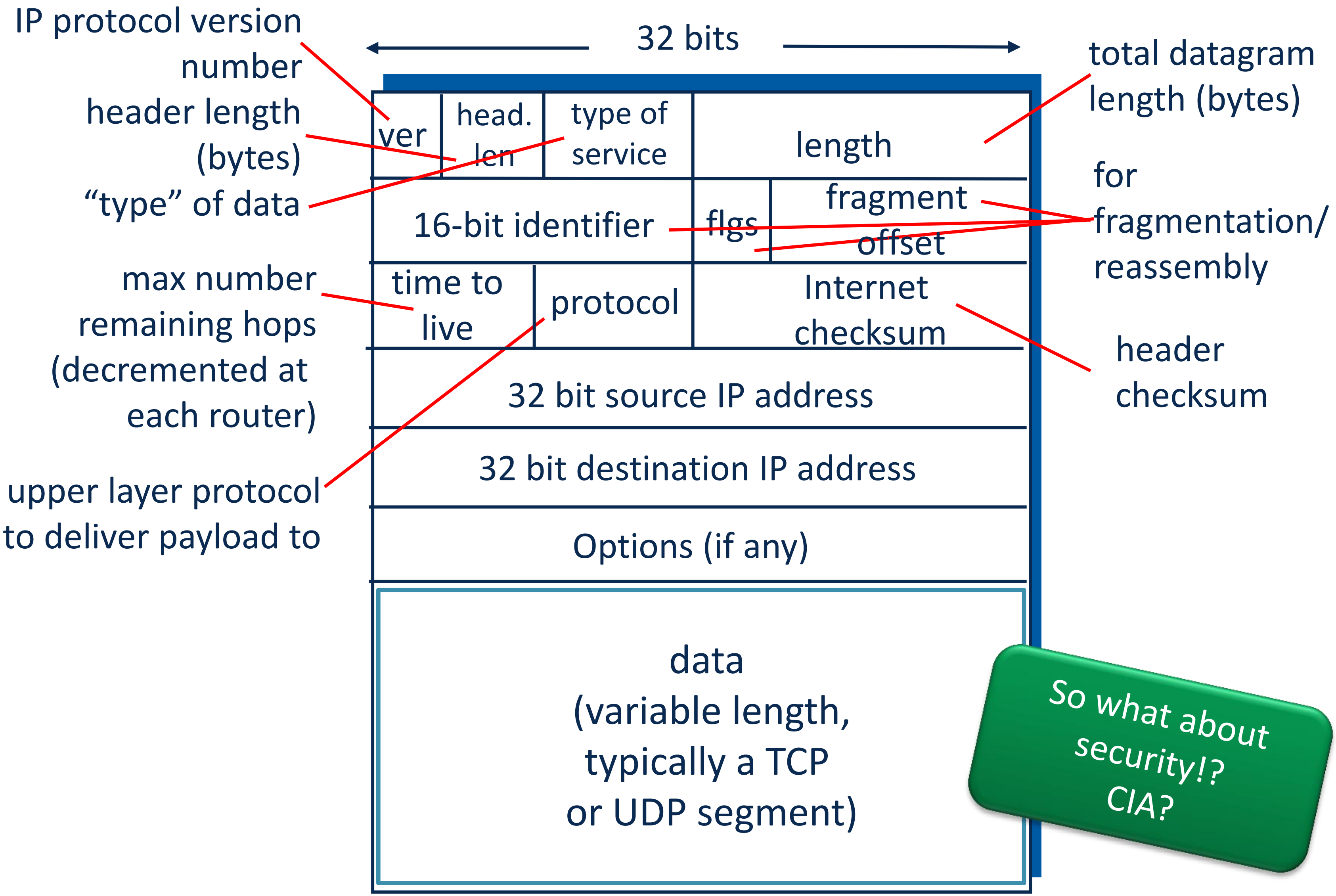
Forwarding Table

IP Protocol
- ❑Addressing conventions
- ❑Datagram format
- ❑Packet handling conventions

ICMP protocol
- ❑Error reporting
- ❑Router “signaling”

Network Interface

IP Packet Format



32 bits

ver | head. len | type of service | length

16-bit identifier | flgs | fragment offset

time to live | protocol | Internet checksum

32 bit source IP address

32 bit destination IP address

Options (if any)

data (variable length, typically a TCP or UDP segment)

IP protocol version number

header length (bytes)

“type” of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to

total datagram length (bytes)

for fragmentation/ reassembly

header checksum

So what about security!? CIA?

IP Packet Format (2)

Version: the IP version number (currently still 4 even though 6 exists)

IHL: IP Header Length in 32-bit words

Type of Service: contains priority information, rarely used

Total Length: the total length of the datagram in bytes (incl. header)

Identification: when an IP packet is segmented into multiple fragments, each fragment is given the same identification; this field is used to reassemble fragments

Flags:

- *DF:* Don’t Fragment
- *MF:* More Fragments; when a packet is fragmented, all fragments except the last one have this bit set

Fragment Offset: the fragment’s position within the original packet (specified in units of 8 octets)

Time to Live: hop count, decremented each time the packet reaches a new router; when hop count = 0, packet is discarded

IP Packet Format (3)

Protocol: identifies which transport layer protocol is being used for this packet (most of the time: either TCP or UDP)

Header Checksum: allows to verify the contents of the IP header

Source and Destination Addresses: uniquely identify sender and receiver of the packet

Options: up to 40 bytes in length; used to extend functionality of IP (examples: source routing, record route)

IP addresses:

- 32 bits long (4 bytes)
- Each byte is written in decimal in MSB order, separated by decimals (example: 128.195.1.80)
- 0.0.0.0 (lowest) to 255.255.255.255 (highest)
- Address Classes: Class A, B, C, D, E, Loopback, Broadcast

Security Problems of the Internet Protocol

IP does ***not*** (cannot) provide:

- Data origin authentication / data integrity:
  - The packet has actually been sent by the “source”
  - The payload has been unaltered
  - The receiving is infact the intended destination

- Confidentiality:
  - The payload per-se is world-readable

=> End-to-End security requires additional measures

Security Objectives of IPsec

Data origin authentication / connectionless data integrity:

- Altered and ***forged*** source/destination shall be detected by receiver
- Integrity of the datagram
- Replay protection: replay of recorded IP packet shall be detected by receiver

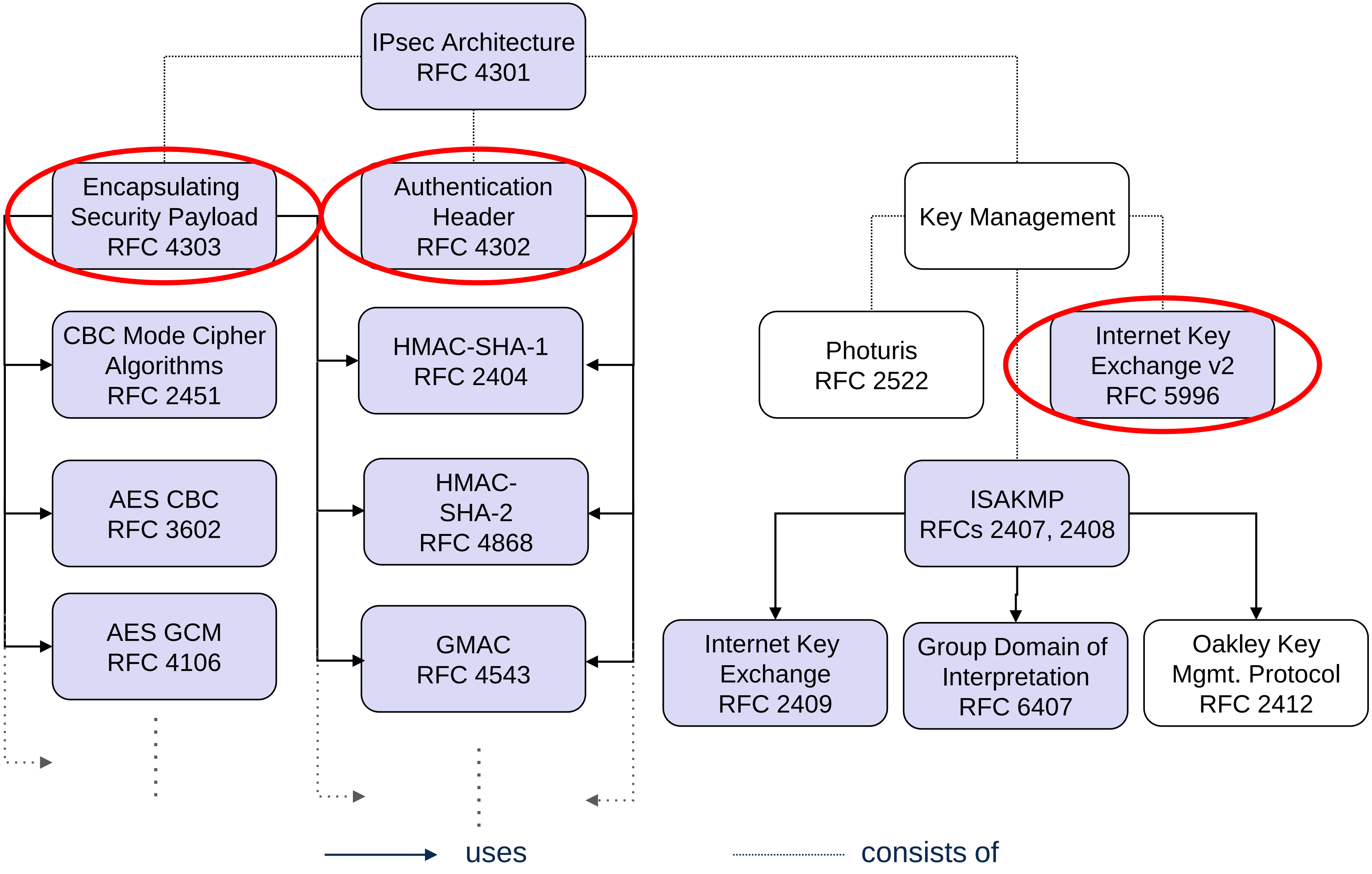
Confidentiality:

- Eavesdropping on the content of IP datagrams is prevented
- Limited traffic flow confidentiality

*Determined by a **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



IPsec Architecture RFC 4301
Encapsulating Security Payload RFC 4303
Authentication Header RFC 4302
Key Management
CBC Mode Cipher Algorithms RFC 2451
HMAC-SHA-1 RFC 2404
Photuris RFC 2522
Internet Key Exchange v2 RFC 5996
AES CBC RFC 3602
HMAC-SHA-2 RFC 4868
ISAKMP RFCs 2407, 2408
AES GCM RFC 4106
GMAC RFC 4543
Internet Key Exchange RFC 2409
Group Domain of Interpretation RFC 6407
Oakley Key Mgmt. Protocol RFC 2412
uses
consists of

Basic Parts of IPsec

RFC 4301 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
- Key Management Procedures:
  - IKE & IKEv2

Crypto in IPsec

RFC 4301 also defines cryptographic primitives with AH and ESP:

- Encryption: 3DES-CBC, AES & other CBC mode cipher algorithms, AES counter mode
- Integrity: HMAC-MD5, HMAC-SHA-1, HMAC-SHA-2, HMAC-RIPEMD-160, AES-GMAC, AES-CMAC, AES-XCBC…
- Authenticated encryption: GCM and `Counter with CBC-MAC' (CCM), both defined for AES

(Hint: check this, when/should you need it)

IPsec Secure "Links": Security Associations

Security Associations (SA) are the basic notion of secure links

- AH / ESP provide the security services to SA

SA are identified by triple:

- security parameter index (SPI)
- IP destination address
- security protocol identifier (AH / ESP)

SA are specified uni-directional

- => Two SA needed for bi-directional communication

Two conceptual databases are associated with SAs:

- The ***security policy database*** (SPD)
- The ***security association database*** (SADB)

Protocol Modes: Transport or Tunnel

IPsec specifies two different protocol modes:

- Transport mode just adds a security specific header (+ eventual trailer):

IP header	IPsec header	protected data

- Tunnel mode encapsulates IP packets:

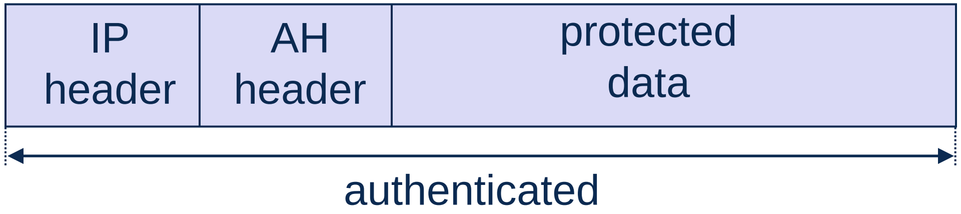
IP header	IPsec header	IP header	protected data

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

IPsec Security Services (AH/ESP)

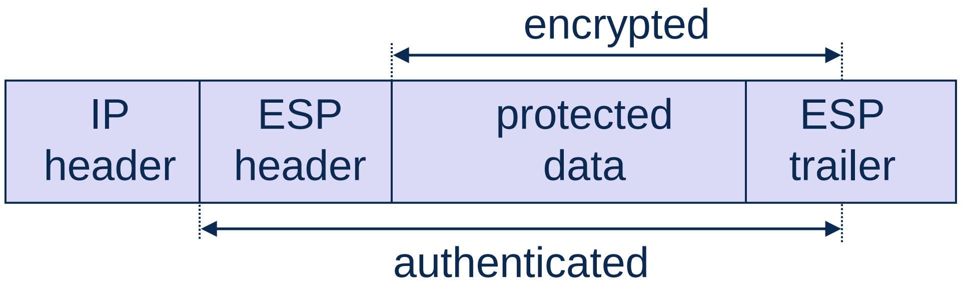
The authentication header (AH):

- Goal: data origin authentication and replay protection
- AH inserts header between the IP header and the data to be protected



The encapsulating security payload (ESP):

- Goals: data origin authentication, confidentiality, and replay protection
- ESP inserts header and a trailer encapsulating the data to be protected



Managing IPsec: ISAKMP/IKE

Setup of security associations is realized with:

- Internet **S**ecurity **A**ssociation **K**ey **M**anagement **P**rotocol (ISAKMP):
  - Generic framework for key authentication, key exchange, and negotiation of security association parameters [RFC2408]
  - No authentication protocol, but:
    - Packet formats
    - Retransmission timers
    - Message construction requirements
  - Use of ISAKMP for IPsec is further detailed in [RFC2407]

- Internet **K**ey **E**xchange (IKE):
  - Authentication and key exchange protocol [RFC2409]
  - Conforms to ISAKMP, may be used for different applications
  - Setup of IPsec SAs between two entities is realized in two phases:
    - Establishment of an IKE SA (defines how to setup IPsec SAs)
    - Setup of IPsec SAs

“Integrity”: The Authentication Header (AH)

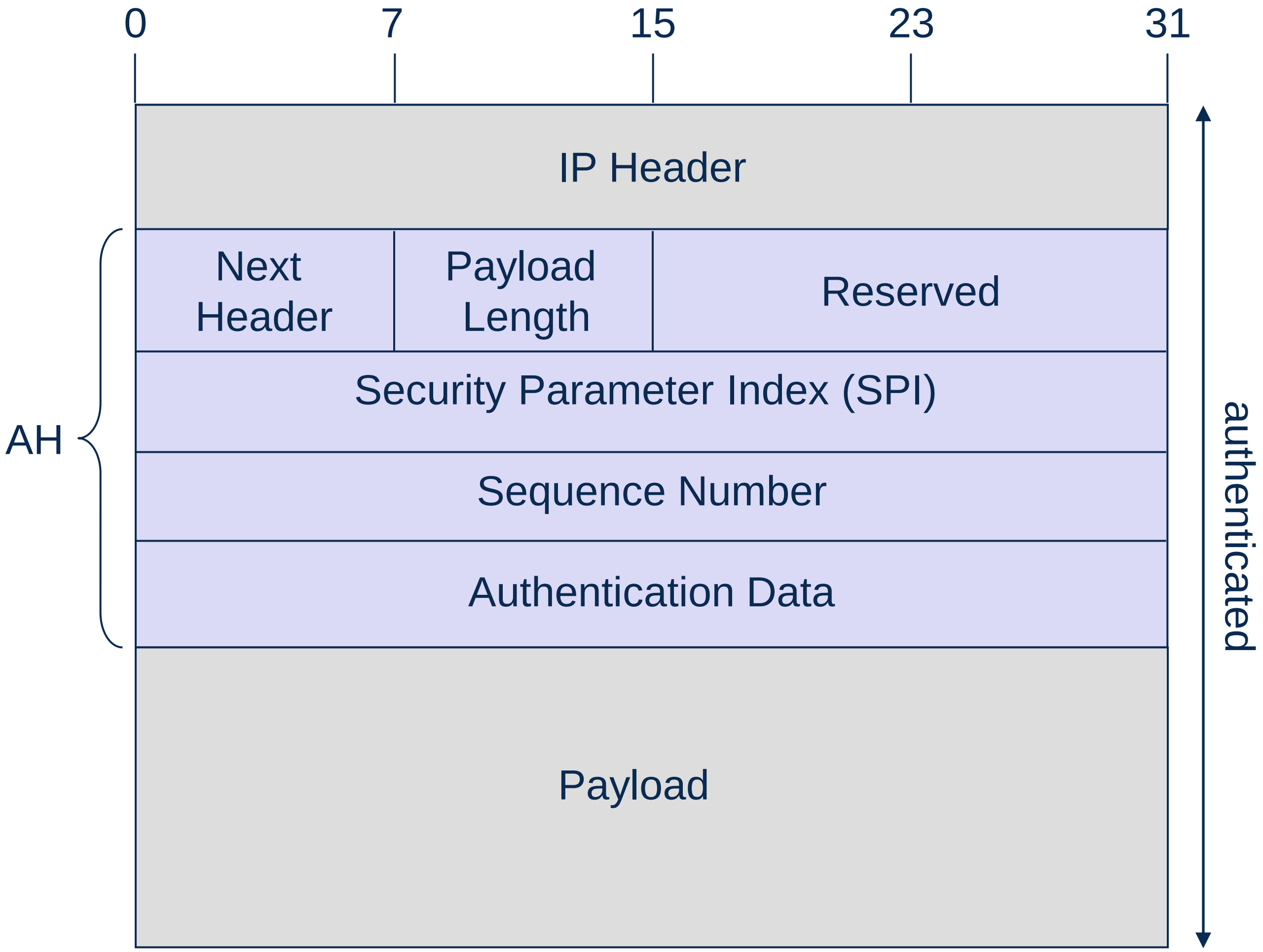
Goals of the Authentication Header (protocol)

- Data origin authentication
- Replay protection

AH spec is divided into two parts:

- The definition of the base protocol
  - Definition of the header format
  - Basic protocol processing
  - Tunnel and transport mode operation
- The use of specific cryptographic algorithms with AH:
  - Authentication: HMAC-MD5-96, HMAC-SHA1-96, HMAC-SHA2, …

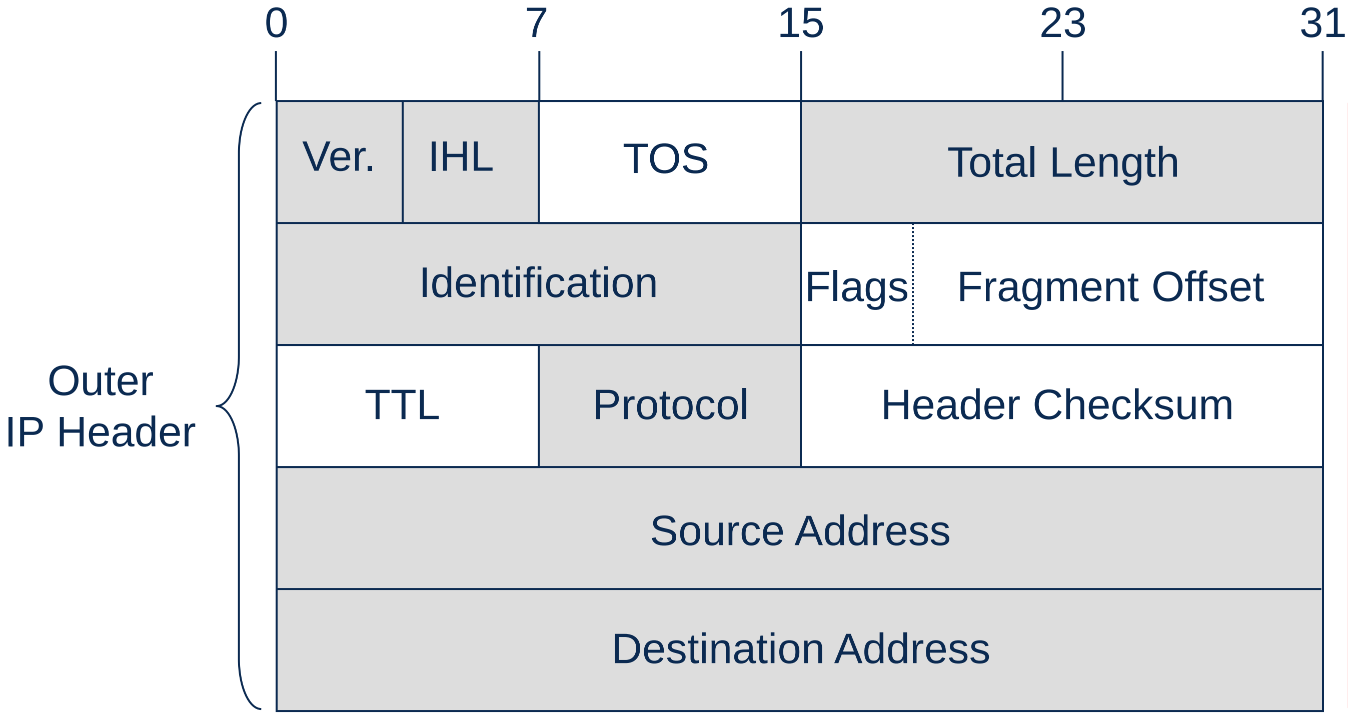
The Authentication Header



Protecting the IP Header

AH has to protect the “outer” IP header

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

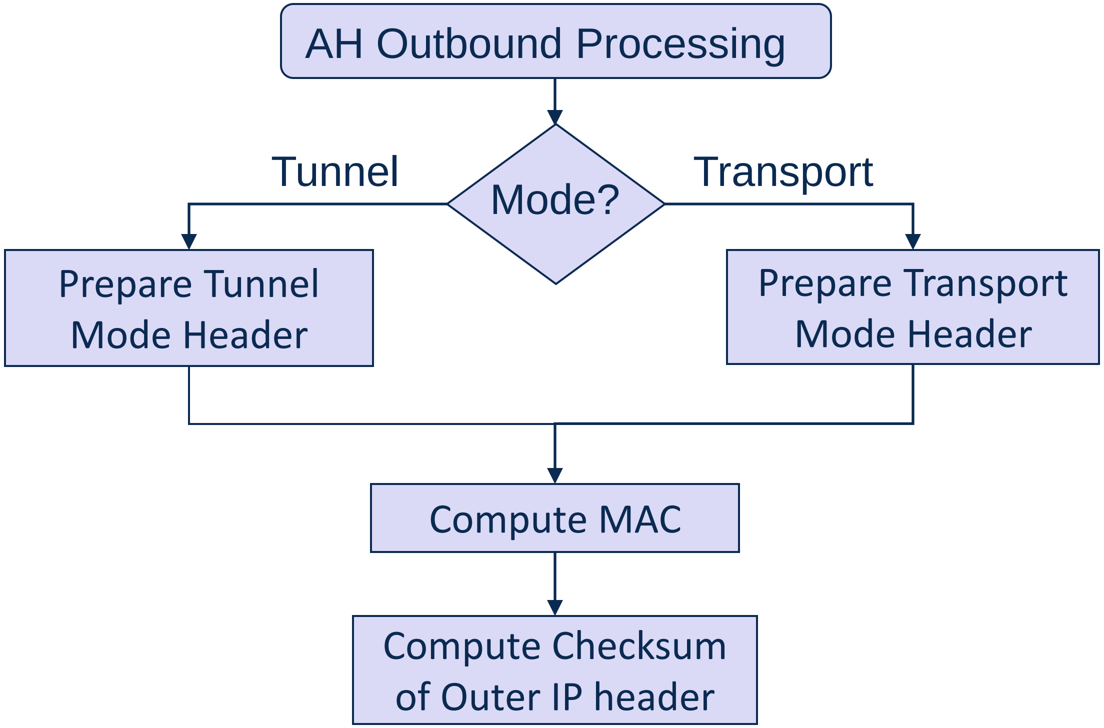


Post lecture update:
- Total length is not „immutable“, but recovered: Reassembly happens before MAC validation
- Identification flag is set by the sender (at random/counter)
- Some TOS bits are set on path (e.g., DSCP)

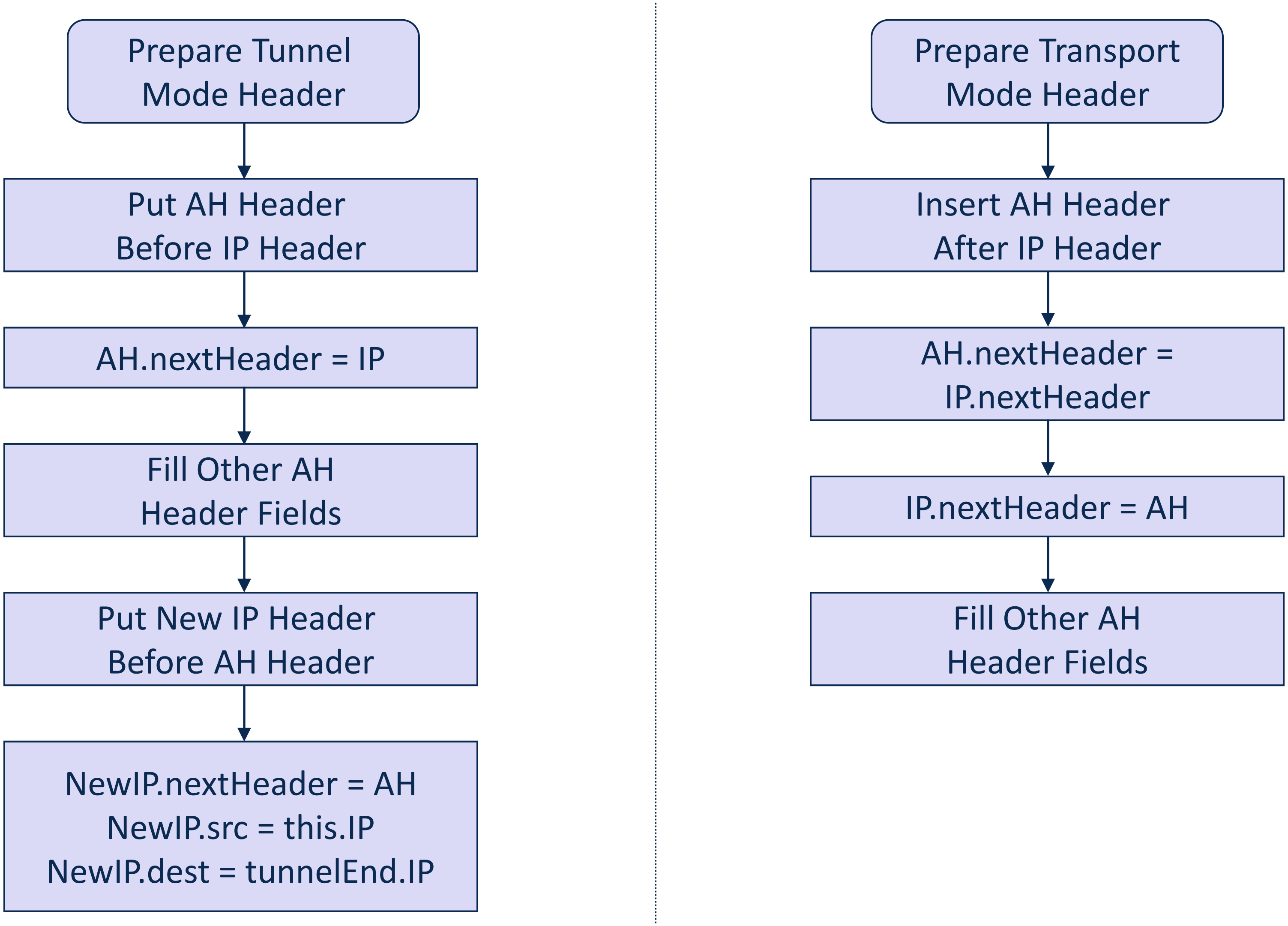
Some fields cannot be protected E2E, they are subject to change

- These fields are assumed being zero when computing the MAC

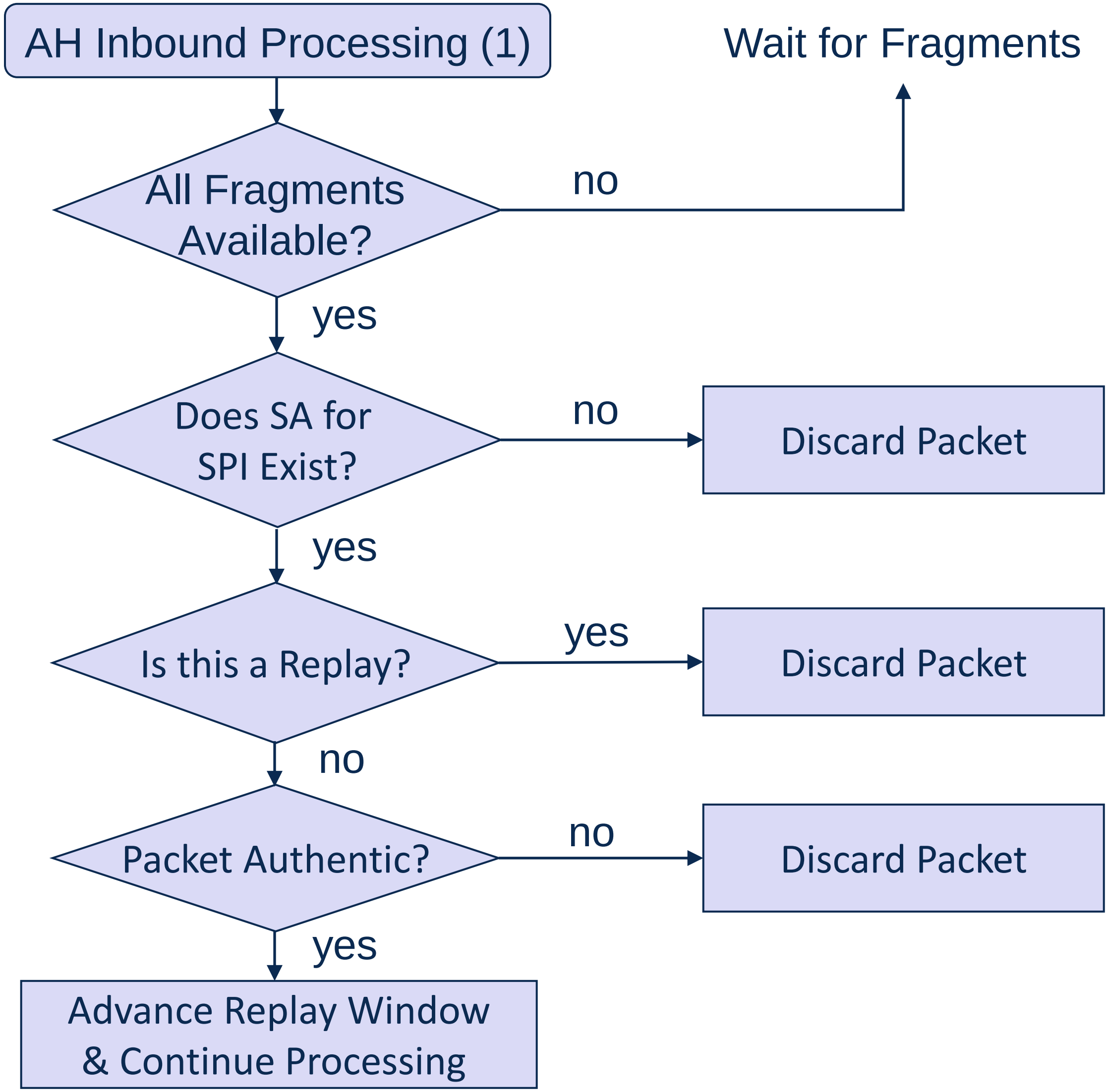
Outbound AH


AH Outbound Processing
Tunnel
Mode?
Transport
Prepare Tunnel Mode Header
Prepare Transport Mode Header
Compute MAC
Compute Checksum of Outer IP header

The Authentication Header (5)

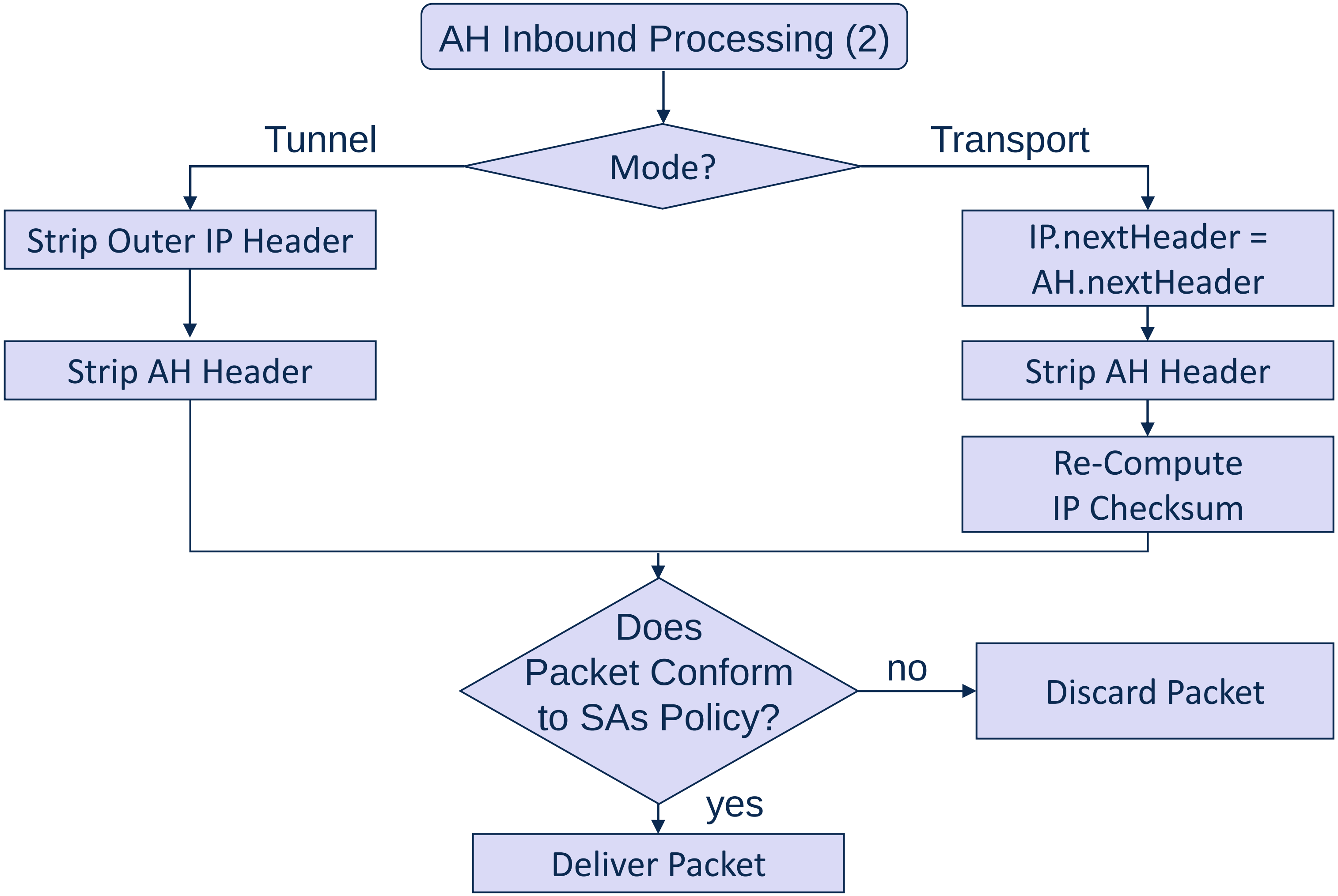

Prepare Tunnel Mode Header
Put AH Header Before IP Header
AH.nextHeader = IP
Fill Other AH Header Fields
Put New IP Header Before AH Header
NewIP.nextHeader = AH
NewIP.src = this.IP
NewIP.dest = tunnelEnd.IP
Prepare Transport Mode Header
Insert AH Header After IP Header
AH.nextHeader = IP.nextHeader
IP.nextHeader = AH
Fill Other AH Header Fields

Inbound AH


AH Inbound Processing (1)
Wait for Fragments
All Fragments Available?
no
yes
Does SA for SPI Exist?
no
Discard Packet
yes
Is this a Replay?
yes
Discard Packet
no
Packet Authentic?
no
Discard Packet
yes
Advance Replay Window & Continue Processing

TECHNISCHE UNIVERSITÄT DRESDEN

The Authentication Header (7)



The Encapsulating Security Payload

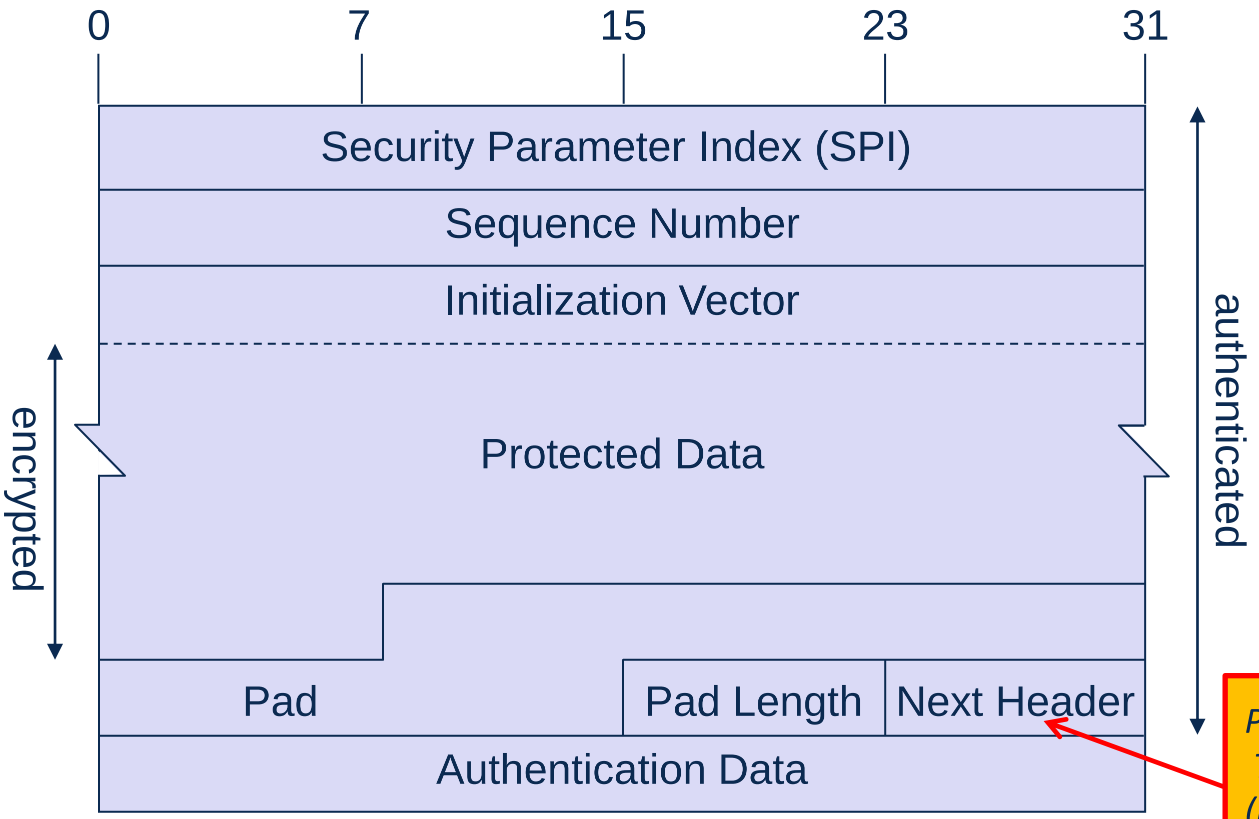
More comprehensive protection: ESP offers (1 and/or 2 **and** 3)

1. Confidentiality (encryption of packet or only payload)
2. Data origin authentication (MACs)
3. Replay protection

The ESP spec is divided into two parts:

- The definition of the base protocol
  - Definition of the header and trailer format
  - Basic protocol processing
  - Tunnel and transport mode operation
- The use of specific cryptographic algorithms with ESP:
  - Encryption: 3DES-CBC, AES-CBC, AES counter mode, use of other ciphers in CBC mode
  - Authentication: HMAC-MD5-96, HMAC-SHA-96,...

The ESP Header



Post lecture update: The „next header“ (IP.proto) was missing, here it is! ☺

The ESP header immediately follows the IP or AH header

ESP Header Fields

SPI field indicates the SA to be used for this packet:

- The SPI value is determined by receiver during SA negotiation as receiver has to process the packet

Sequence number for replay protection

IV for initialization vector, if crypto algorithm requires it (transmitted in the clear in every packet)

Pad field serves to ensure:

- padding of the payload up to the required block length of the cipher in use

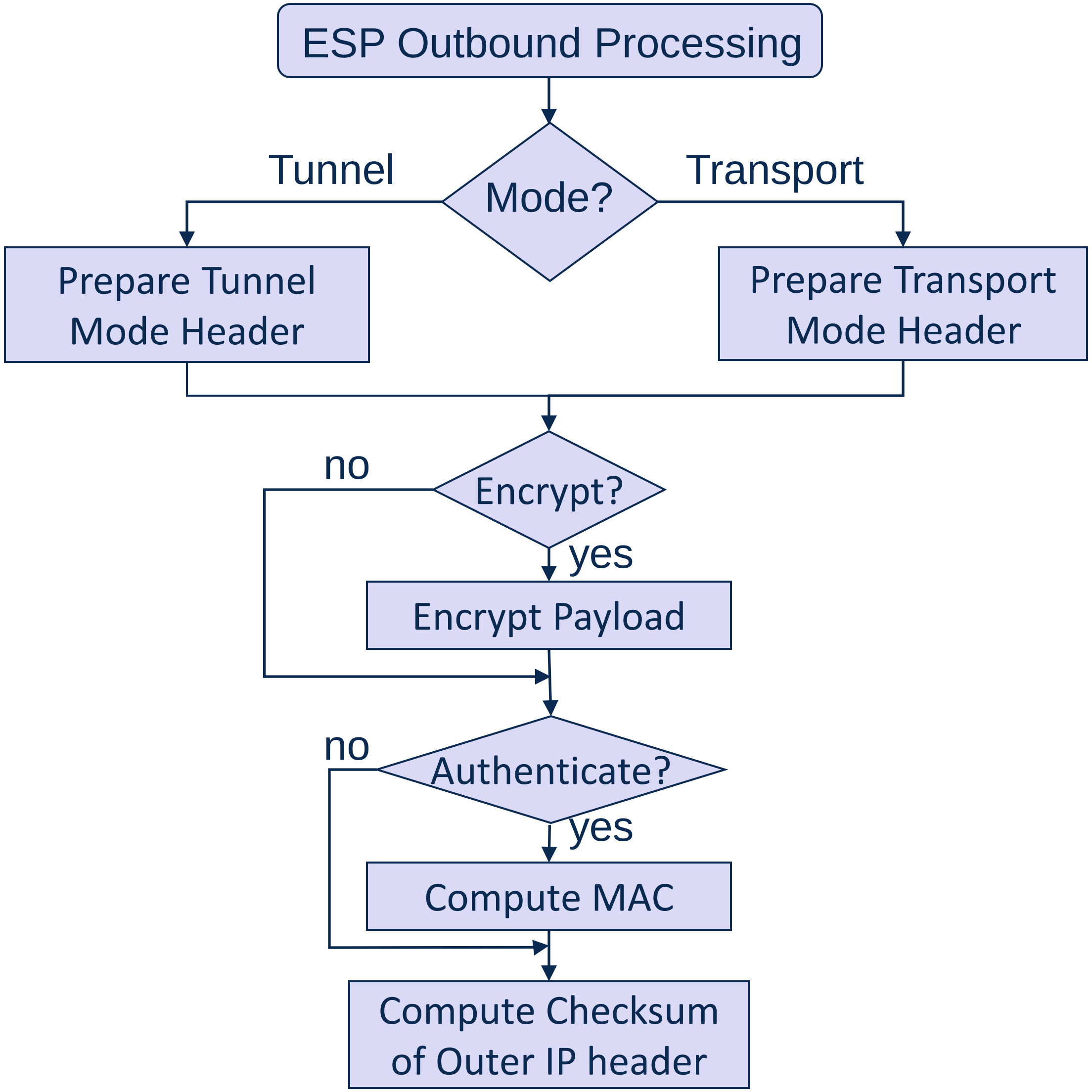
Pad length indicates the amount of padding bytes added

next-header field of the ESP header indicates the encapsulated payload:

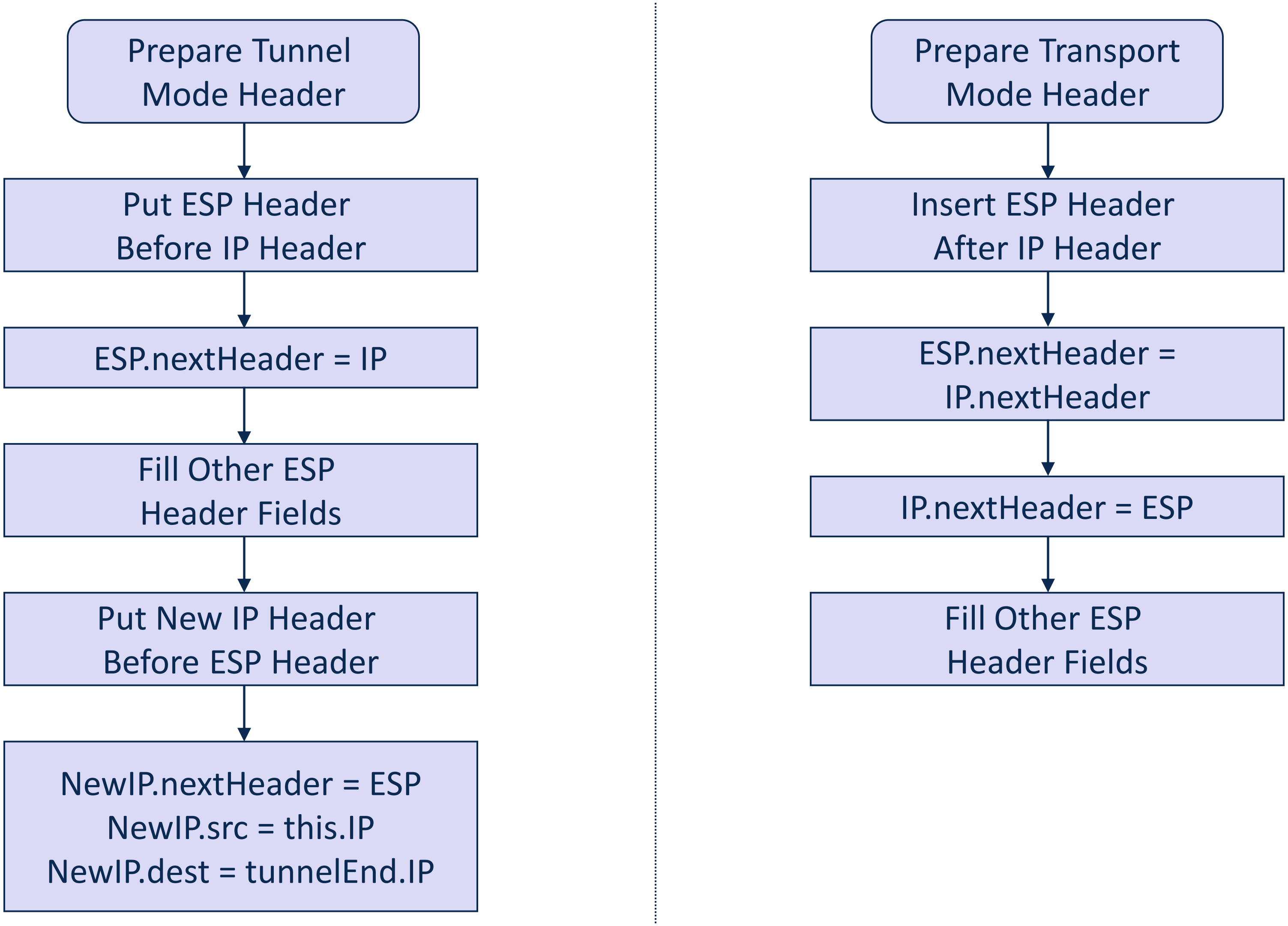
- In case of tunnel mode: IP
- In case of transport mode: any higher-layer protocol as TCP, UDP, ...

Optional authentication-data field contains a MAC, if present

Outbound ESP



The Encapsulating Security Payload (5)



Prepare Tunnel Mode Header

Put ESP Header Before IP Header

ESP.nextHeader = IP

Fill Other ESP Header Fields

Put New IP Header Before ESP Header

NewIP.nextHeader = ESP
NewIP.src = this.IP
NewIP.dest = tunnelEnd.IP

Prepare Transport Mode Header

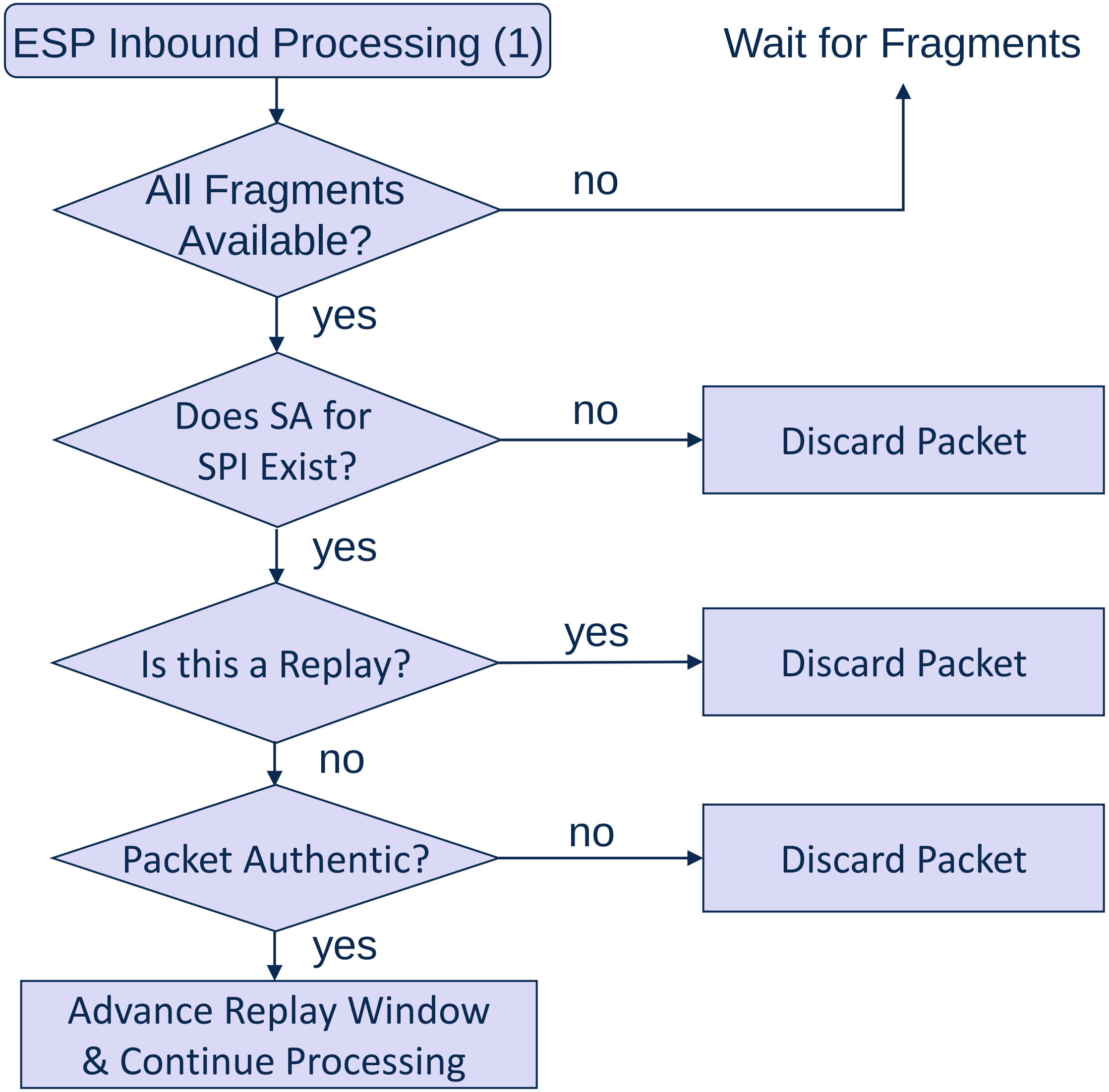
Insert ESP Header After IP Header

ESP.nextHeader = IP.nextHeader

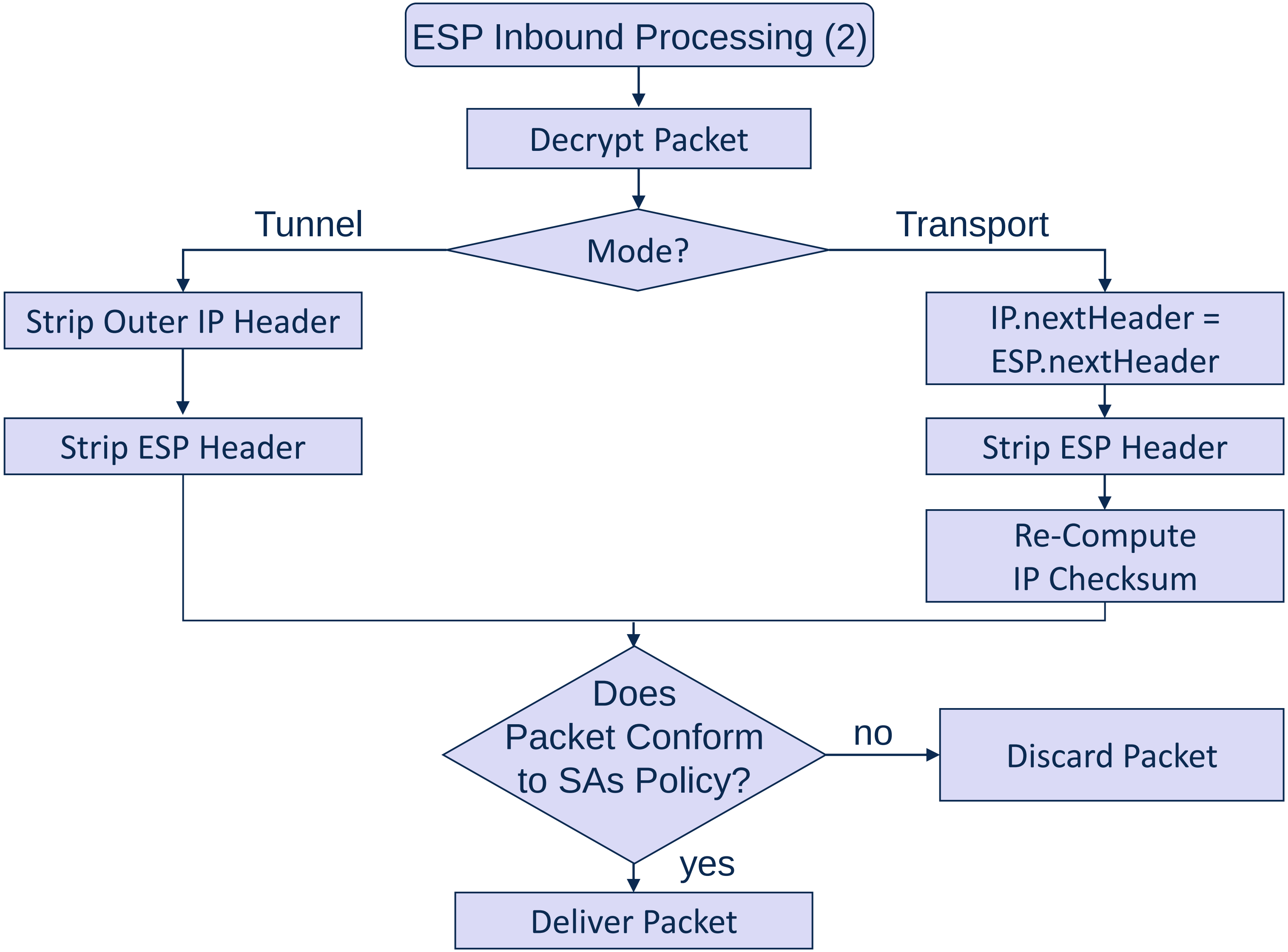
IP.nextHeader = ESP

Fill Other ESP Header Fields

Inbound ESP


ESP Inbound Processing (1)
All Fragments Available?
no
Wait for Fragments
yes
Does SA for SPI Exist?
no
Discard Packet
yes
Is this a Replay?
yes
Discard Packet
no
Packet Authentic?
no
Discard Packet
yes
Advance Replay Window & Continue Processing

Inbound ESP (Tunnel vs. Transport)



Establishing Security Associations

Prior to any packet being protected by IPsec, a SA has to be established between the two "cryptographic endpoints" providing the protection

Requires Security Policy Definitions

Specific fields 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
- 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

Establishing SAs

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

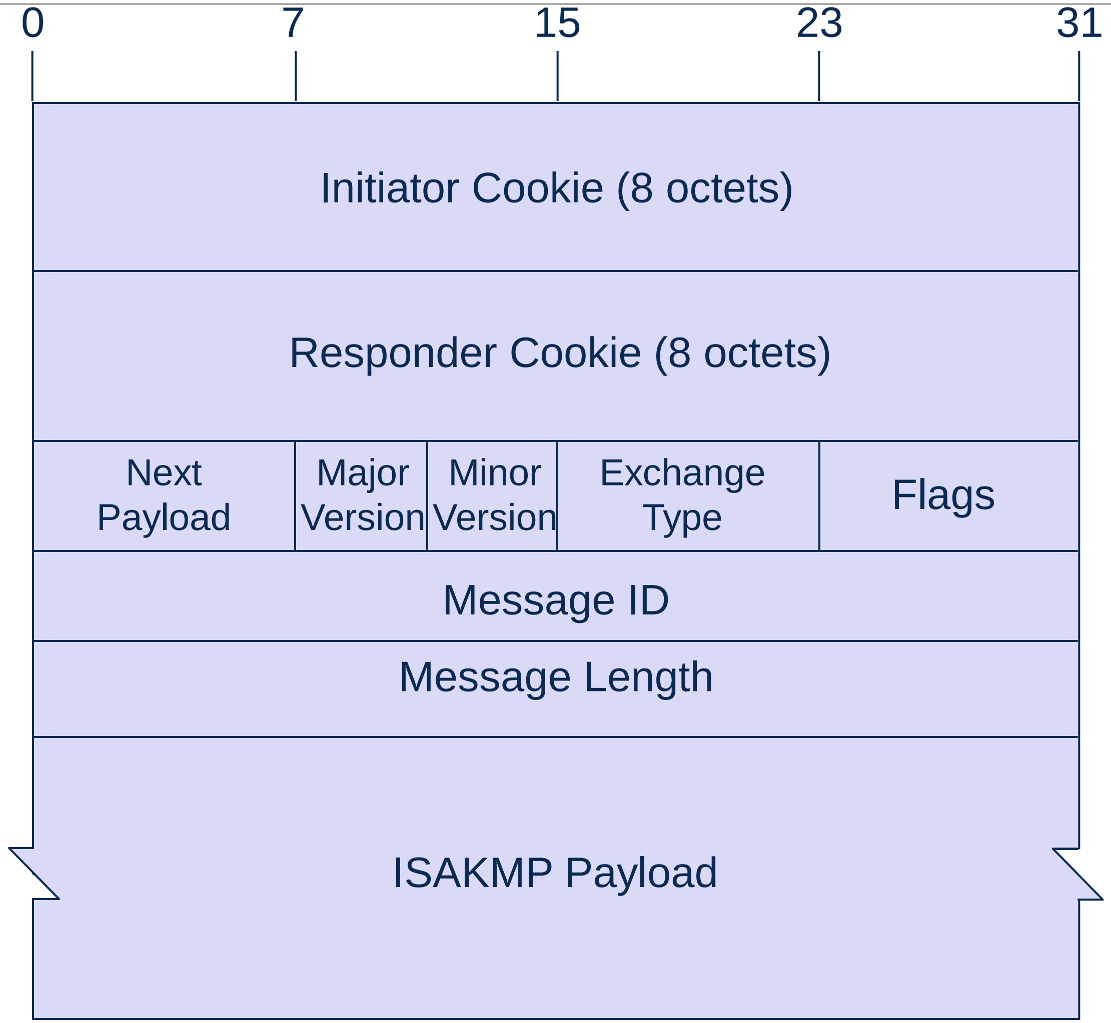
The ISAKMP base protocol is a generic protocol, that can be used for various purposes:

- 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 [RFC6407]
  - 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 – Basic Message Format (1)



0 7 15 23 31

Initiator Cookie (8 octets)

Responder Cookie (8 octets)

Next Payload	Major Version	Minor Version	Exchange Type	Flags

Message ID

Message Length

ISAKMP Payload

ISAKMP – Basic Message Format (2)

Initiator & responder cookie:

- Identify an ISAKMP exchange, or security association, respectively
- Also serve as a limited protection against denial of service attacks (explained below)

Next payload: specifies which ISAKMP payload type is the first payload of the message

Major & minor version: identify the version of the ISAKMP protocol

Exchange type:

- Indicates the type of exchange being used
- There are five pre-defined generic exchange types, further types can be defined per DOI

Flags:

- Encrypt: if set to one, then the payload following the header is encrypted
- Commit: used for key synchronization purposes
- Authenticate only: if set to one, only data origin authentication protection is applied to the ISAKMP payload and no encryption is performed

ISAKMP – Basic Message Format (3)

Message ID:

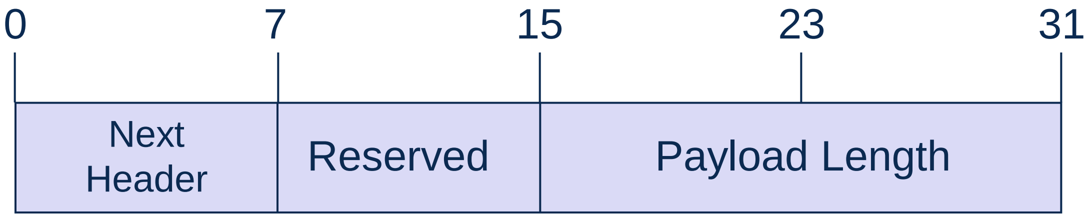
- Used to identify messages belonging to different exchanges

Message Length:

- Total length of the message (header + payload)

Payload:

- The payload of one ISAKMP message can, in fact, contain multiple “chained” payloads
- The payload type of the first payload in the message is indicated in the next payload field of the ISAKMP header
- All ISAKMP payloads have a common payload header:



0 7 15 23 31

Next Header | Reserved | Payload Length

Next Header: the payload type of the next payload in the message

Payload Length: total length of current payload (including this header)

Internet Key Exchange (IKE)

Internet Key Exchange specifies protocol to negotiate IPsec SAs
(ISAKMP defines basic data formats/procedures to negotiate arbitrary 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

… IKEv1 is considered slightly overloaded…

IKEv2

Consolidation of several IKEv1 RFCs (and several extensions)

- Makes things easier for developers & testers
- Clarifies several unspecific points

Simplifications

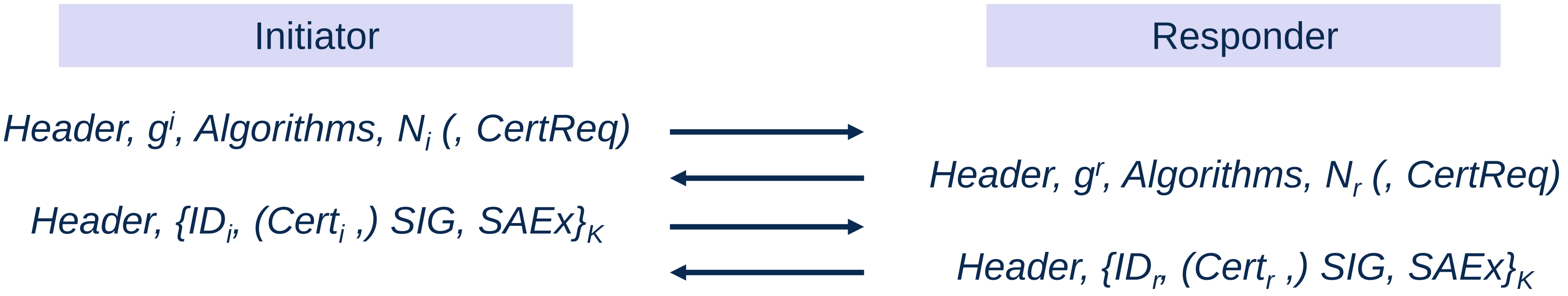
- Number of different key exchanges reduced to one
- Encryption like in ESP
- Simple Request/Response mechanism

Decrease Latency

Negotiation of traffic selectors

Graceful changes to allow existing IKEv1 software to be upgraded

IKEv2 – Key Exchange Procedure



Initiator

Responder

Header, g^i, Algorithms, N_i (, CertReq) →

← *Header, g^r, Algorithms, N_r (, CertReq)*

Header, $\{ID_i$, (Cert$_i$,) SIG, SAEx$\}_K$ →

← *Header, $\{ID_r$, (Cert$_r$,) SIG, SAEx$\}_K$*

where: K key derived by PRF(PRF(Ni || Nr, gir), Ni || Nr || SPIi || SPIr)

PRF “some” pseudo-random function – usually an HMAC

SIG asymmetric signature or MAC over the first two messages

SAEx a piggybacked “Quick-Mode-Exchange”

Only a single exchange type

Four messages exchanged (= 2 * RTT)

Initiator triggers all retransmissions

IKEv2 – Properties of the Key Exchange Procedure

First SA exchange is piggybacked

- Lower latency, as it saves one RTT

Message 4 was discussed to be piggybacked to message 2, but

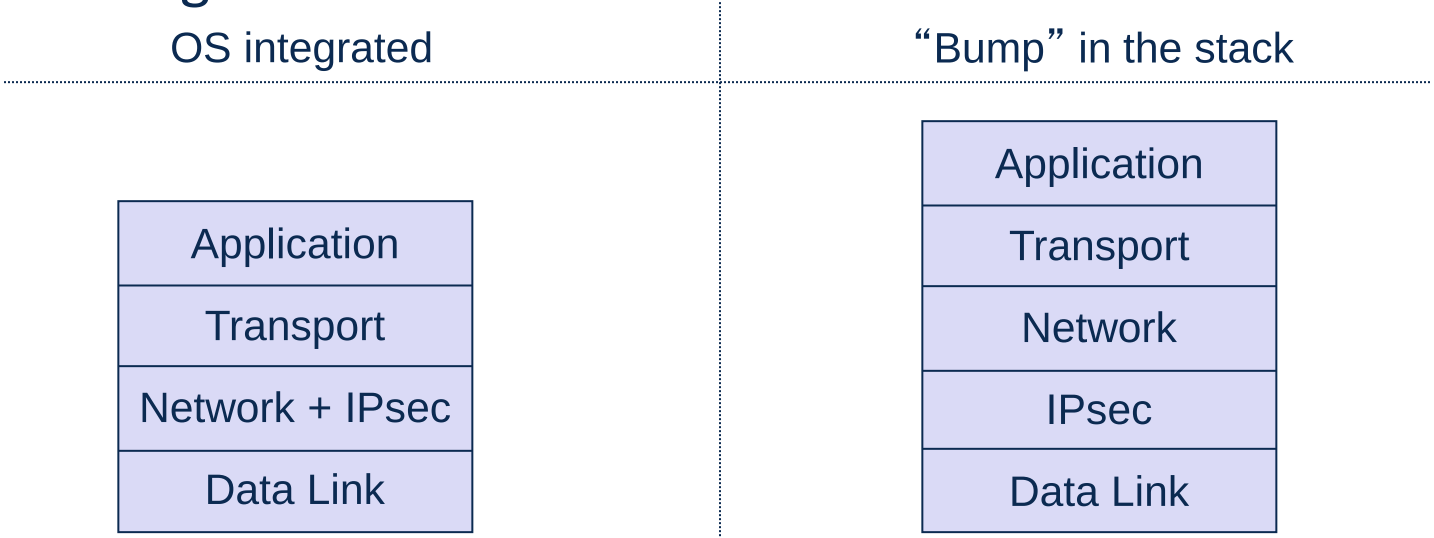
- Message 3 verifies that initiator received message 2 (SPI ~ Cookie)
  - Serves as a DoS protection if computational intensive tasks are performed afterwards
- Identity of responder only disclosed after verification of initiator
  - Protects from scanning for a party with a specific ID
- Initiator would not know when it is safe to send data
  - (Packets may be received out of order)
- Would require more complicated retransmission strategy
- Responder cannot decide on a policy for the child SA

IPsec 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:



OS integrated

Application
Transport
Network + IPsec
Data Link

“Bump” in the stack

Application
Transport
Network
IPsec
Data Link

True OS integration is the method of choice,
as it avoids duplication of functionality

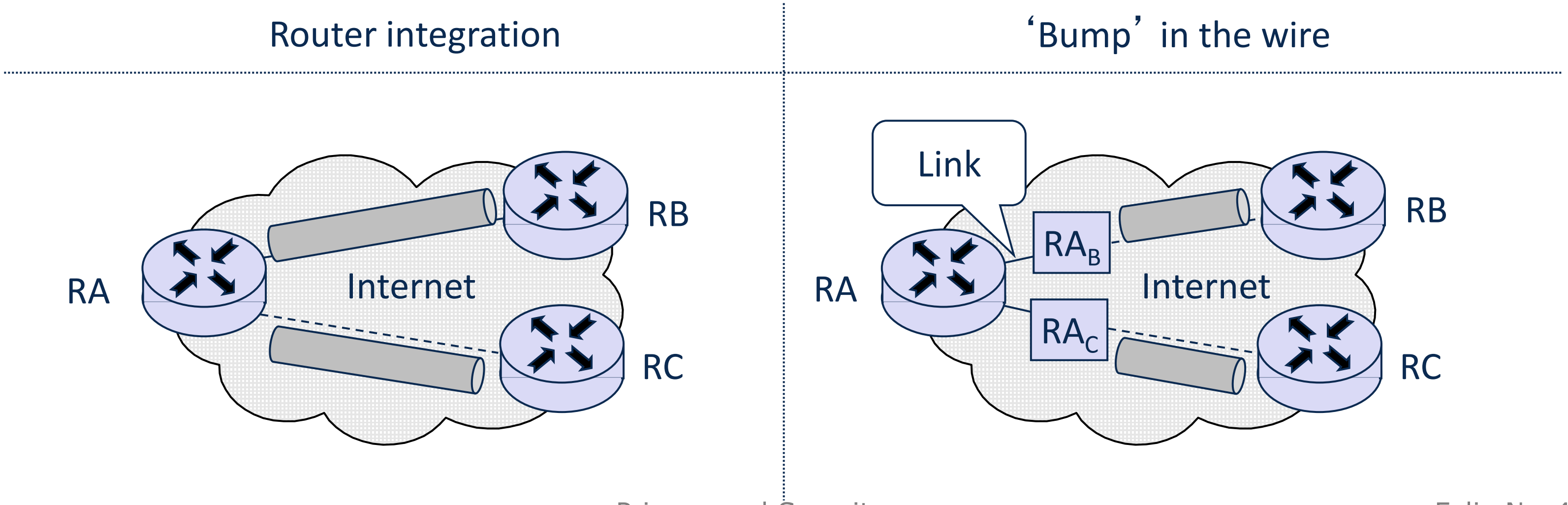
If the OS can not be modified, IPsec
is inserted above the data link driver

IPsec 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

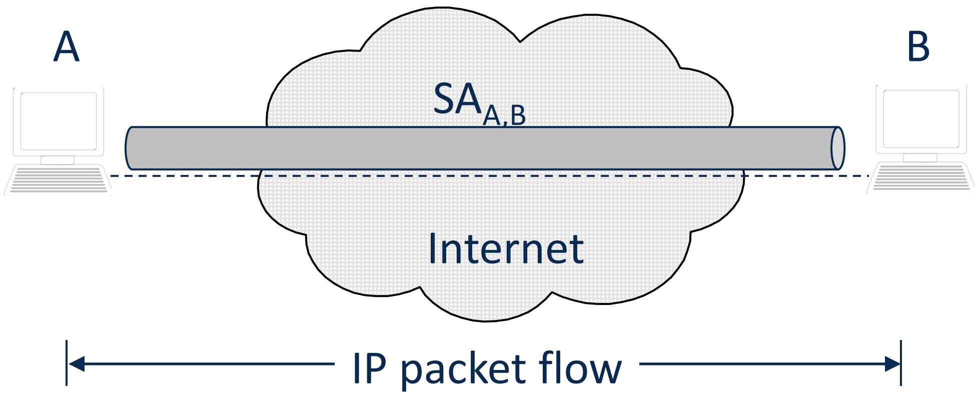
Two main implementation alternatives:



When to use Transport Mode

When endpoints of secure connection are communication endpoints

- Cryptographic endpoints: the entities that generate / process an IPsec header (AH or ESP)
- Communication endpoints: source and destination of an IP packet



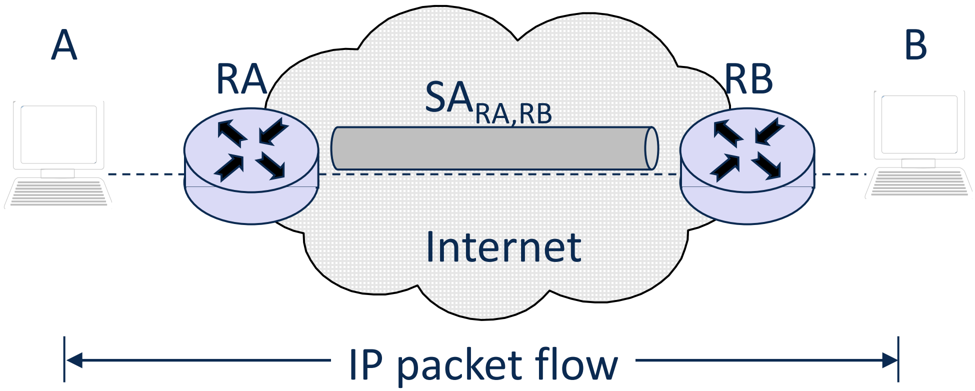
In most cases, communication endpoints are hosts (workstations, servers), but this is not necessarily the case:

- Example: a gateway being managed via SNMP by a workstation

When to use Tunnel Mode

If at least one “cryptographic endpoint” is not a “communication endpoint” of the secured IP packets

- Allows for gateways that protect IP traffic on behalf of other entities



Packet structure

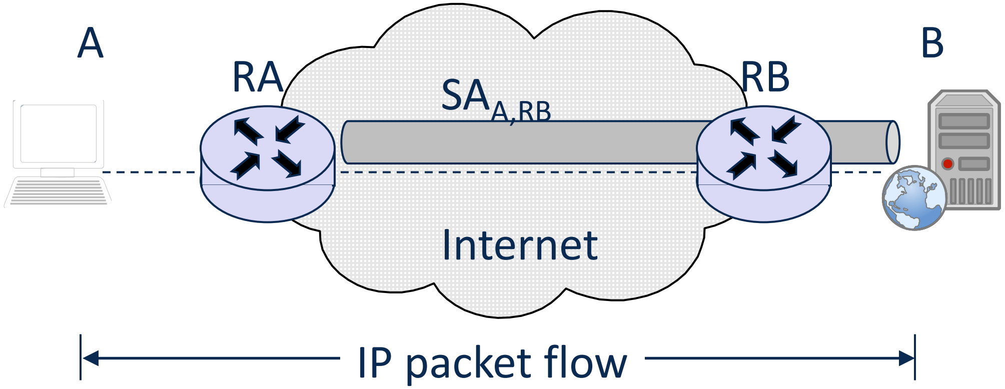
IP Header	IPsec Header	IP Header	Protected Data

Src = RA
Dst = RB
		Src = A
		Dst = B	

A Special Case for the Tunnel Mode

If 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”)



Packet structure

IP Header	IPsec Header	IP Header	Protected Data

Src = A
Dst = RB
		Src = A
		Dst = B	

Some Issues with IPsec

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

Compression

- Encryption causes noise-like content => no efficient subsequent compression
- *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

Conclusion

IPsec is IETF' s security architecture for the Internet Protocol

IPsec provides the following security services to IP packets:

- Data origin authentication
- Replay protection
- Confidentiality

Two fundamental security protocols have been defined:

- Authentication header (AH)
- Encapsulating security payload (ESP)

SA negotiation and key management is realized with:

- Internet security association key management protocol (ISAKMP)
- Internet key exchange (IKE)

Implementation in either end systems or intermediate systems:

- End system implementation: OS integrated or "bump in the stack"
- Gateway implementation: Router integrated or "bump in the wire"