

Advanced Networking

IPsec Security Architecture for IP

Csaba Kiraly
kiraly@disi.unitn.it

based on slides from Prof. Giuseppe Bianchi

Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ Architecture (much more than a protocol)

⇒ protocols (ESP, AH)

⇒ Management (SAD, SPD)

⇒ Signaling (IKE)

⇒ History (RFC series)

→ VPN

Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ Architecture (much more than a protocol)

⇒ protocols (ESP, AH)

⇒ Management (SAD, SPD)

⇒ Signaling (IKE)

⇒ History (RFC series)

Topics

→ **Overview of security services**

⇒ *Based on ISO OSI security reference model*

→ **How some known protocols map to the ISO OSI model?**

⇒ To layers

⇒ To security model

→ **IPsec**

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ Architecture (much more than a protocol)

⇒ protocols (ESP, AH)

⇒ Management (SAD, SPD)

⇒ Signaling (IKE)

⇒ History (RFC series)

Networking & Security

Security services as defined by ISO

- ⇒ Defined in the same set of standards as the famous ISO OSI 7 layers (ISO 7498-1) (1984)
- ⇒ ISO 7498-2 OSI Basic Reference Model Part 2: Security Architecture (1989)
 - Security **services**: what to do
 - Security **mechanism**: how to achieve it
 - **Mapping** between services and mechanisms
 - Potential **mapping** to 7 layers: where to implement

Further reading: ISO 7498-2 is not free, but you can download free equivalent from ITU as ITU-T X.800

Security Services

Authentication

- ⇒ To know who it is: the process of **proving** identity
 - Mutual: both parties identified
 - One-way: only one side proves identity

Access control

- ⇒ Control access rights to a resource (communication; read/write/delete of data)
 - Good authentication is a pre-condition!

Data confidentiality

- ⇒ protection of data from unauthorized disclosure

Data integrity

- ⇒ Preventing/detecting modification of the data

Non-repudiation

- ⇒ Preventing an individual or entity from denying having performed a particular action
- ⇒ The recipient of data is provided with proof of the origin of data
- ⇒ The sender of data is provided with proof of delivery of data.

Security Mechanisms

Some examples only!

→ Encryption

⇒ symmetric key cryptography

→ knowledge of the encryption key implies knowledge of the decryption key and vice versa;

⇒ asymmetric (or “public”) key cryptography

→ knowledge of the decryption key (public key) does not imply knowledge of the encryption key (private key).

Used in: mainly in confidentiality, but also in authentication

→ Digital signatures

Used in: authentication, data integrity, non-repudiation

→ Authentication exchange

Used in: authentication

Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ ***How some known protocols map to the ISO OSI model?***

⇒ *To layers*

⇒ *To security model*

→ IPsec

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ Architecture (much more than a protocol)

⇒ protocols (ESP, AH)

⇒ Management (SAD, SPD)

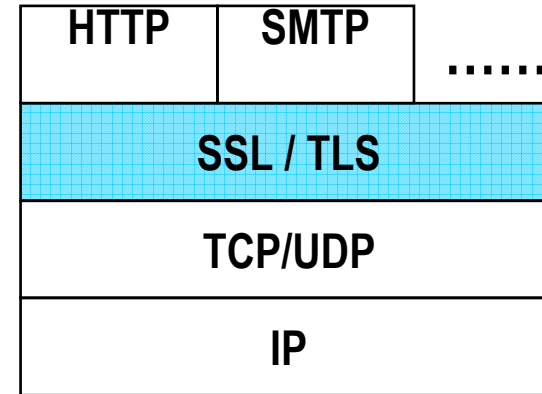
⇒ Signaling (IKE)

⇒ History (RFC series)

Protocols you might use (or know) layer 3 and above

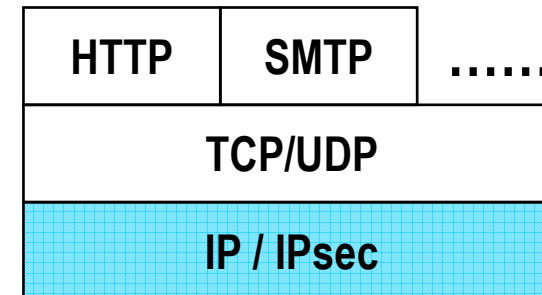
→ SSL/TLS over TCP

- ⇒ Layer: 4+ (above TCP)
- ⇒ Security services:
 - Authentication (mutual / one-way)
 - Data confidentiality
 - Data integrity



→ IPsec

- ⇒ Layer: 3
- ⇒ Security services:
 - Authentication (mutual)
 - Access control
 - Data confidentiality
 - Data integrity



Protocols you might use (or know)

layer 1,2

Wired

→ physical protection of the wire!

Wireless

→ **WEP (Wired Equivalent Privacy)**

⇒ Layer: 2

⇒ Security services:

→ Authentication (weak)

→ Data confidentiality (weak)

→ Data integrity (weak)

→ **802.1x (port-based Network Access Control)**

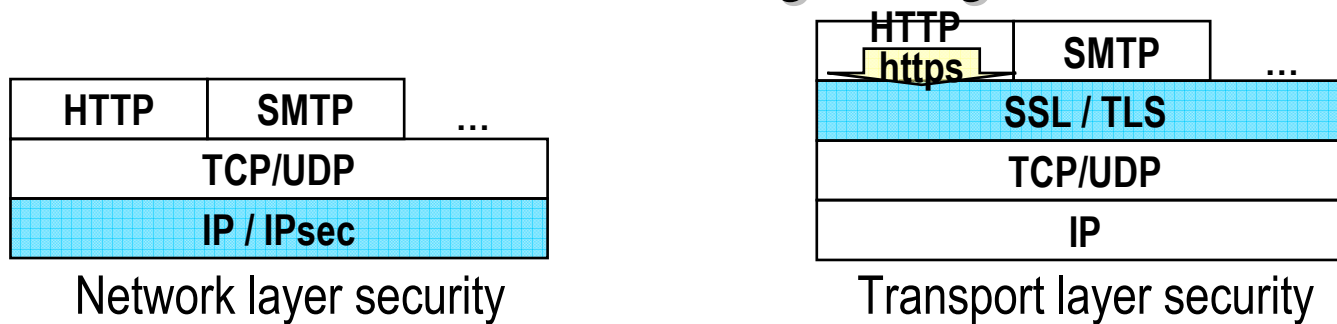
⇒ “port” is the LAN port (not the TCP/UDP one)

⇒ Layer: 2

⇒ Security services:

→ Access control

IPsec: why layer 4



☺ **TLS is transparent for routers**

- ⇒ It operates over TCP ... well above IP
 - IP header is the same => IP routing is not affected
 - The TCP stream is encrypted, but a router should not look at that
 - There are some port numbers typically used with TLS, but this is not mandatory (443:https, 993:imaps)

☺ **TLS is implemented above Layer 4, in the application**

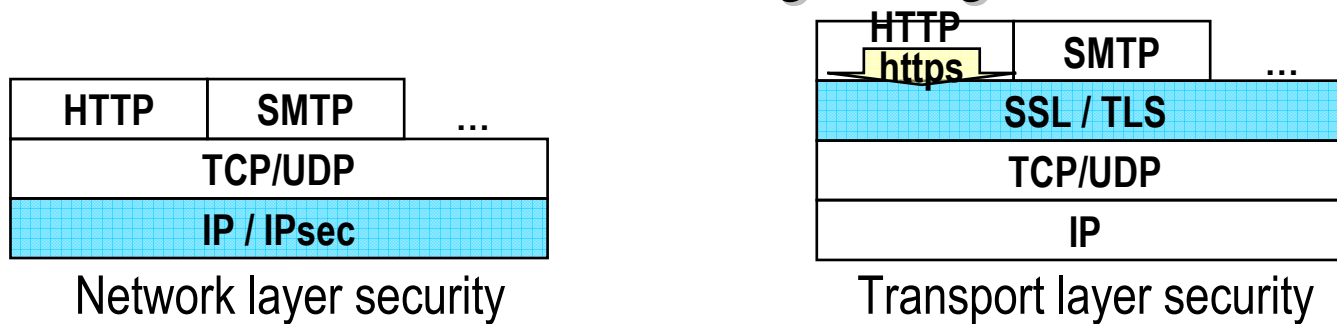
- ⇒ No need to change the OS => fast deployment
 - Early versions (1994) came as part of Netscape browser
- ⇒ Easy to come up with new modified versions
 - Dangerous for security protocols!

☹ **TLS relies on TCP's reliable stream delivery service**

- ⇒ What about security for applications using UDP?
- ⇒ What about other protocols over IP?

☹ **Each application should be changed**

IPsec: why layer 3



☺ **IPsec is transparent for routers**

- ⇒ IPsec operates within (as an upper sub-layer of) layer 3
 - Uses extension header mechanism: seen by routers as “next protocol” in IP header
 - packets are routed just as plain IP packets

☺ **Applications/terminals unaware of IPsec**

- ⇒ IPsec can protect all protocols that rely on IP (but it is hard to differentiate between applications, only TCP/UDP port based differentiation)
- ⇒ It can protect the traffic of whole subnets (tunnel mode, VPN)

☹ **Works only if IP routing works**

- ⇒ Has difficulties passing NAT/NAPT
- ⇒ Not suitable if application level (e.g. HTTP) proxies are used

☹ **Should be implemented in layer 3**

- ⇒ In the kernel of the operating system, not in the application

Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ *Introduction (operation modes, relation to IPv6, extension headers)*

⇒ *Architecture (much more than a protocol)*

⇒ *protocols (ESP, AH)*

⇒ *Management (SAD, SPD)*

⇒ *Signaling (IKE)*

⇒ *History (RFC series)*

Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ *Introduction (operation modes, relation to IPv6, extension headers)*

⇒ Architecture (much more than a protocol)

⇒ protocols (ESP, AH)

⇒ Management (SAD, SPD)

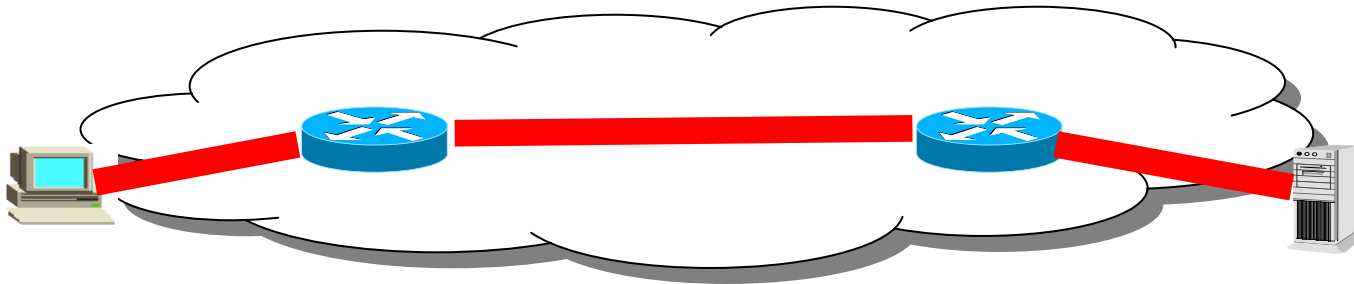
⇒ Signaling (IKE)

⇒ History (RFC series)

IPsec operation modes

→ Transport mode

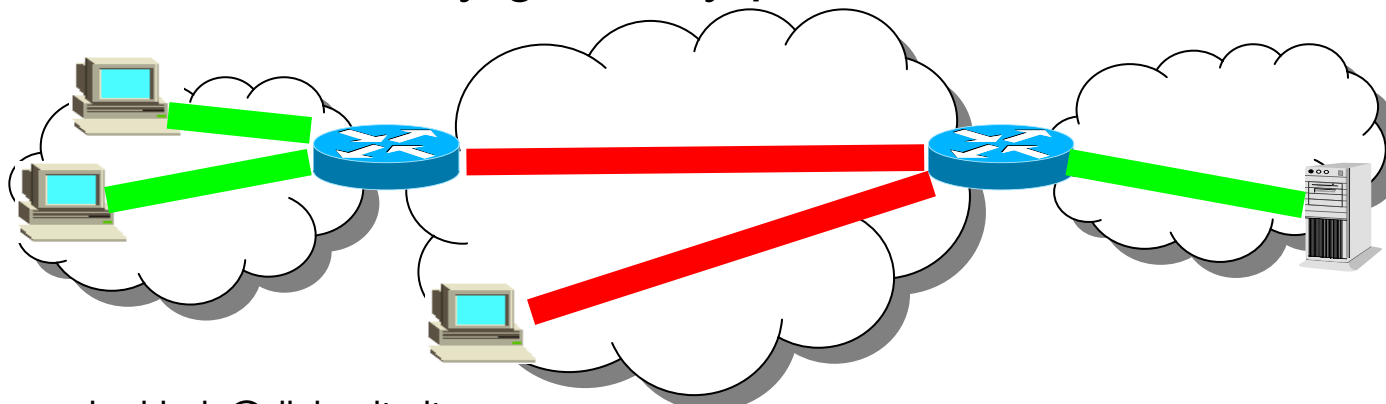
⇒ End-to-end protection



→ Tunnel mode

⇒ Security gateway to Security gateway protection

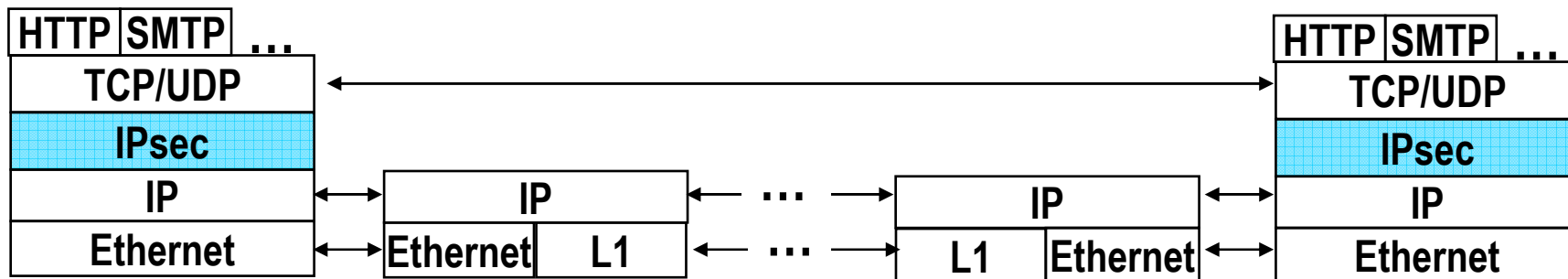
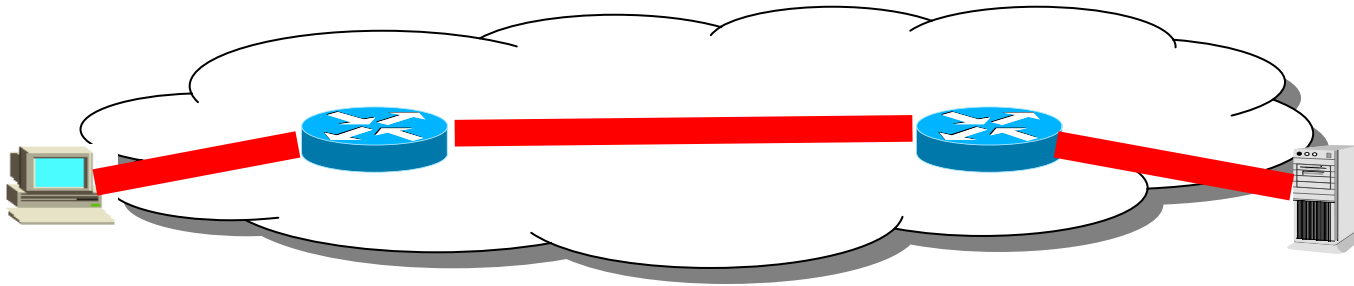
⇒ Host to Security gateway protection



IPsec operation modes

→ Transport mode

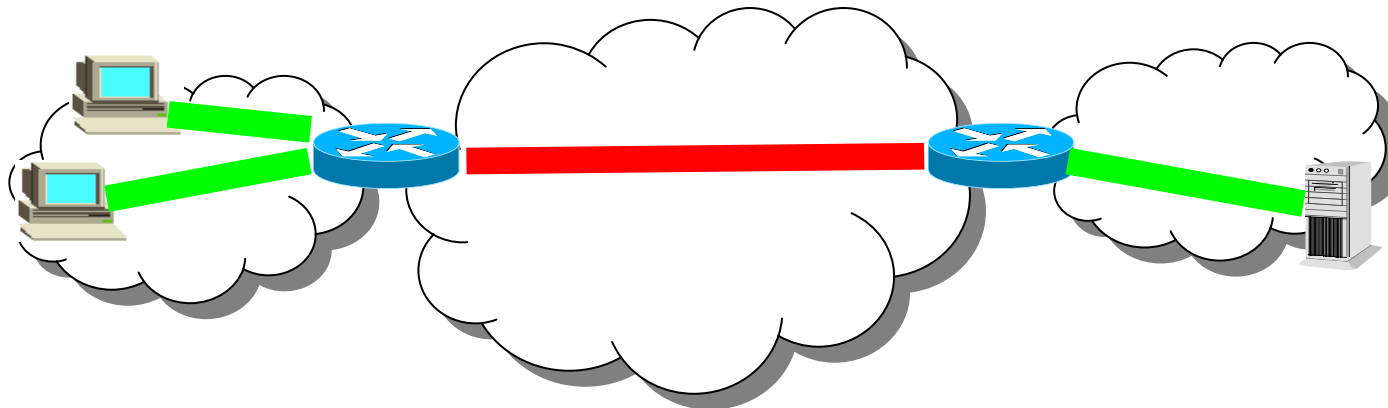
⇒ End-to-end protection



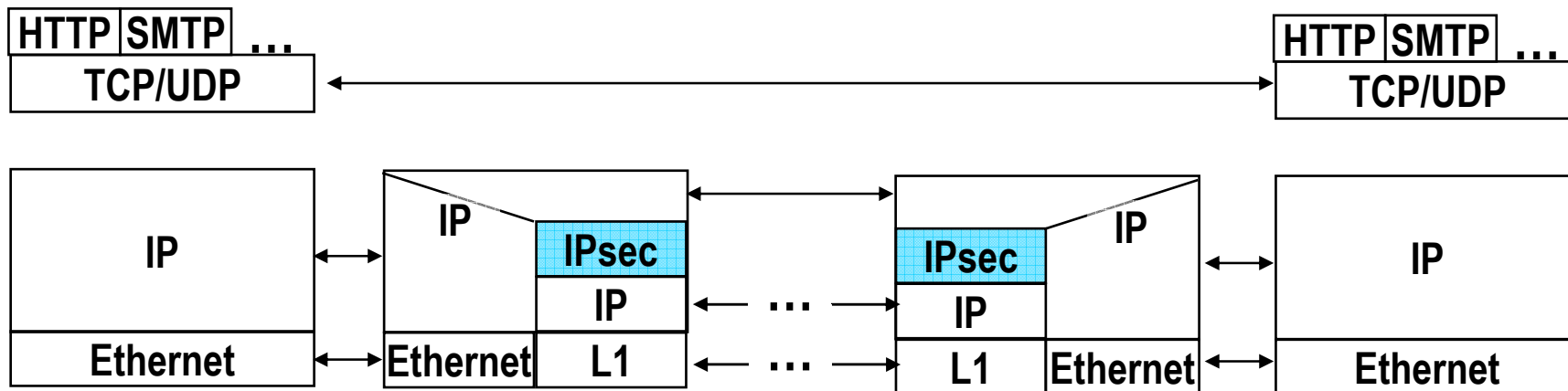
IPsec operation modes

→ Tunnel mode

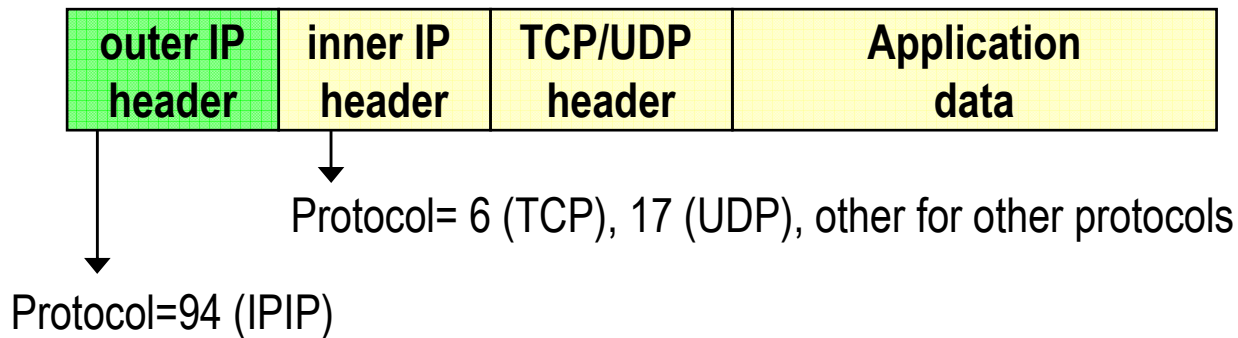
⇒ Security gateway to Security gateway protection



⇒ protocol stacking with IP-in-IP tunneling



IP-in-IP tunneling



- **Encapsulate an IP packet in an IP packet**
 - ⇒ IP can encapsulate other PDUs, not just TCP/UDP/ICMP
 - ⇒ Why not IP itself?
 - ⇒ the “protocol” field should be filled: 94=IPIP
- **Routing is done based on the outer header’s destination IP**
 - ⇒ Internal IP header is not checked by routers
 - ⇒ Protocol field not used in routing (firewalls are problematic)
- **Once this IP packet arrives to its destination (outer), the internal IP packet is decapsulated**
 - ⇒ Routing can continue based on internal destination IP

When transport? When Tunnel?

→ Transport mode: only end-to-end

⇒ End-to-end can also be implemented using tunnel mode, but this implies more overhead (extra IP header)

→ Tunnel mode:

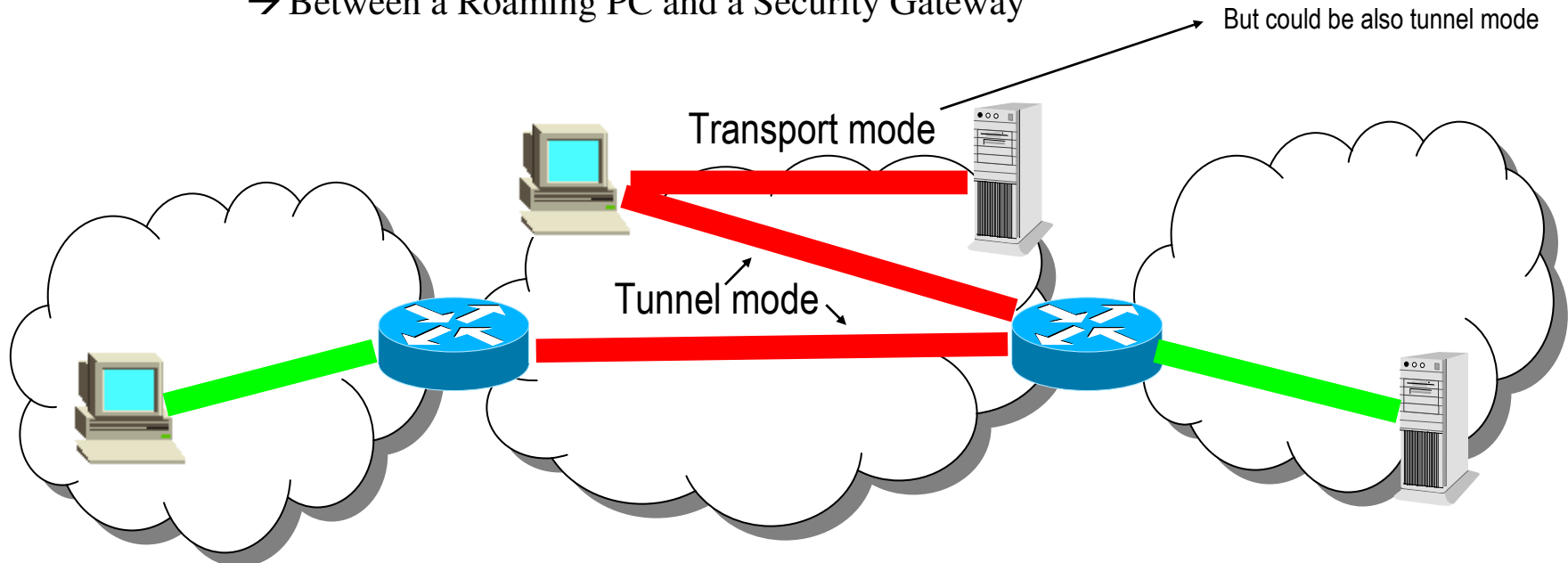
⇒ Nodes can use transport mode only for connections originating/terminating there

⇒ Not when they are intermediary between host and server!

⇒ Tunnel mode used

→ Between Security Gateways

→ Between a Roaming PC and a Security Gateway



Extension headers

→ IPsec was born as part of the “new” IPv6

⇒ In the early `90s: this was 15 years ago !

⇒ Realized soon that IPv4 remains for a while; it needs security as well

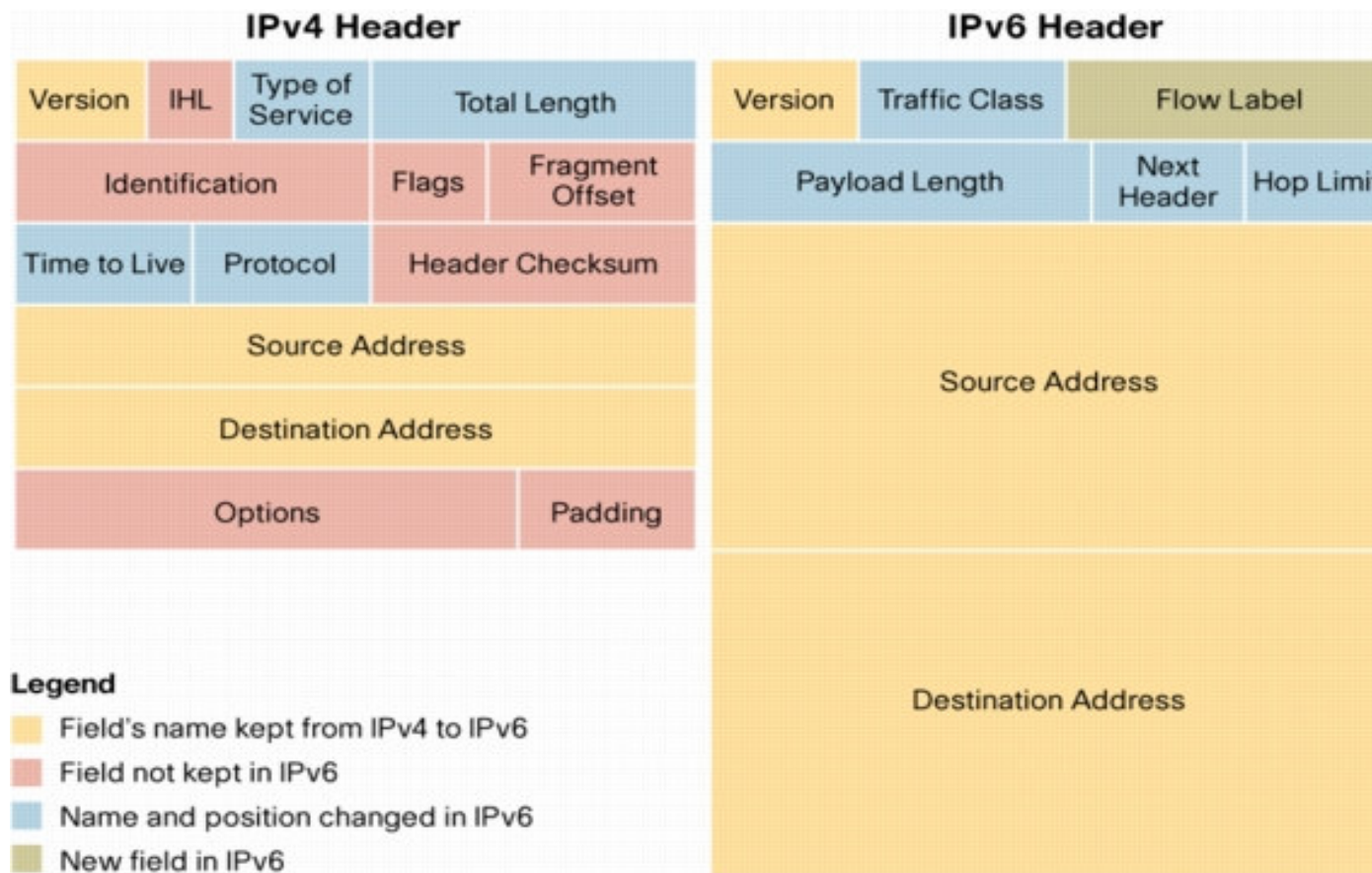
→ made compatible with IPv4

→ But: Header structure of IPsec security protocols (ESP, AH) comes from IPv6 **extension header** concept!

IPv6 Extension headers

→ Header format simplification

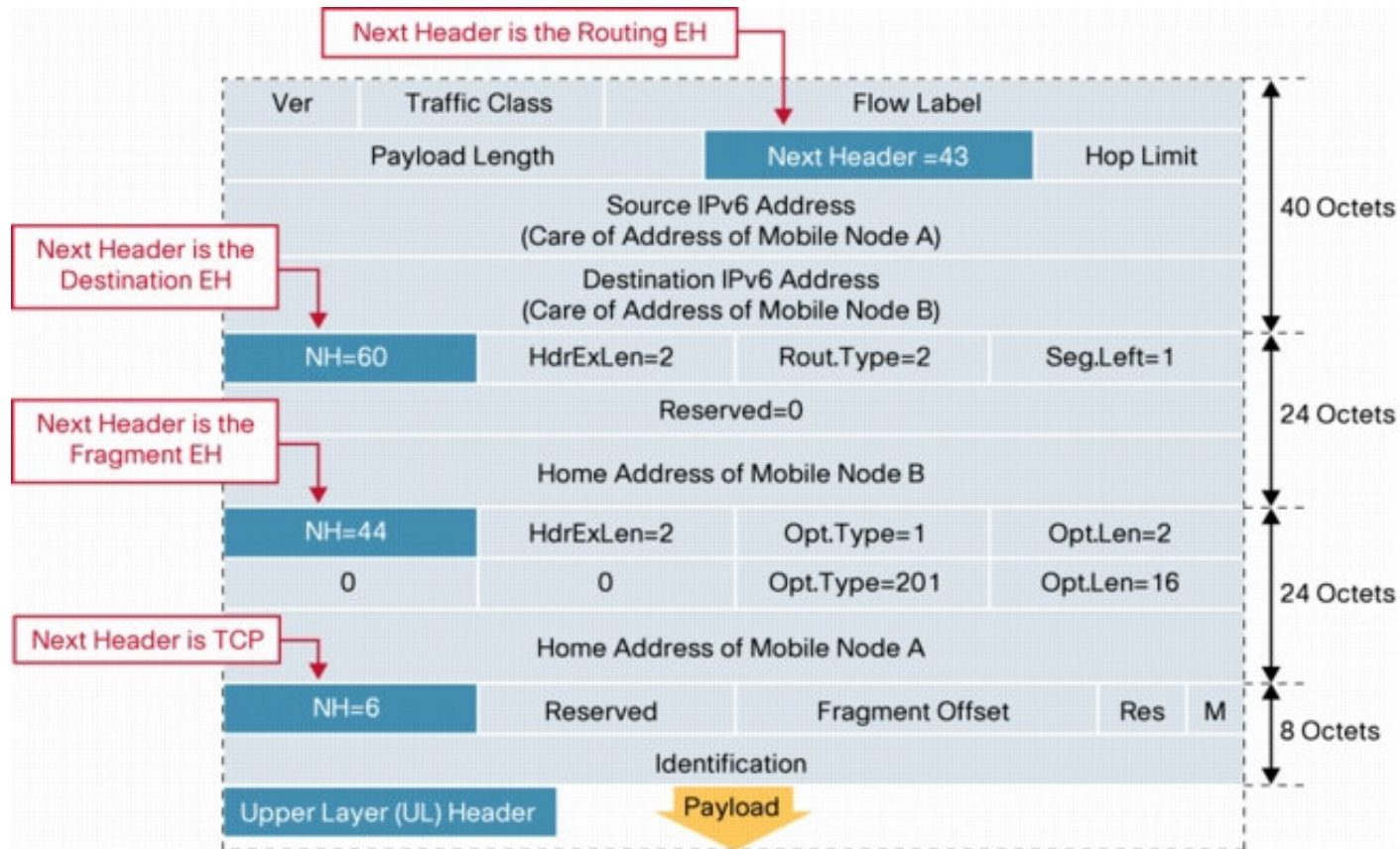
⇒ Options and some fields moved out of the main header



IPv6 Extension headers

→ Protocol stacking inside L3

- ⇒ Next Header: field at fixed position allows identification
- ⇒ HdrExLen: at a fixed position allows skipping if not recognized



“Backporting” extension headers

→ Use IPv4 “protocol” field

- ⇒ The extension header becomes the protocol over IP
- ⇒ But it is still below TCP/UDP

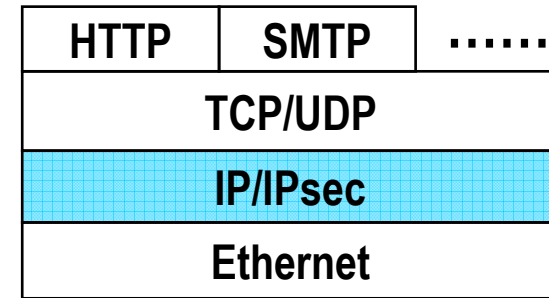
→ Simple trick to add IPsec to IPv4, but

- ⇒ Many implementation difficulties!
- ⇒ So difficult that implementation approaches became part of the RFC

IPsec implementation approaches

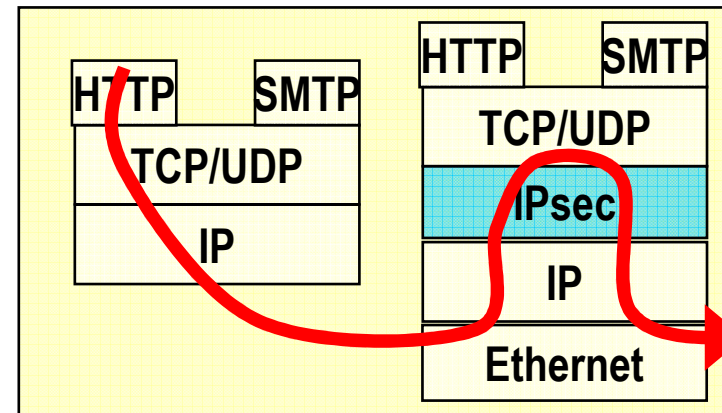
→ Inside the native IP code

- ⇒ Best approach
 - encapsulation order = processing order
- ⇒ But hard to deploy as requires to access and modify IP source code



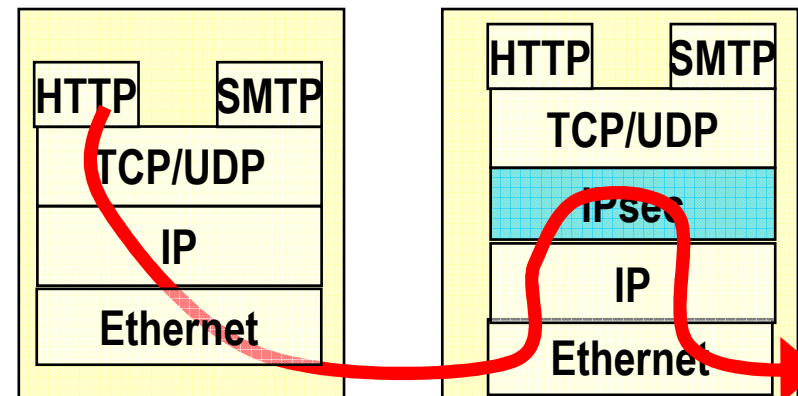
→ Bump in the Stack (BITS)

- ⇒ Ugly!
 - encapsulation order != processing order
- ⇒ Between native IP code and device driver
- ⇒ First, IP PDU is created as usual
- ⇒ Then, IPsec part inserted
 - IP checksum recalculated



→ Bump in the Wire (BITW)

- ⇒ Even worse!
- ⇒ Implemented in dedicated hardware
- ⇒ External security processor changes things above IP



Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ *Architecture (much more than a protocol)*

⇒ protocols (ESP, AH)

⇒ Management (SAD, SPD)

⇒ Signaling (IKE)

⇒ History (RFC series)

IPsec: Security Architecture for IP

→ **IPsec is not a protocol, but a complete architecture!**

Components:

1. Security Protocols (ESP, AH), each having different
 - Protocol header
 - Implemented security mechanisms
 - Provided security services
2. Cryptographic Algorithms (3DES, etc.)
 - Used by security protocols
 - Each having advantages/disadvantages, e.g.
 - » Computational complexity
 - » Block size
3. Management concepts and local management databases
 - Security Policies (SP):
 - » established and maintained by a user or system administrator
 - » select IP packets where IPsec should be applied
 - Security Associations (SA):
 - » simplex "connection" that affords security services to the traffic carried by it
4. Signaling protocols
 - Internet Key Exchange (IKEv2)

Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ Architecture (much more than a protocol)

⇒ *Protocols (ESP, AH)*

⇒ Management (SAD, SPD)

⇒ Signaling (IKE)

⇒ History (RFC series)

IPsec Security Protocols

AH, ESP

(discuss IPv4 only)

Services provided

→ **AH: Authentication Header**

- ⇒ Data integrity protection and data origin authentication
 - Covers both payload and parts of IP header that do not modify in transfer
- ⇒ Protection against replays
 - Optional, through extended sequence numbers

→ **ESP: Encapsulated Security Payload**

- ⇒ Same services as AH
 - authentication limited to IP payload only!
- ⇒ Confidentiality through encryption
- ⇒ Traffic flow confidentiality
 - Improved privacy against eavesdropping
 - Through padding and dummy traffic generation

AH

Authentication Header

Authentication Header

→ Security services

- ⇒ data integrity protection
 - detect modification
 - detect replay
- ⇒ data origin authentication

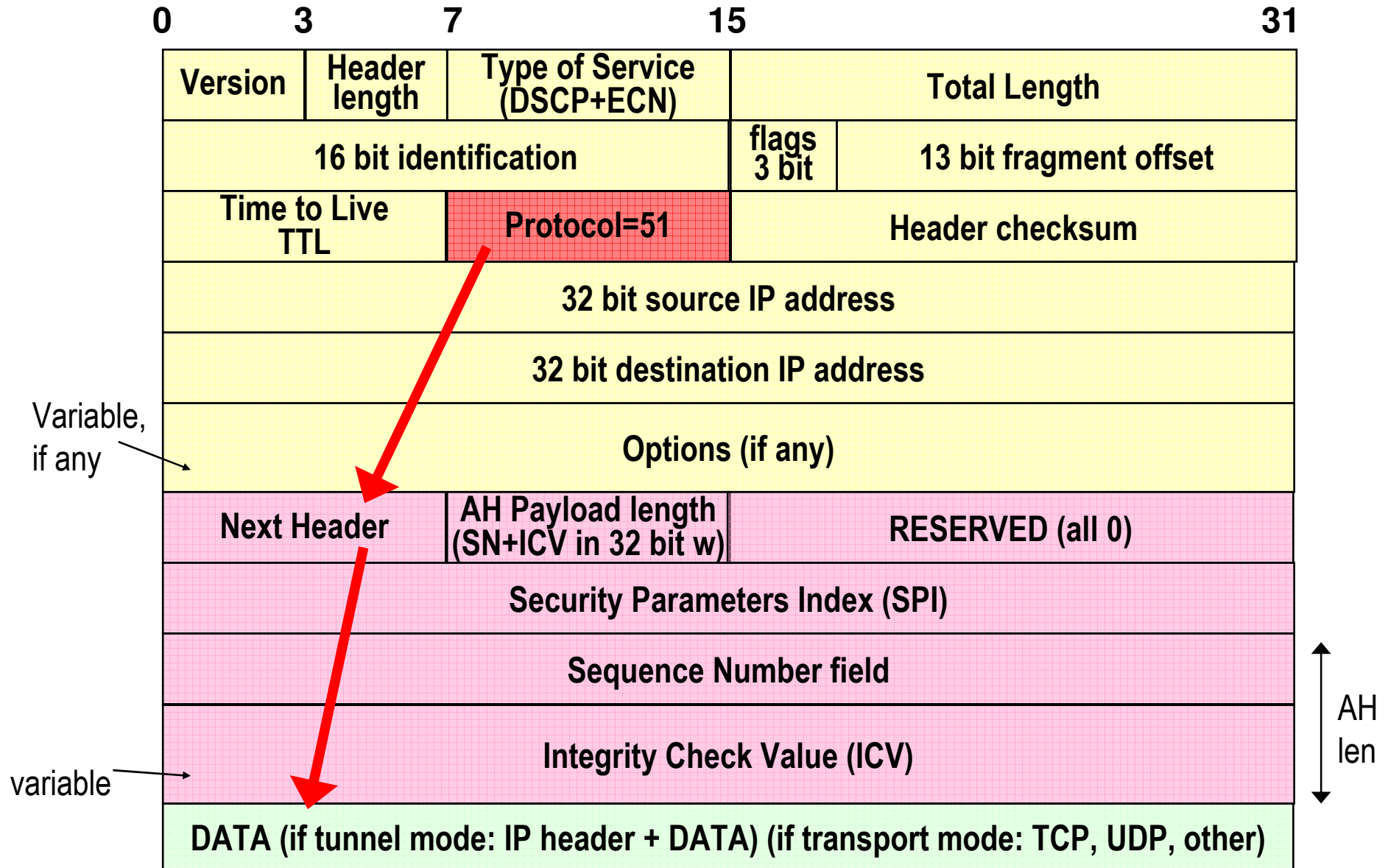
→ Security mechanisms used

- ⇒ Integrity Check Value (ICV) calculation
 - applied packet-per-packet
 - based on a **key** and
 - » symmetric encryption algorithms (e.g., AES) or
 - » one-way hash functions (e.g., MD5, SHA-1, SHA-256, etc.).
 - Algorithm and key stored in Security Association Database
- ⇒ Replay protection based on sequence numbers

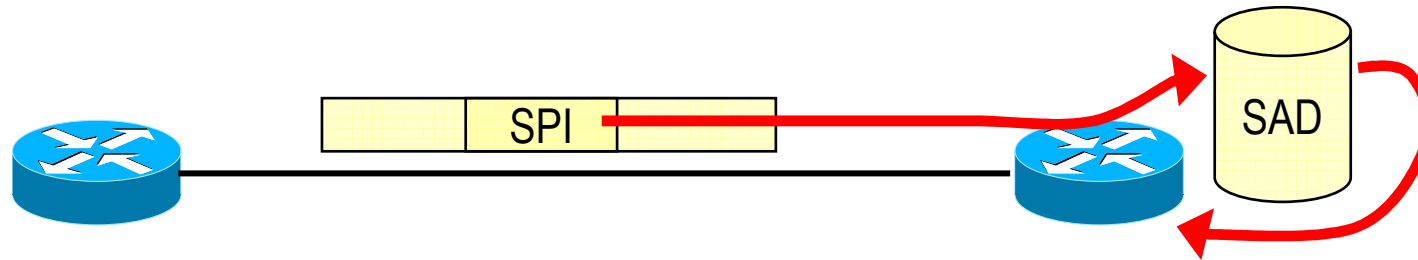
→ Role of Security Policy

- ⇒ Decide whether packet should be protected
- ⇒ If protection needed and Security Association is not available -> trigger IKEv2 for the SA negotiation

Authentication Header



Security Parameters Index



→ **32 bit index**

→ **Role: like port number in TCP and UDP**

→ **Used to lookup the SAD at destination**

⇒ Lookup also uses

→ destination address

→ source address

→ security protocol (AH/ESP)

→ **Retrieves algorithms and parameters that allow to process received packet**

Integrity Check Value computation

→ Only on immutable fields in the IP header

⇒ Or mutable but predictable

→ e.g. destination address with strict/loose source routing option

→ Mutable fields set to 0 during ICV computation

⇒ Highlighted in red in next figure

→ Note: AH apply before fragmentation, and checked after reassembly

→ Options classified as either mutable or not

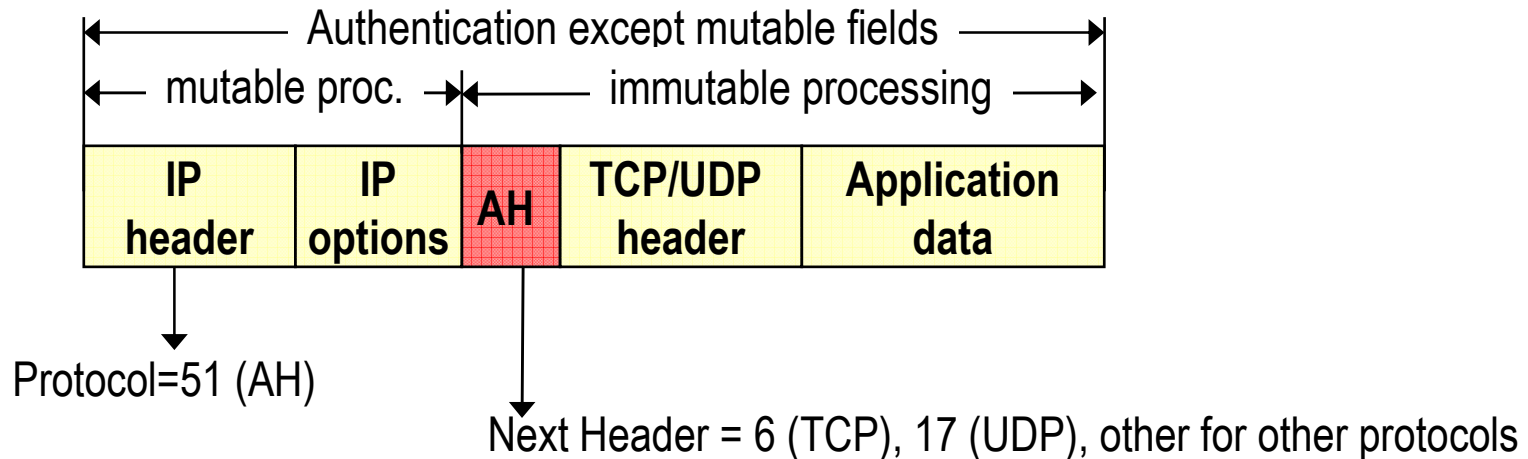
→ Mutable options: details in appendix A RFC 4302

→ mutable options = all zeroed

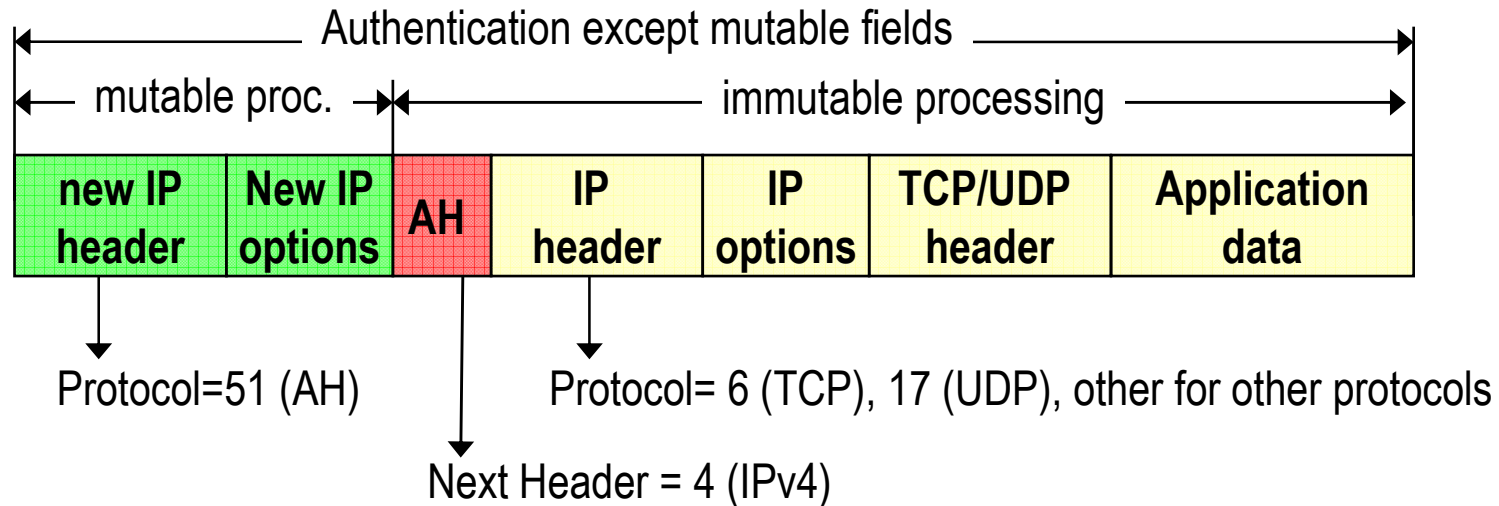
Version	Header length	Type of Service (DSCP+ECN)	Total Length	
16 bit identification			flags 3 bit	13 bit fragment offset
Time to Live TTL	Protocol=51 (AH)		Header checksum	
32 bit source IP address				
32 bit destination IP address				

Transport mode, tunnel mode

Transport mode:



Tunnel mode:



Why sequence number?

→ IP header **DOES NOT** contain a sequence number!

⇒ Hence replay of an authenticated IP packet is possible

→ And may alter in an unpredictable manner the overlaying service (e.g. ICMP replies can be dangerous ☺)

→ **Sequence number: 32 bit counter**

⇒ Initialized to 0 when the Security Association is established

⇒ Increments of 1 per each transmitted packet

→ First transmitted packet: SN=1

⇒ Maximum value $2^{32}-1$, afterwards Security Association must be terminated

→ No counter cycling allowed when anti-replay service active

→ Anti-replay: optional (but default = on)

» Anti-replay typically OFF when manual (static) keys configured

Extended Sequence Number

→ $2^{32} \sim 4.3$ billion

⇒ A lot, but not REALLY a lot!

→ Packet size = 1500 (1460 bytes payload)

→ $2^{32} \times 1460$ bytes = 6270 GB

→ About 14 h transmission of a 1 Gbps link

→ **Extended Sequence Number:**

⇒ 64 bits - this should be enough, now 😊

⇒ Transmit only low order 32 bits

⇒ But use high order 32 bits in ICV computation!

Anti-replay

→ Sliding Window W

⇒ Size locally decided at receiver

→ Minimum = 32; default = 64; higher values recommended for high speed links

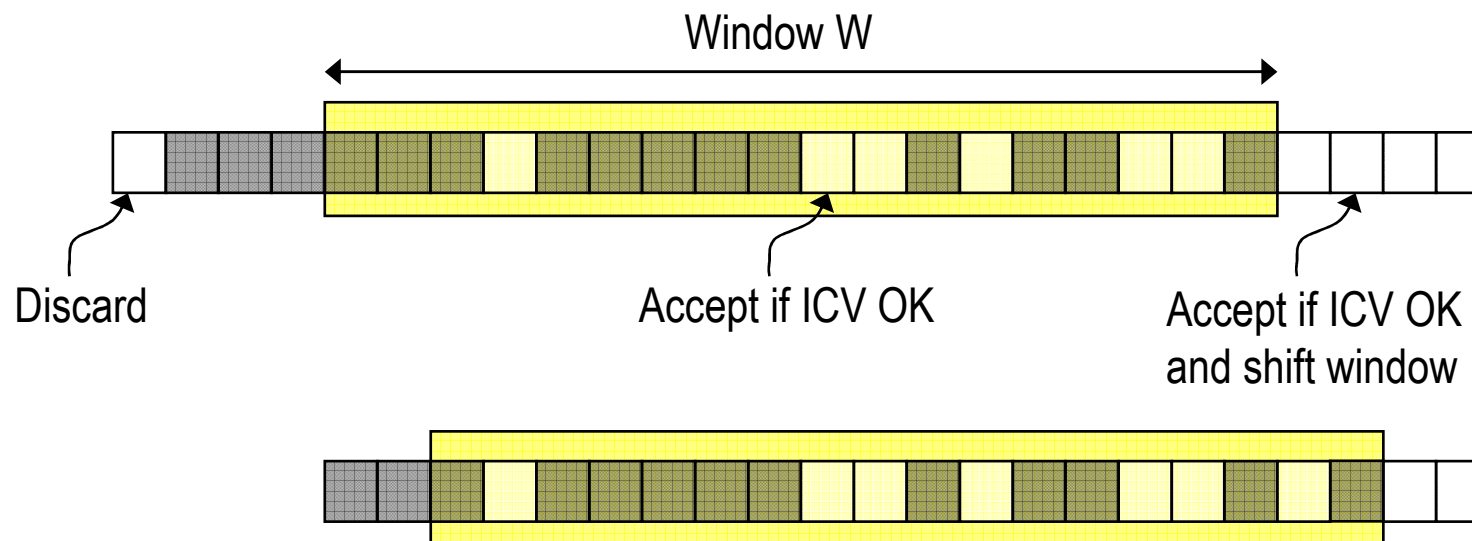
→ eventually very large: maximum $2^{31}-1$ with SN and $2^{32}-1$ with ESN

⇒ Window right margin = highest NS packet received

→ Duplicates discarded

→ Packets out of left window edge discarded

→ Packets greater than right window margin make W shift



ESP

Encapsulated Security Payload

Encapsulated Security Payload

→ Security services

⇒ Same services as AH

→ authentication limited to IP payload only!

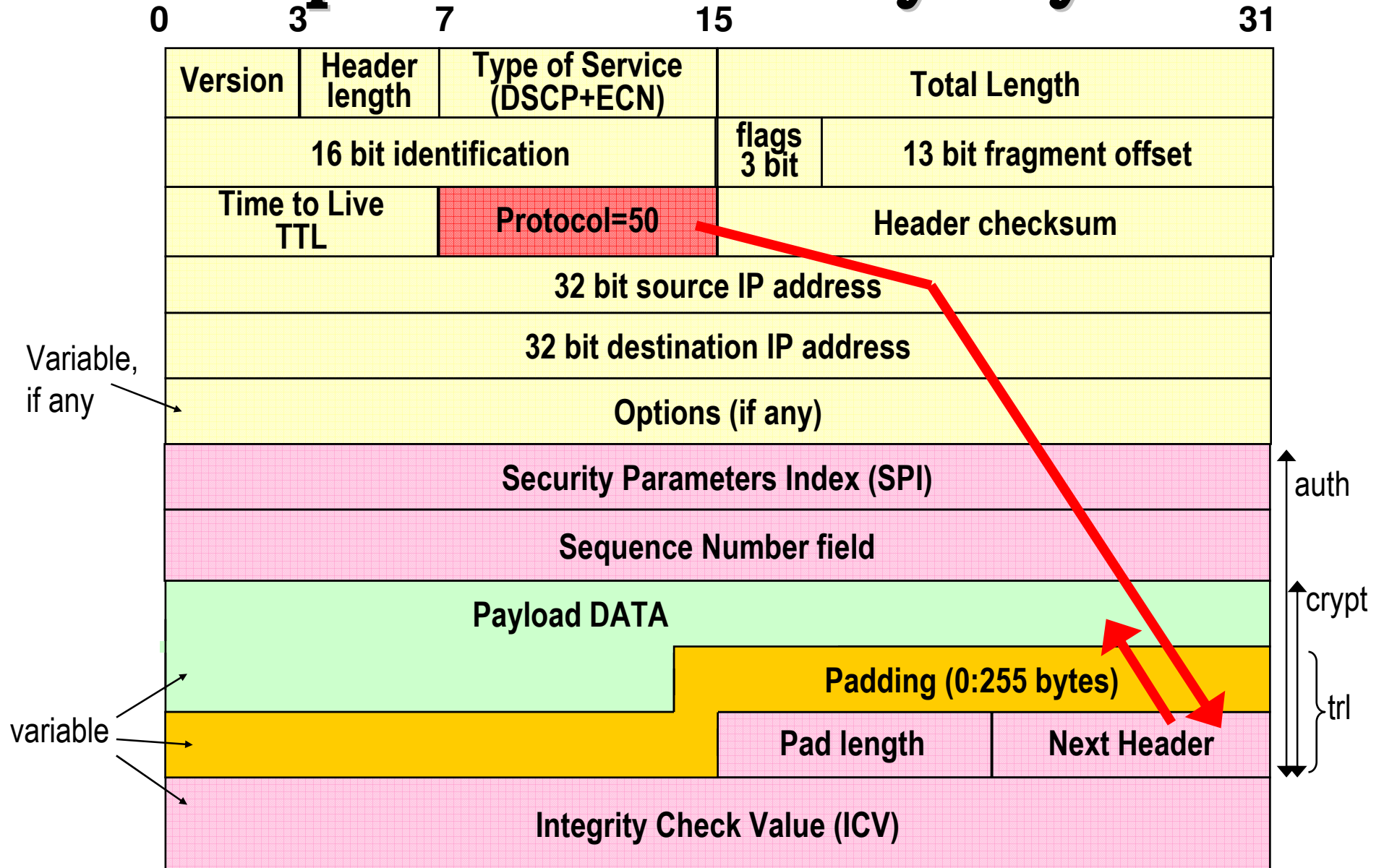
⇒ Confidentiality through encryption

⇒ Traffic flow confidentiality

→ Improved privacy against eavesdropping

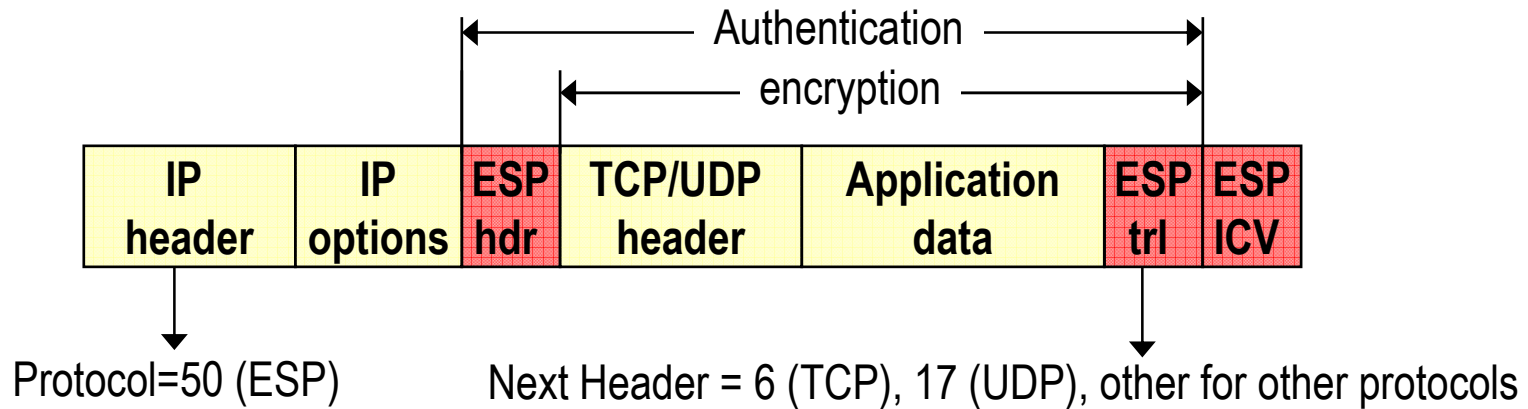
→ Through padding and dummy traffic generation

Encapsulated Security Payload

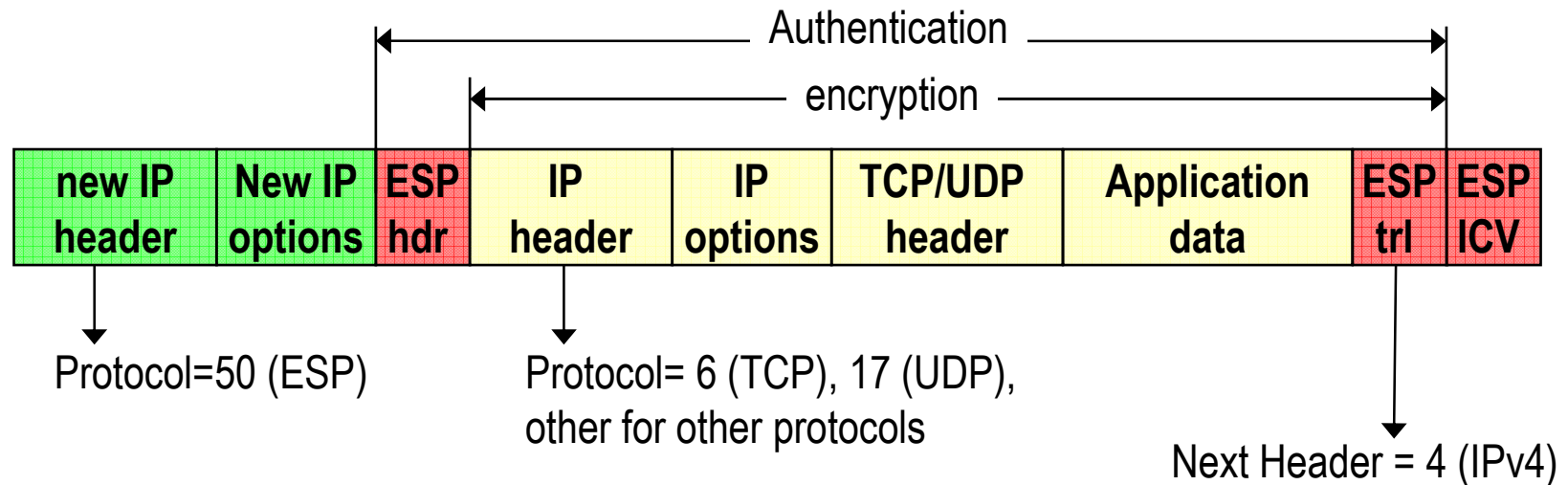


Transport mode, tunnel mode

Transport mode:



Tunnel mode:

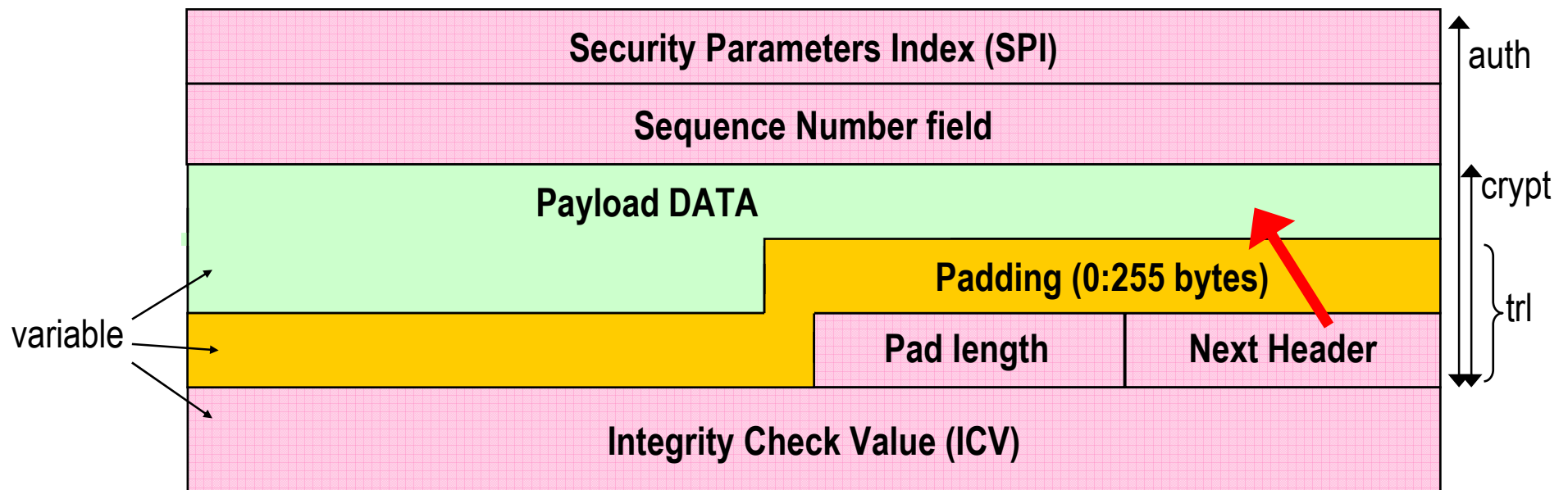


ESP Header

→ 8 bytes, SPI + SN

⇒ Same as AH (including ESN and optional anti-replay)

⇒ Plain text



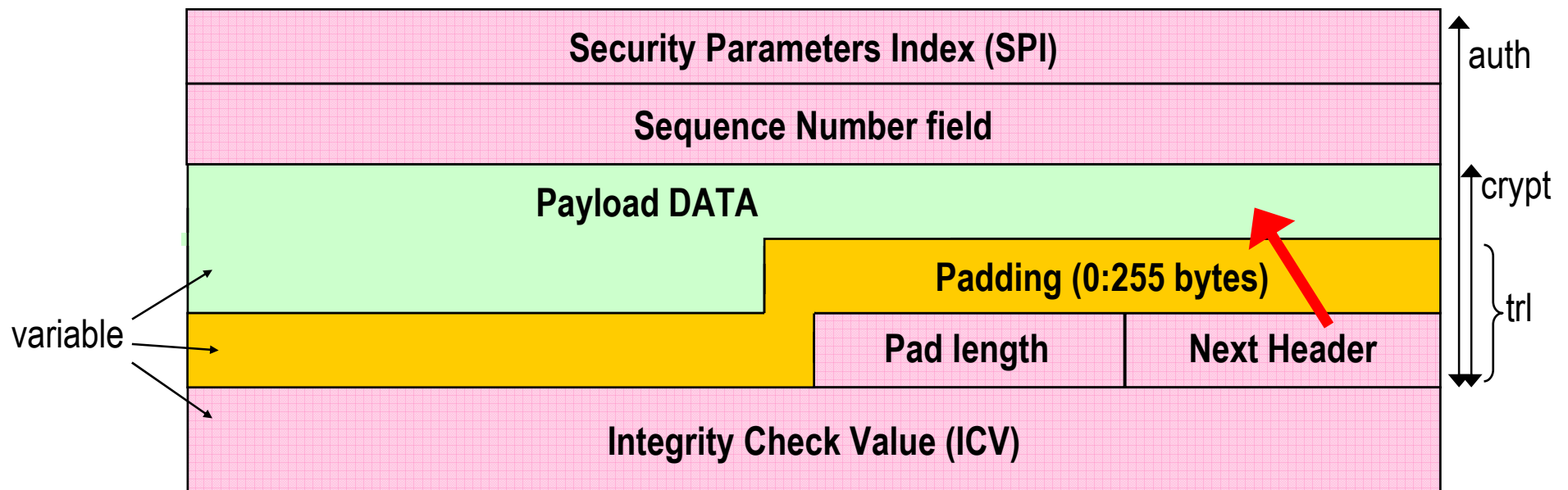
ESP Trailer

→ Padding (variable size) + pad length + next header

- ⇒ Pad length = 1 byte = 0:255
- ⇒ Next Header = 1 byte = type of data payload
 - Next Header = 4 → Tunnel mode (IP header inside)
 - Next Header = 59 → Dummy packet!!

→ Padding: for two reasons

- ⇒ Encryption algorithm may require plaintext to be a multiple of some number of bytes
 - E.g. block size of a block cipher
- ⇒ Resulting ciphertext must terminate on a 4-byte boundary



Encryption & IP unreliability

→ **A problem: some good encryption algorithms need to maintain synchronization**

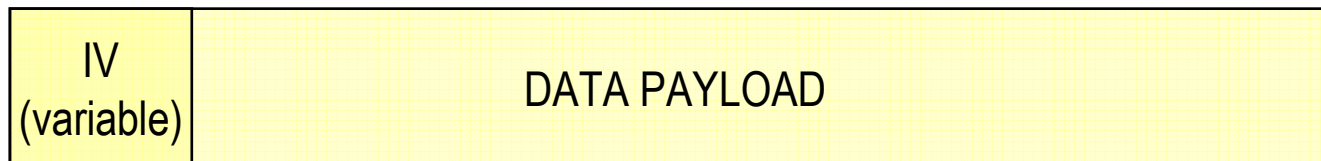
⇒ E.g. Cipher Block Chaining, etc

→ **However IP packets may arrive out of order or may be lost!**

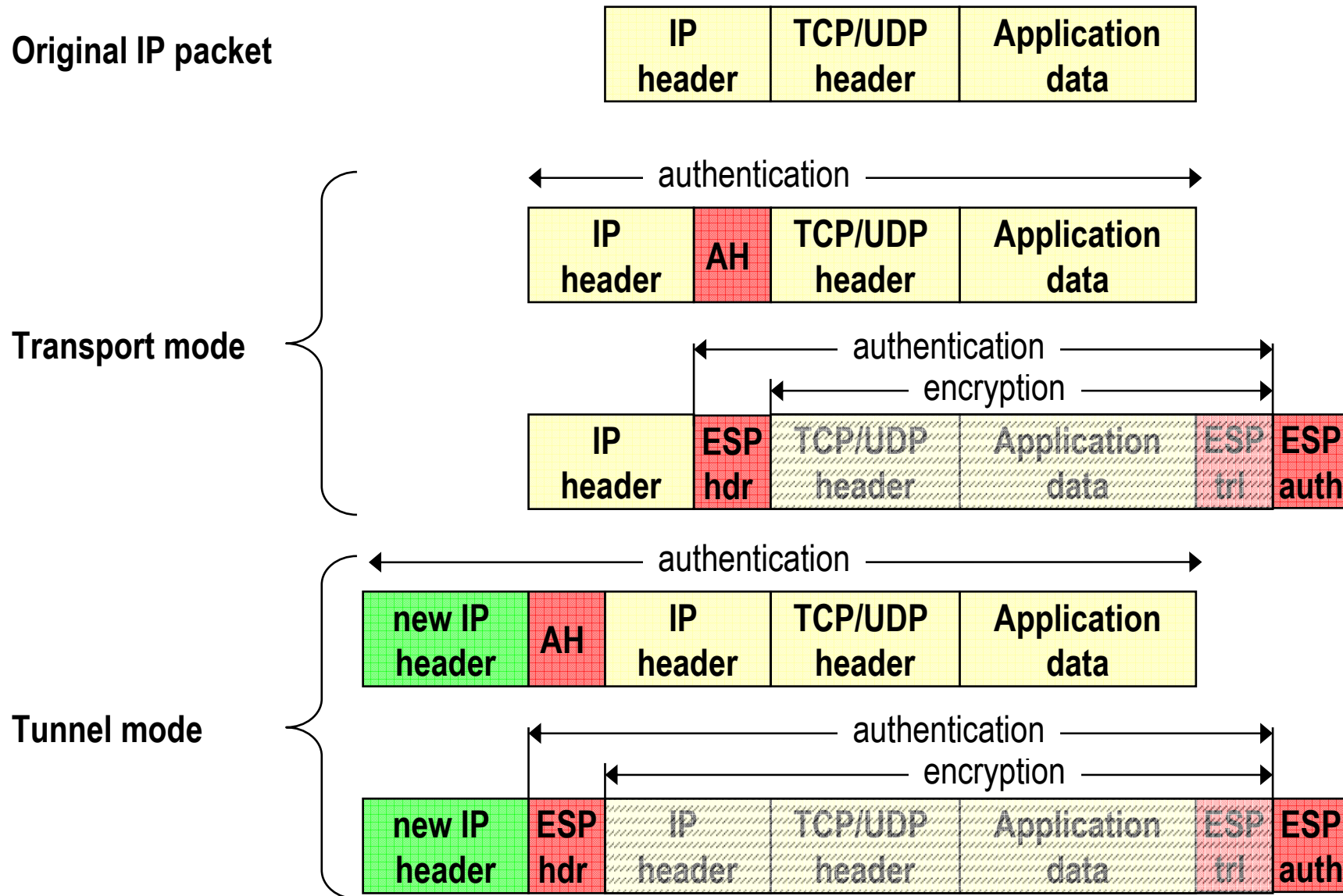
→ **Consequence: for some encryption algorithms, packets must carry data needed to resync the deciphering**

⇒ If such data (per-packet Initialization Vector) necessary, IV prepended to DATA payload

→ Not an ADDITIONAL ESP header! i.e. it is “invisible” to ESP



Transport vs Tunnel – AH and ESP



Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ Architecture (much more than a protocol)

⇒ protocols (ESP, AH)

⇒ *Management (SAD, SPD)*

⇒ Signaling (IKE)

⇒ History (RFC series)

IPsec management

SA: Security Association
SAD: SA Database

SP: Security Policy
SPD: SP Database

SPI: Security Parameters Index

Security Association

→ **Fundamental concept in IPsec**

→ **May involve:**

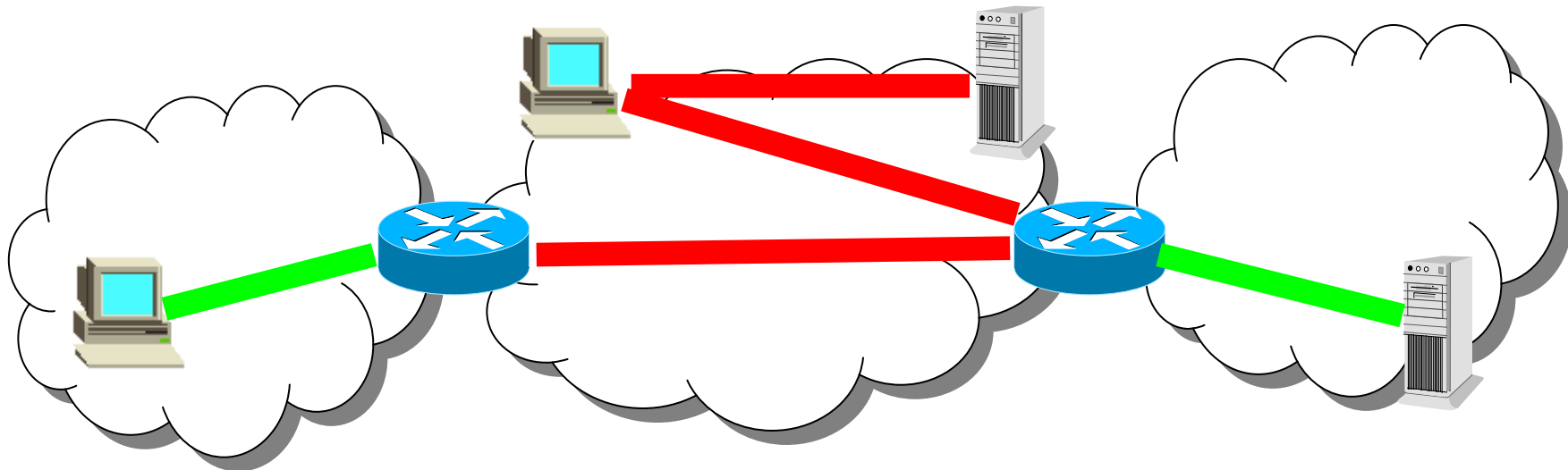
⇒ Host to host

⇒ Host to intermediate router (security gateways)

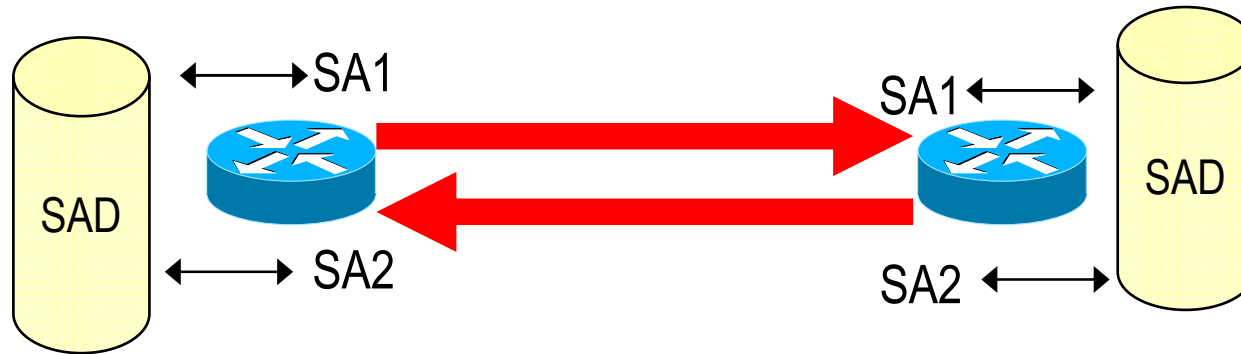
⇒ Security gateway to security gateway

→ **Defines the boundaries for IP packets authentication/encryption**

⇒ A “connection” with security services active



SA: unidirectional!



→ SPI = Security Parameters Index

⇒ The (somewhat) unique “name” of an SA

→ SAD = Security Associations Database

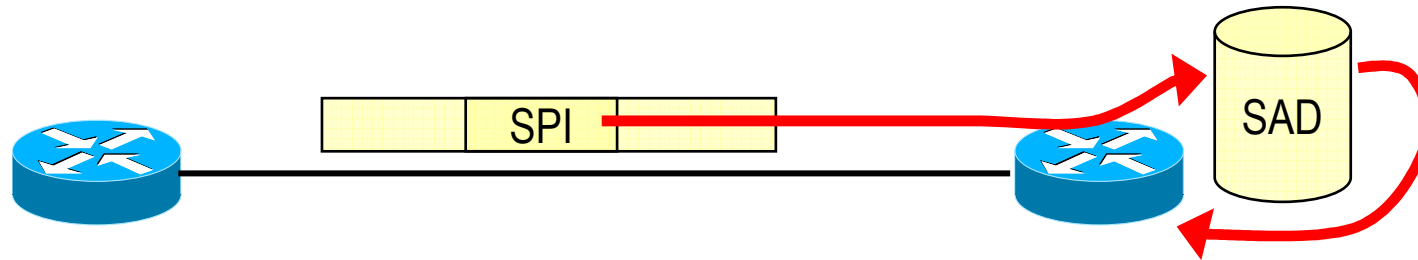
⇒ SPI = search key (at least)

⇒ Stores type of security protocol per each SA, with related parameters

→ E.g. which encryption algorithm; shared key for encryption, SA lifetime, Sequence number counter, etc.

⇒ SA should be in SAD on both sides, at sender and at receiver!

Security Parameters Index



- ➔ **32 bit index**
- ➔ **Role: like port number in TCP and UDP**
 - ⇒ Allows multiple SAs between the same two hosts
- ➔ **Used to lookup the SAD at destination**
 - ⇒ Lookup also uses
 - destination address
 - source address
 - security protocol (AH/ESP)
- ➔ **Retrieves algorithms and parameters that allow to process received packet**

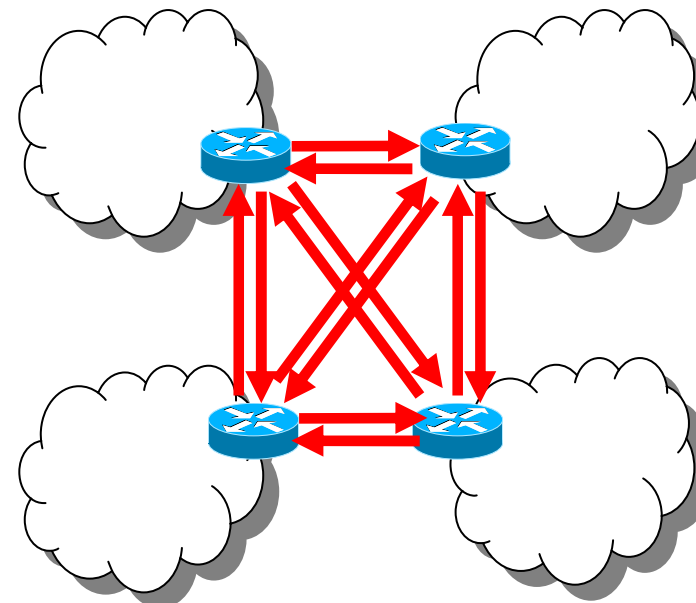
Security Association and Key management

→ Manual

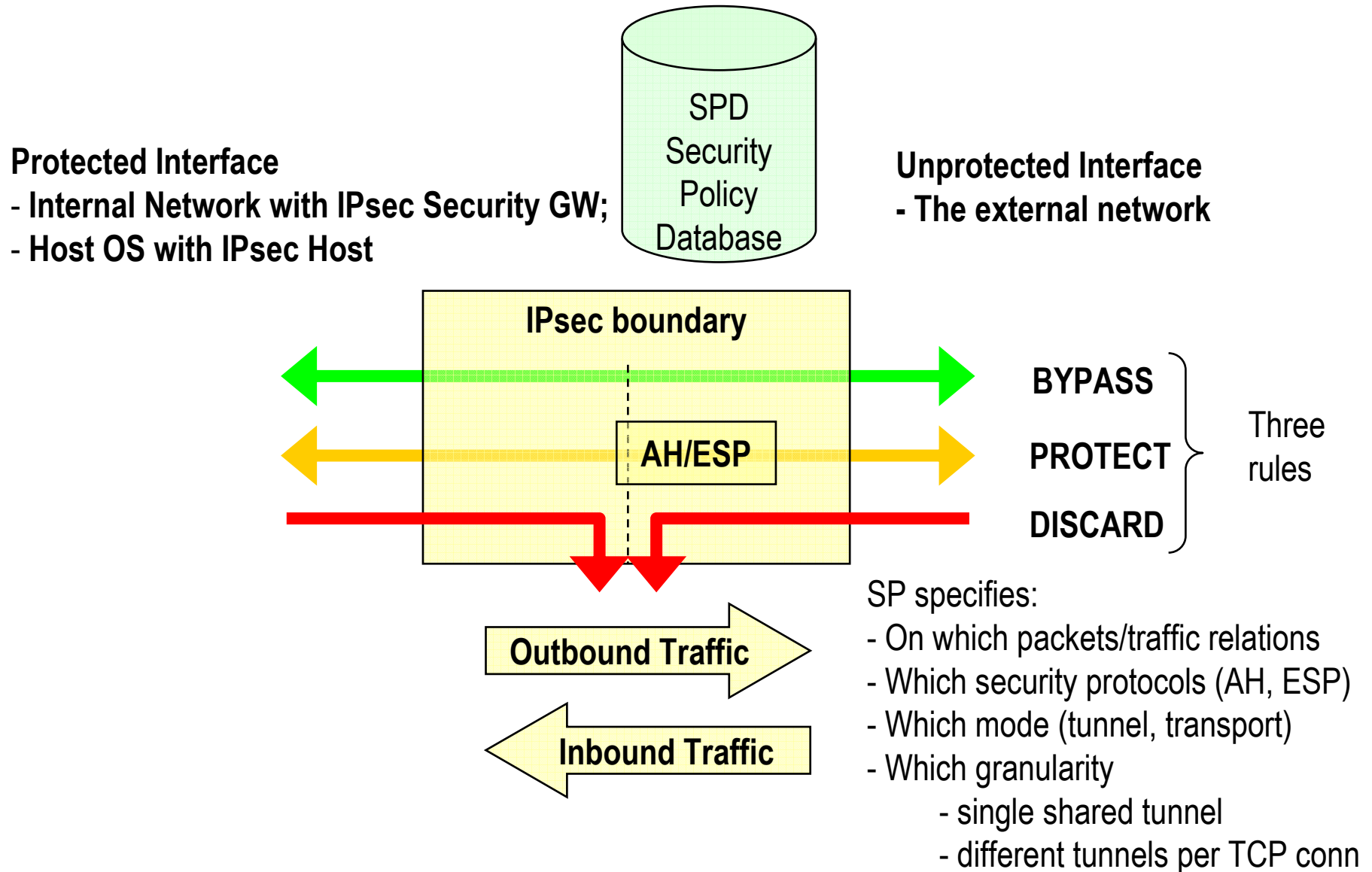
- ⇒ Manually configure each SA and related crypto keys
 - static, symmetric
- ⇒ Typical in small-scale VPNs
 - Few security gateways, e.g. one per site
 - Meshed SA connections

→ Automatic

- ⇒ SA management through IKEv2
 - In the past, through the combined operation of several protocols
 - » IKE+ISAKMP+others
- ⇒ On-demand SA creation
- ⇒ Session-oriented keying/rekeying



IPsec protection & access control



IPsec processing

1. PDU enters IPsec processing: two possibilities

- ⇒ Host: PDU from upper layer arrives, or
- ⇒ Security GW: IP packet arrives

2. SPD searched for matching SP

- ⇒ Search based on IP addresses, higher layer protocol, port number, etc.

3. If SP found:

- a) If BYPASS: no IPsec processing needed
- b) If DISCARD: PDU dropped (like in a firewall)
- c) If PROTECT: we know that we have to protect, but we don't know how! It is defined in an SA. Search for corresponding SA in SAD

4. If SA found, apply it

- ⇒ Encapsulate in ESP or AH, with the parameters of the SA
- ⇒ Encapsulate in IP if tunnel mode

5. Send protected packet

IPsec processing

What happens If SP is not found?

- ⇒ No problem, IPsec treatment not needed
- ⇒ PDU goes as it would go otherwise

What happens If SA is not found?

- ⇒ That is a problem: packet must be protected, but we don't know how
- ⇒ SA should be negotiated with other side
- ⇒ Automatic keying is triggered, IKE starts ...

Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ Architecture (much more than a protocol)

⇒ protocols (ESP, AH)

⇒ Management (SAD, SPD)

⇒ *Signaling (IKE)*

⇒ History (RFC series)

IPsec signaling IKEv2

Rationale for IKE

→ shared state must be maintained between source and sink

- ⇒ Which security services (AH, ESP)
- ⇒ Which Crypto algorithms
- ⇒ Which crypto keys

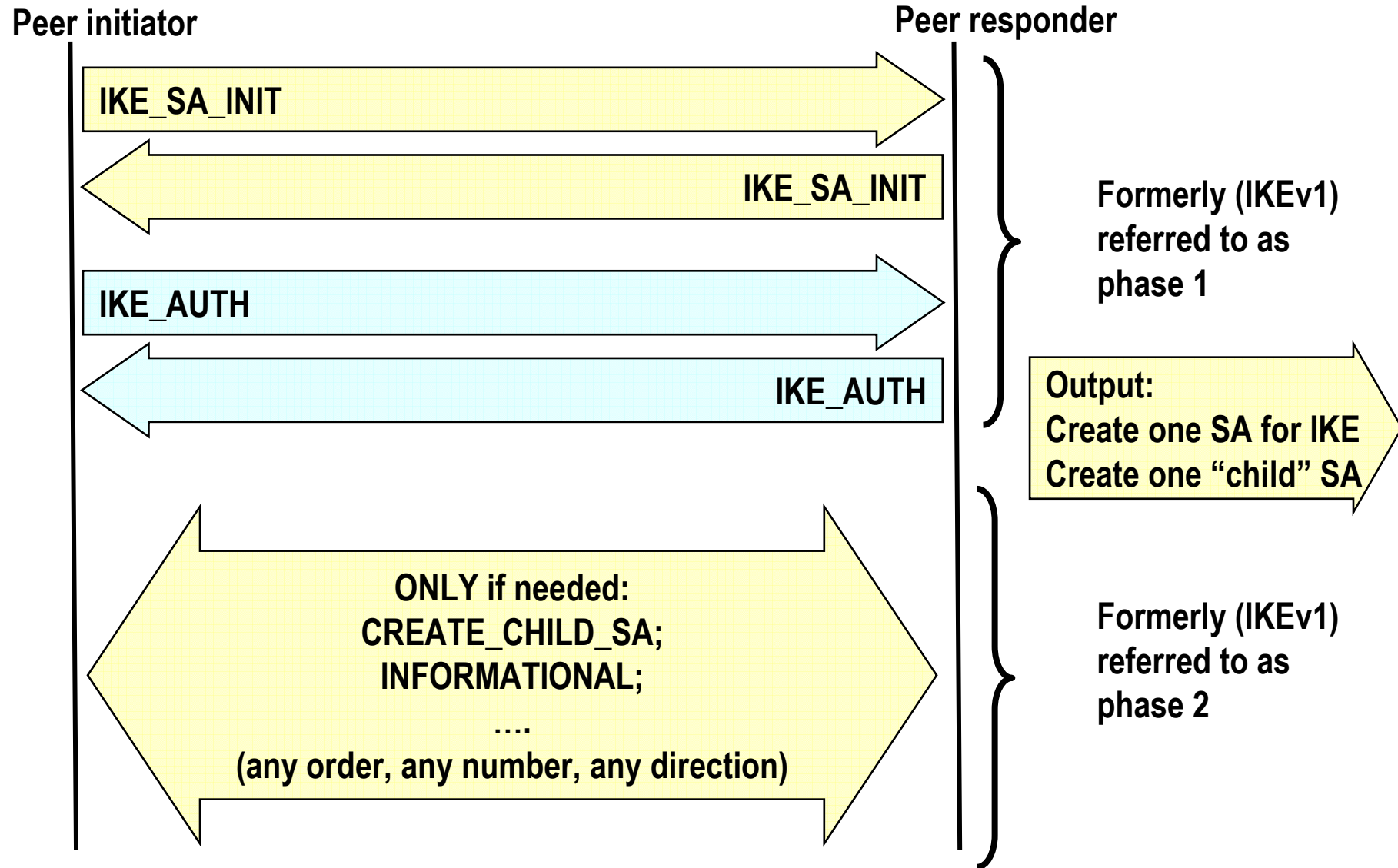
→ Manual maintenance not scalable

- ⇒ Partially OK only for small scale VPNs
- ⇒ In any case, weak approach
 - Infinite lifetime SA → no rekeying!

→ IKE = Internet Key Exchange protocol

- ⇒ Goal: dynamically establish and maintain SA
- ⇒ IKE now (december 2005, RFC 4306) in version 2
 - Replaces protocols specified in RFCs 2407, 2408, 2409 (IKE, ISAKMP, DOI)
 - IKEv2 quite different (and much cleaner!!) than former specifications

IKE phases at a glance



IKE SA and CHILD SA

→ IKE SA

⇒ Security association to exchange IKE messages

→ CHILD SA

⇒ Security association to exchange data messages

→ Making use of AH or ESP

⇒ Many CHILD SA may be set up between two peers

IKE message format

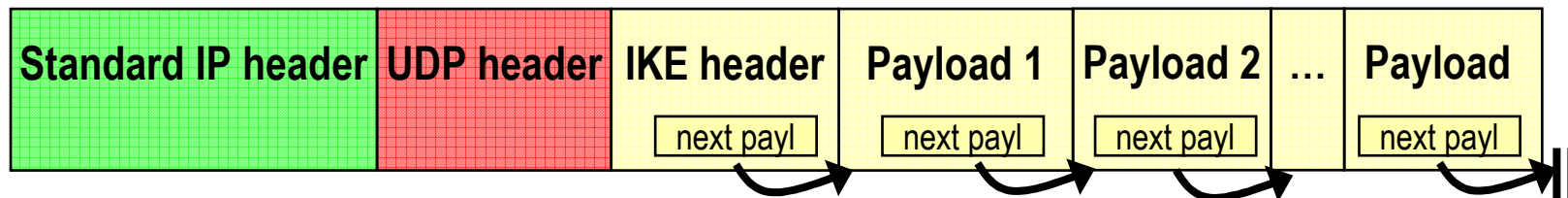
→ UDP encapsulated

- ⇒ Ports 500 and/or 4500
- ⇒ Reliable delivery managed by IKE through retransmission
 - New in IKEv2

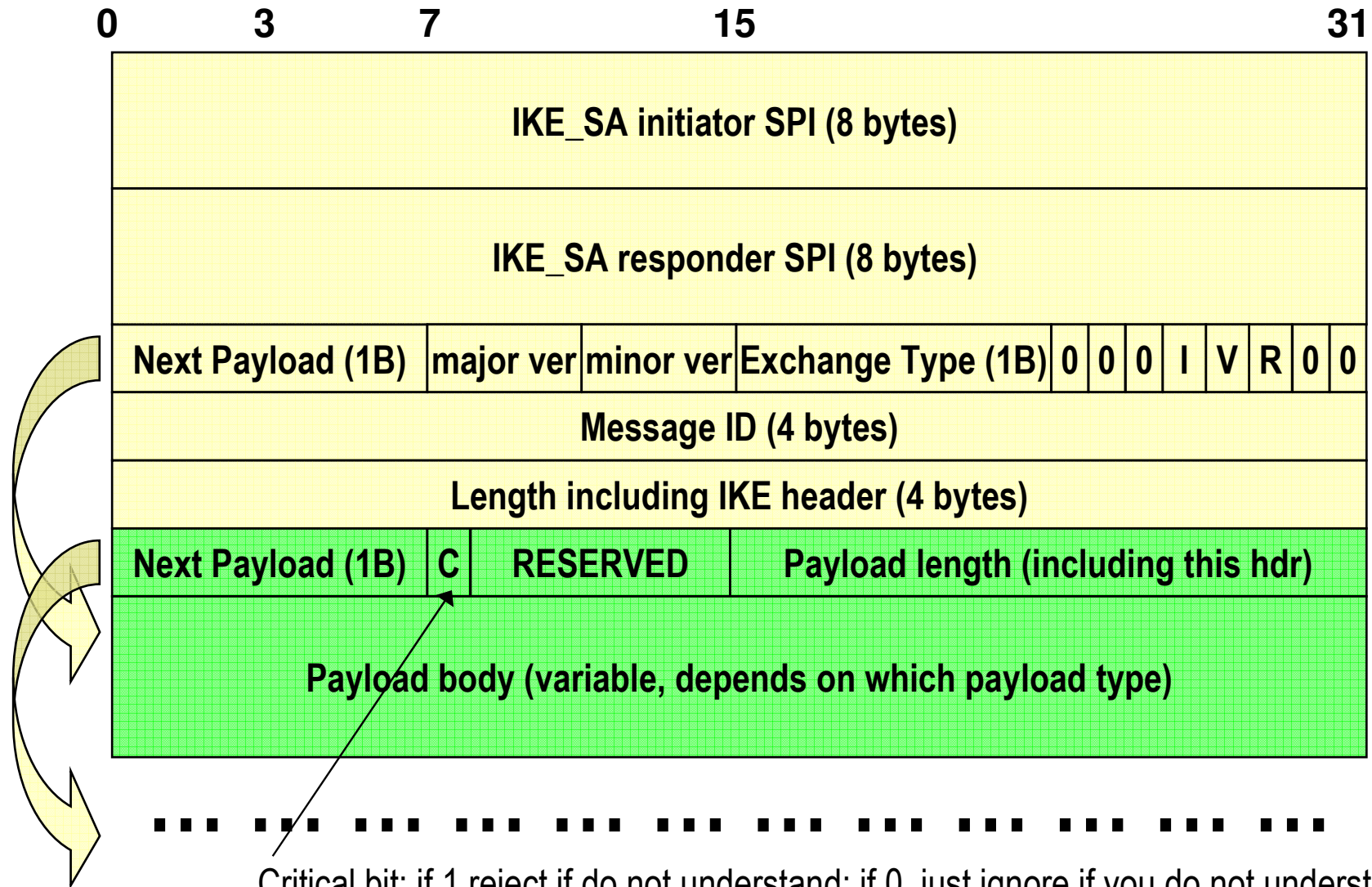
→ IKE header first

→ Followed by one or more IKE payloads

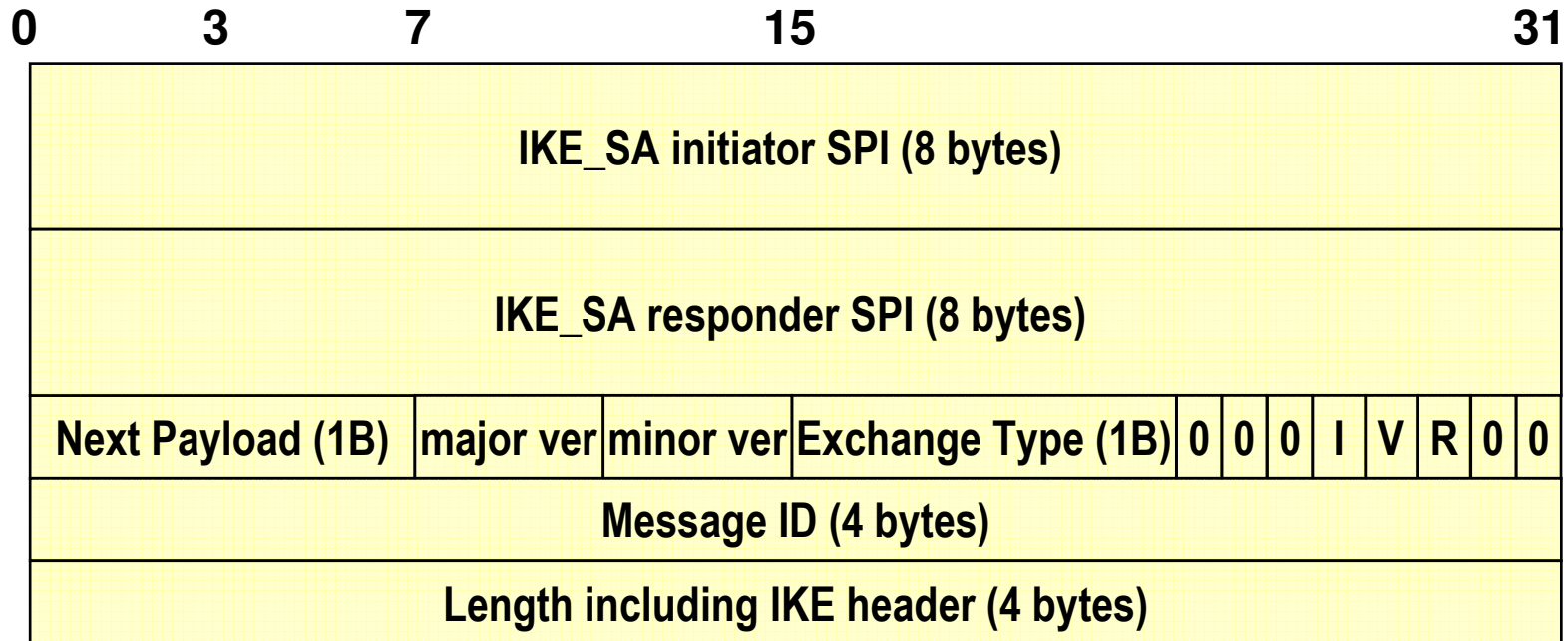
- ⇒ for a perhaps stretched analogy think to Attribute-Value Pair (AVP) concept;
- ⇒ a more appropriate analogy is with the extension header concept of IPv6)
 - flexible approach: new payloads added at later stages



IKE hdr, generic payload hdr



IKE header (explanation)



- **Initiator SPI, responder SPI:**
 - ⇒ SA id for the IKE protocol. Initially, responder SPI set to 0; responder fills it in
 - don't confuse with the SPI for the CHILD AH/ESP SA
- **Versions: major = 2, minor = 0**
 - ⇒ Bit V=1 says that implementation can "speak" higher version than what written in the hdr
- **Exchange type: one of 4 messages: IKE_SA_INIT, IKE_AUTH, CREATE_CHILD_SA, INFORMATIONAL**
 - ⇒ Direction: bit R (response=1, request=0); bit I (original initiator=1, original responder=0; needed to properly match SPIs)
- **Message ID: Sequence number counter, to match requests with responses, to manage retransmission, to combat replay attacks**

IKE_SA_INIT exchange

→ Clear Text Request followed by Clear Text response

⇒ Negotiates security parameters for the IKE_SA

→ Output

⇒ Generate a session key (SKEYSEED) from which all the other encryption and authentication keys will be generated

IKE_AUTH exchange

→ Authenticated and Encrypted request/response

⇒ transmit identities

⇒ prove knowledge of the secrets corresponding to the two identities

⇒ set up an SA for the first (and often only) AH and/or ESP CHILD_SA.

→ Output

⇒ SA entry created in the SAD

INFORMATIONAL exchange

→ Notification messages

→ Configuration messages

⇒ E.g. assign internal IP address to remote terminal tunneled into an IPsec SA

Topics

→ Overview of security services

⇒ Based on ISO OSI security reference model

→ How some known protocols map to the ISO OSI model?

⇒ To layers

⇒ To security model

→ IPsec

⇒ Introduction (operation modes, relation to IPv6, extension headers)

⇒ Architecture (much more than a protocol)

⇒ protocols (ESP, AH)

⇒ Management (SAD, SPD)

⇒ Signaling (IKE)

⇒ *History (RFC series)*

IPsec standardization History

Three major “series” of RFCs

→ Serie 1: RFC 1825-1827 (august 1995)

⇒ IPsec concepts first drafted

→ Serie 2: RFCs 2401-2412 (november 1998)

⇒ Significant revision of ALL the IPsec architecture

⇒ Describes IPsec as we know it today

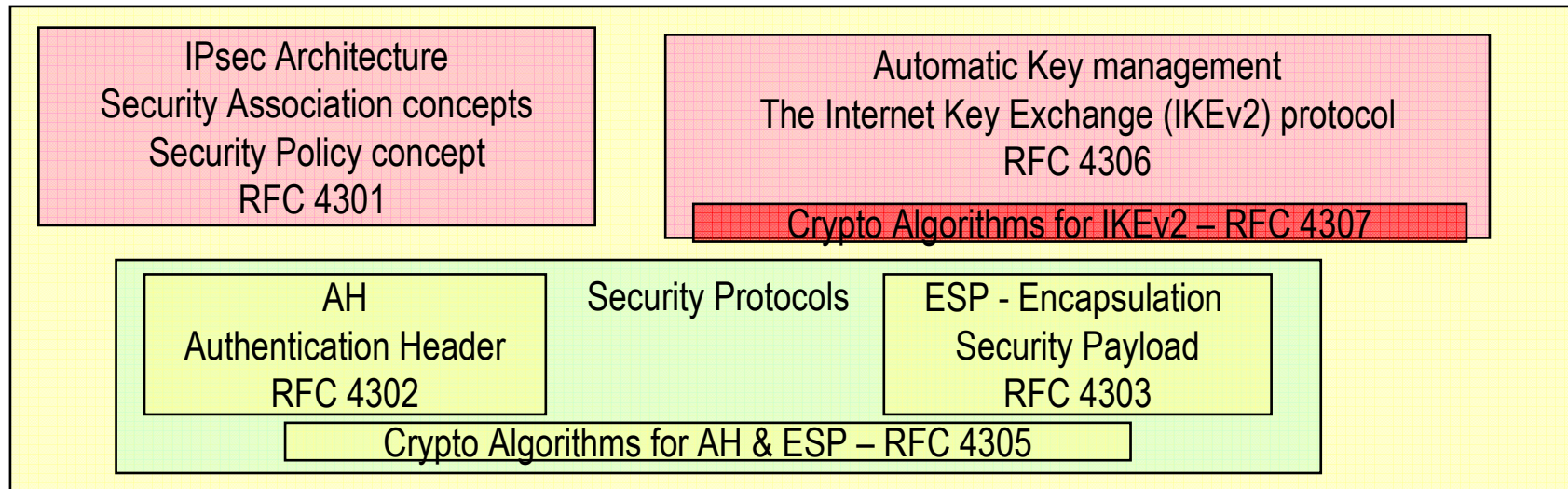
→ Serie 3: RFC 4301-4307 (december 2005)

⇒ Born after long discussion in WG (almost 5 years)

⇒ basically touches/extends all the IPsec architecture

⇒ Most important: major revision of IKE (Internet Key Exchange protocol)

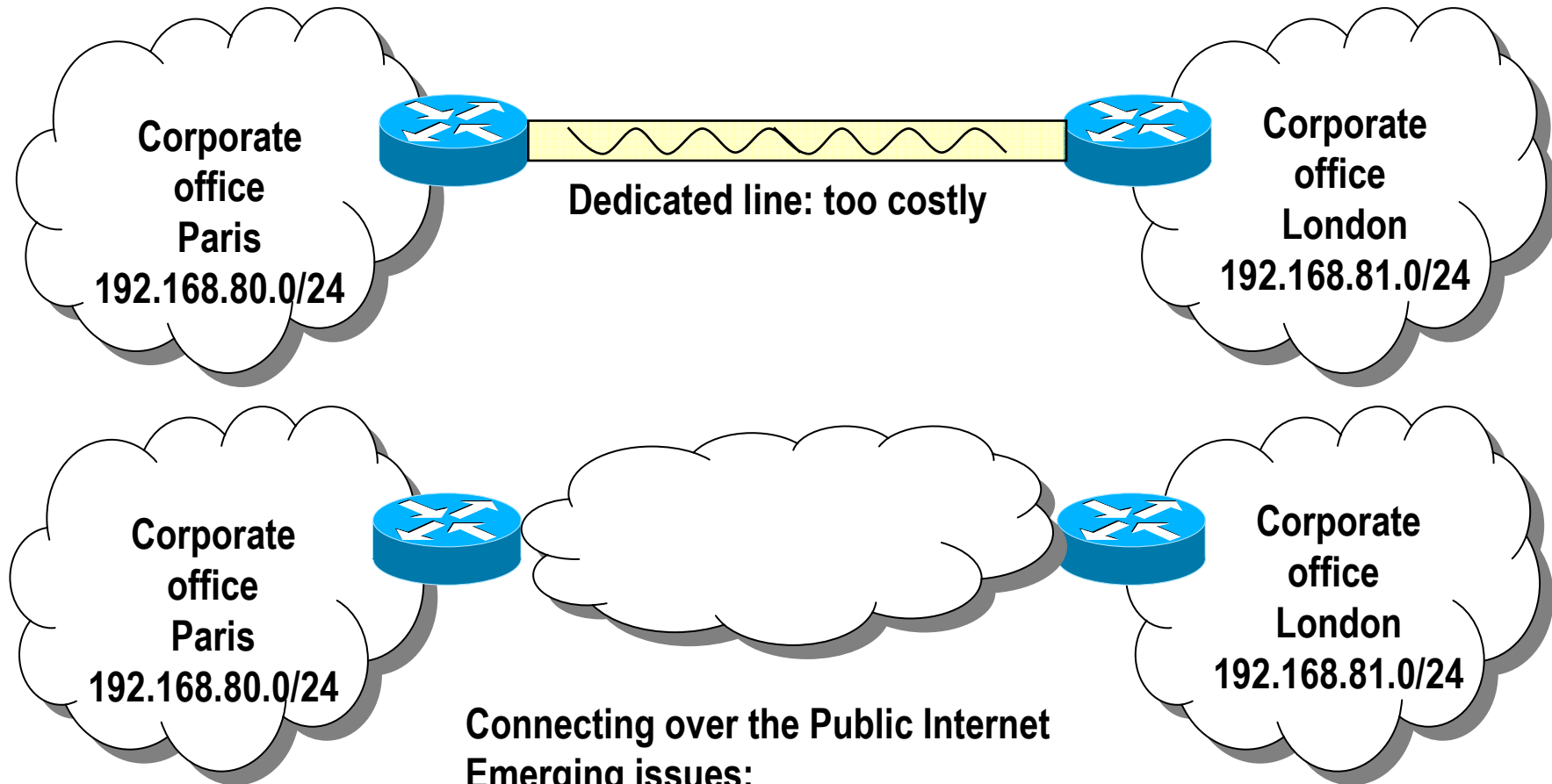
→ IKEv2 now simplifies and glues several protocols (ISAKMP, IKE, Oakley) into one



VPN

Virtual Private Network

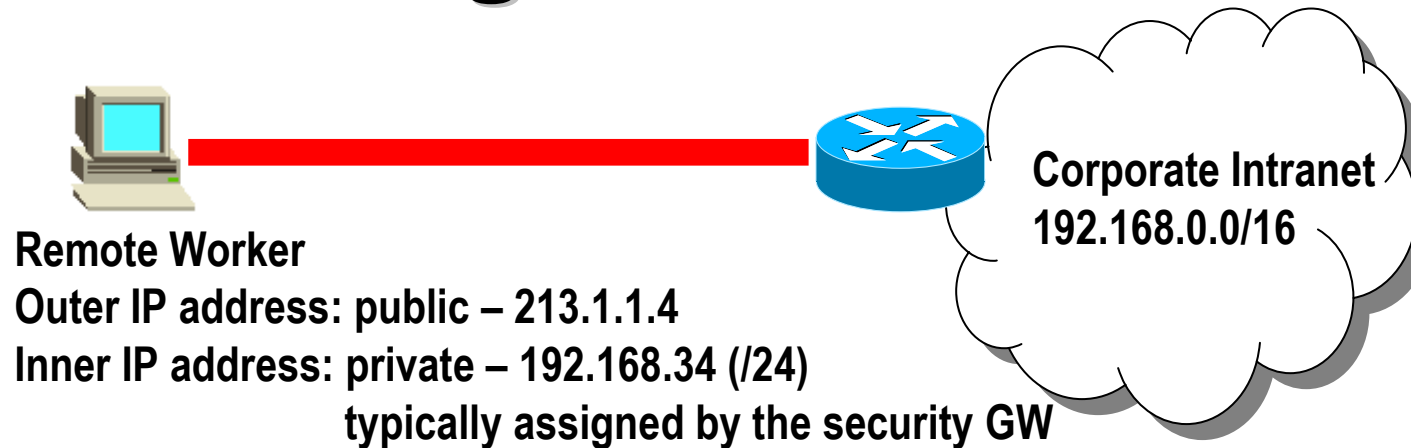
Virtual Private Networks: why?



Connecting over the Public Internet
Emerging issues:

- How to manage routing across distributed sites?
- How to protect data in transit?

Virtual Private Networks: why? host-to-gw tunnels in VPN



→ Using a private IP address inside the tunnel:

⇒ Allows to access to all services provided in the intranet, exactly like in the case the worker is connected inside the corporate

Virtual + Private Networks

→ VPN =

⇒ Virtual Networks (tunnels)

+

⇒ Private Networks (authentication, encryption)

→ IPsec: a **POSSIBLE** tool for building VPN

⇒ But IPsec and VPNs are NOT synonymous

→ as some beginner might think

→ IPsec VPNs not viable when non-IP traffic must be transported!

⇒ IPsec: not only tunnels; also e2e encrypted/authenticated transport

→ VPN alternatives:

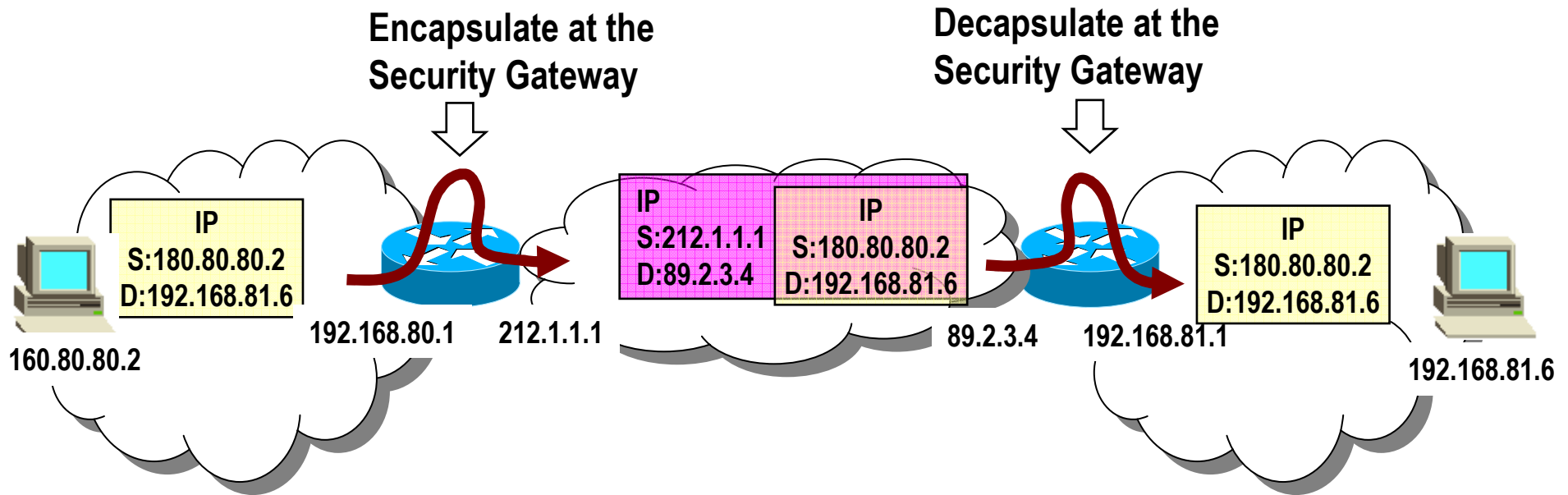
→ Layer 2: GRE/PPTP, L2TP

→ Layer 3 (actually 3-): MPLS

→ Layer 4 (actually between 4 and 7): SSL tunnels

→ Layer 7: SSH tunnels

Virtual Networks over IP



→ Security Gateway

⇒ Routing functionality

→ Based on routing table entries

⇒ En/decapsulation functionality

→ Based on Security Policy database

Trying IPsec: StrongSwan virtual laboratories

<http://www.strongswan.org/uml/>