IPSec and VPN

Diana Gratiela Berbecaru
diana.berbecaru@polito.it

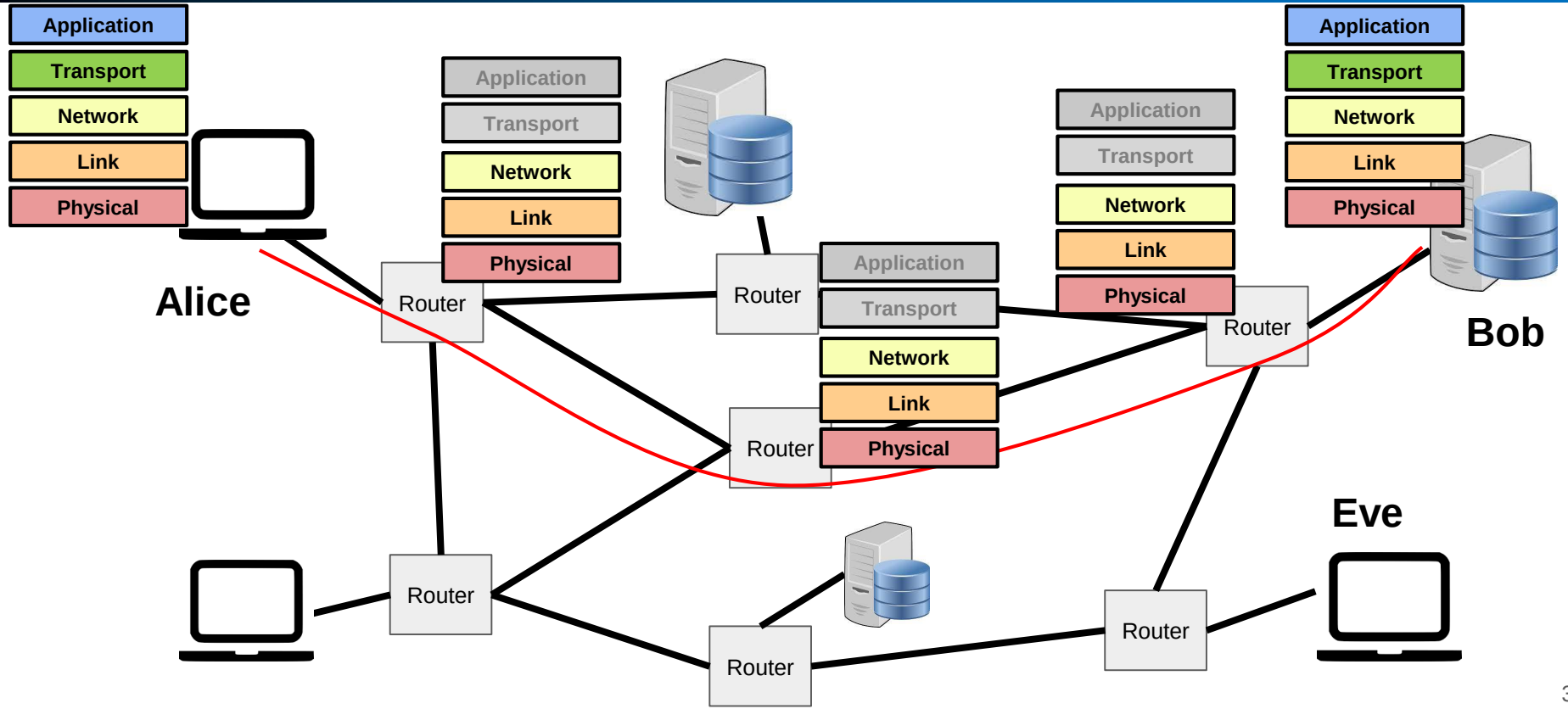
Politecnico di Torino
Dip. Automatica e Informatica

AY. 2023 - 2024

Acknowledgment

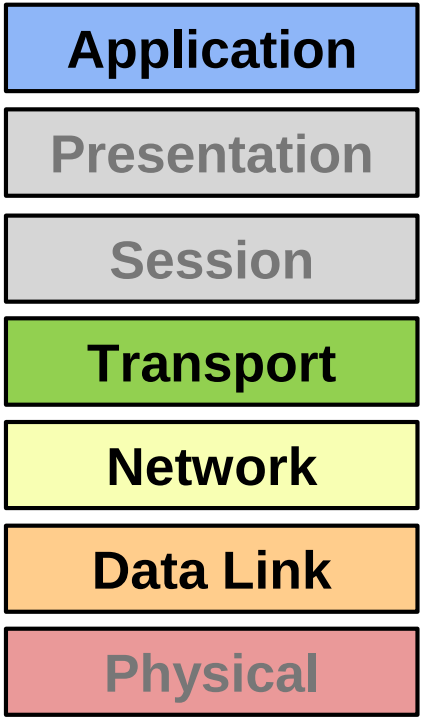
- **Slides derived from the material prepared by Prof. Antonio Lioy for the course Information Systems Security (2005 - 2022)**
- **… so this set of slides is entirely compatible with the course of the previous year(s)**

Why IPSec (and VPN)?



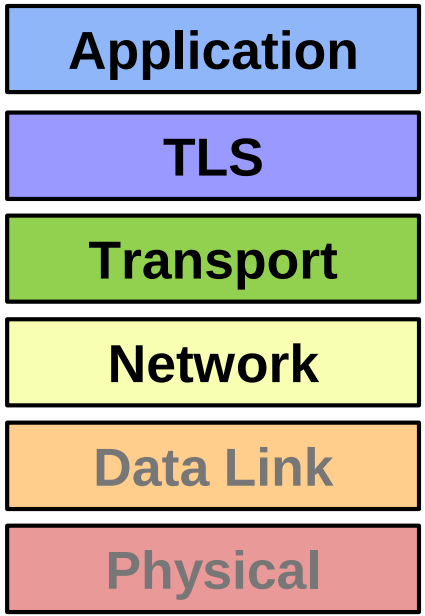
At which (OSI) layer should we add security?

- **which layer is the most appropriate to implement security services?**
- **different layers provide different services, and it might be necessary to add security services at different layers**
  - the lower we go in the stack, the more quickly we can "expel" the intruders … but the fewer the data for the decision (e.g. only the MAC or IP addresses, no user identification, no commands)
  - the upper we go in the stack, the more specific are the security functions (e.g. it's possible to identify the user, commands, data) and independent from the underlying network … but application(s) and/or OS must be changed to support security



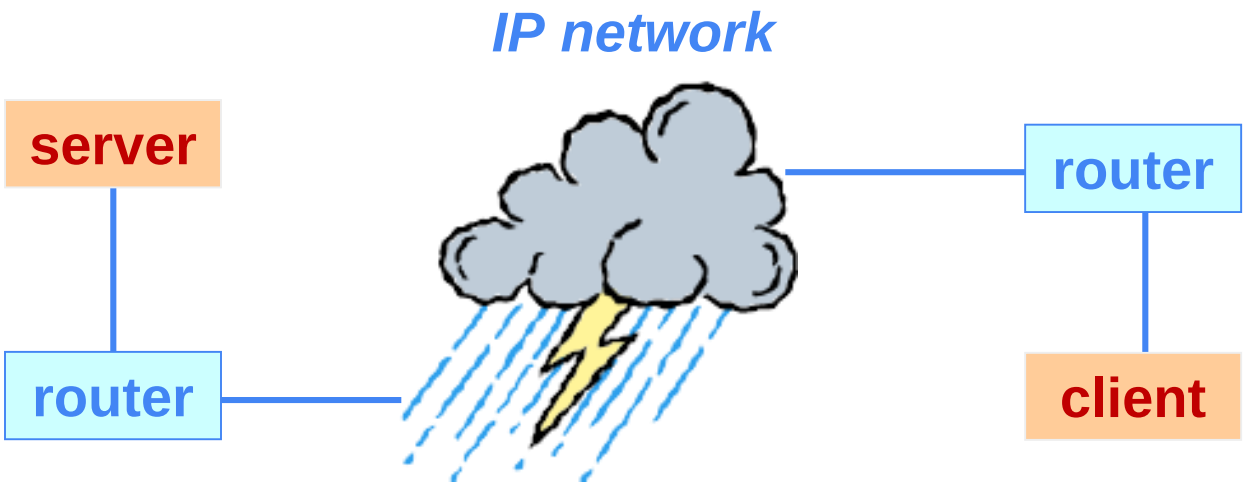
High-level view of applying security

- **Application layer security**
  - implemented by each application, e.g. secure e-mail (S/MIME, PGP)
  - disadvantages:
    - change each application and/or operating system
    - (sometimes) hard to implement, error prone
    - no protection for headers, and lower-level data
- **Transport Layer Security**
  - widely available, (quite) easy to implement and use
  - disadvantages:
    - protects only if used - requires changing each application
    - protects only end-to-end communication
    - headers are exposed



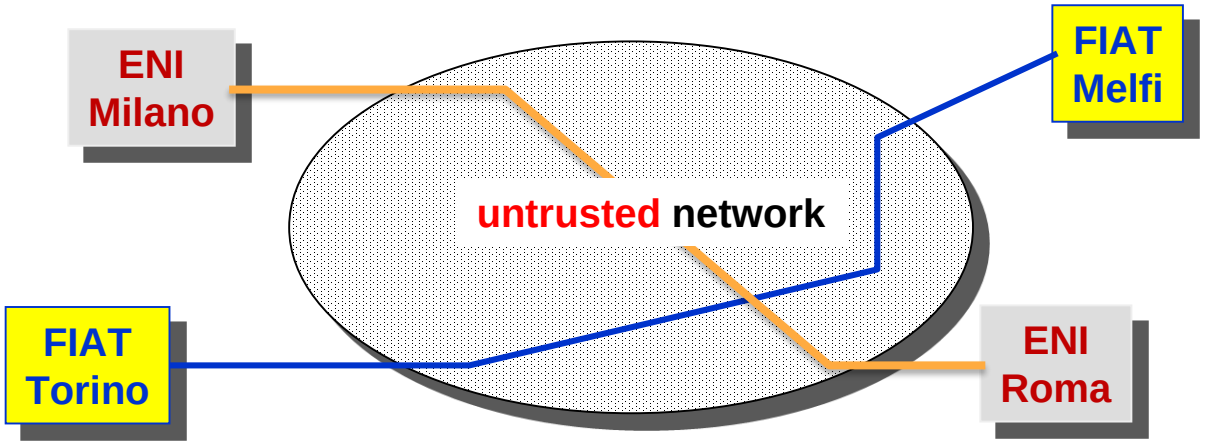
Security at network level (L3)

- **end-to-end protection for L3-homogeneous networks (e.g. IP networks)**
- **creation of VPN (Virtual Private Network)**



What is a VPN?

- a technique (hardware and/or software) to create a private network ...
- ... while using shared (or anyway untrusted) channels and transmission devices



Why do we need techniques for VPN?

- **because IP (Internet protocol) was not born with the security in mind …**
  - protocol is connectionless
  - protocol is unreliable (packets may be lost or replayed)
  - protocol does not guarantee order of the packets
  - IP packets can be sniffed
  - IP addresses can be spoofed

Techniques to create a VPN

- **via private addressing**
- **via protected routing (IP tunnel)**
- **via cryptographic protection of the network packets (secure IP tunnel)**

1. VPN via private IP addresses

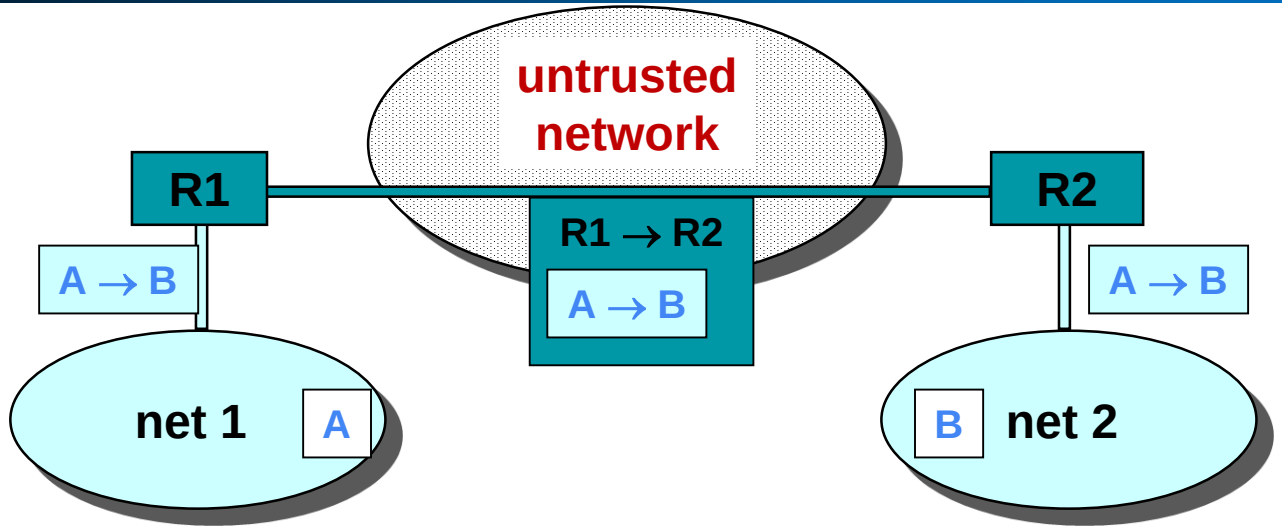
- **the networks to be part of the VPN use non-public IP addresses so that they are unreachable from other networks (e.g. private IANA networks as per RFC-1918)**

- **this protection can be easily defeated if an attacker:**
  - guesses or discovers the addresses
  - can sniff the packets during transmission
  - has access to the communication devices (routers)

2. VPN via tunnel

- **the routers encapsulate the entire L3 packets as a payload inside another packet**
  - IP in IP
  - IP over MPLS (Multiprotocol Label Switching)
  - other (e.g. IP over TLS)
- **the routers perform access control to the VPN by ACL (Access Control List)**
- **this protection can be defeated by anybody that manages a router or can sniff the packets during transmission**

VPN via IP tunnel



IPv4 header (end-to-end)	TCP/UDP header + data

IPv4 header (tunnel)	IPv4 header (end-to-end)	TCP/UDP header + data

IP tunnel: fragmentation

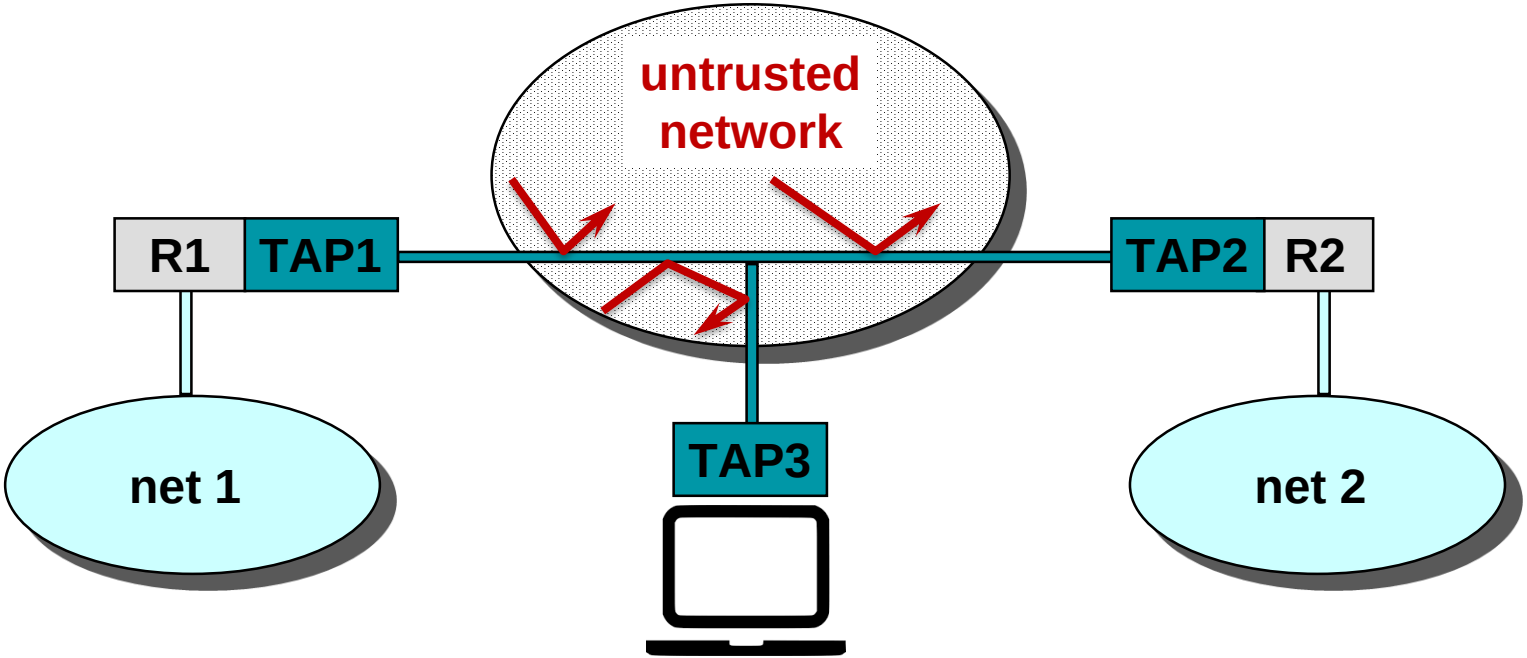
- **if the packet has size equal to the MTU, then encapsulation will only possible with fragmentation**
- **maximum performance loss = 50%**
- **largest loss for applications with large packets (typically the non-interactive applications, e.g. file transfer)**

3. VPN via secure IP tunnel

- **before encapsulation, the packets are protected with:**
  - MAC (integrity + authentication)
  - encryption (confidentiality)
  - numbering (to avoid replay attacks)

- **if the cryptographic algorithms are strong, and the keys are secure then the only possible attack is to stop the communications**
- **also known as S-VPN (Secure VPN)**

VPN via secure IP tunnel



IPsec

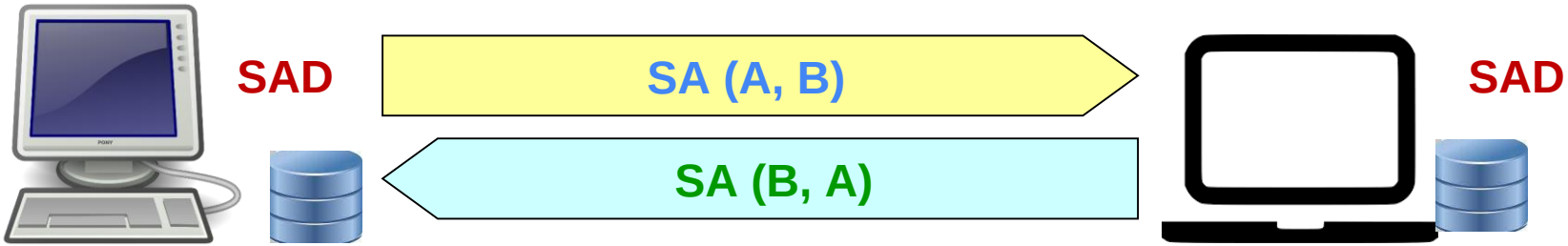
- **IETF architecture for L3 security in IPv4 / IPv6:**
  - to create S-VPN over untrusted networks
  - to create end-to-end secure packet flows
- **definition of one architecture and two specific packet types:**
  - Arch – IPSec architecture (RFC 2401, defined in 1998)
  - AH (Authentication Header) (RFC 2402, defined in 1998)
    - for integrity, authentication, no replay
  - ESP (Encapsulating Security Payload) (RFC 2406, 1998)
    - for confidentiality (+AH)
- **protocol for key exchange:**
  - IKE (Internet Key Exchange) (RFC 2409, 1998)

IPsec security services

- **authentication of IP packets:**
  - data integrity and data (source) authentication
  - (partial) protection against “replay” attacks
- **confidentiality of IP packets:**
  - data encryption
- **Notes:**
  - IP packets that undergo protection by IPsec are still standard IP packets at the outmost level
  - algorithms, keys and other security parameters of IPSec are negotiated ahead of time between sender and receiver of IP stream (manually, or by using IKE protocol)

IPsec Security Association (SA)

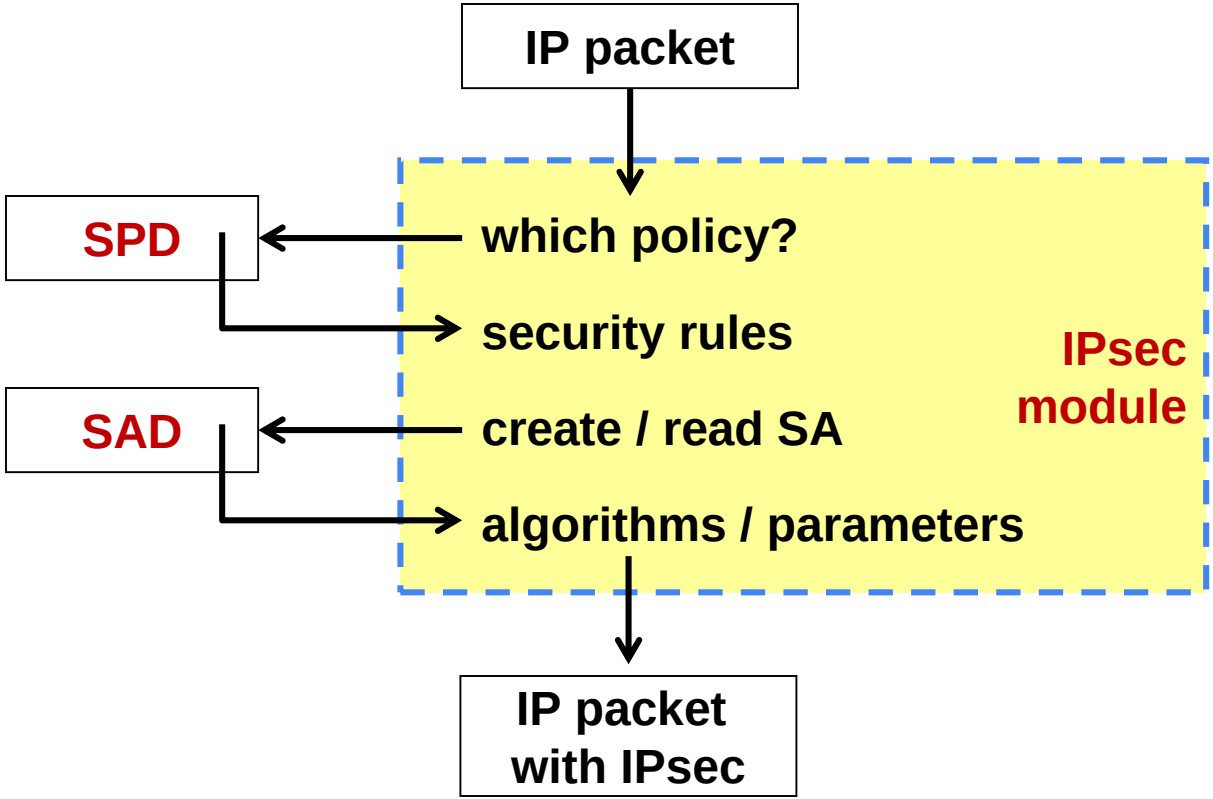
- the security properties of the secure IPSec channel created between the sender and the receiver are stored in a SA structure
- the SA is unidirectional, it contains all the (security) data necessary for IPsec processing of outgoing and incoming packets
- an IPSec device stores all active SAs it currently has in a database, called SAD (logically divided into outgoing and incoming)
- each SA may have (associated) different security services
- two SA are needed to get complete protection of a bidirectional packet flow



IPsec local database(s)

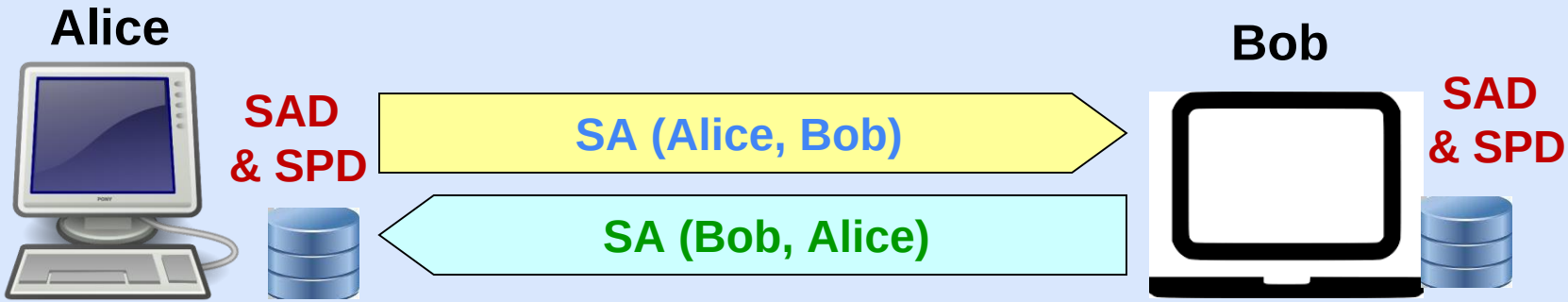
- **SAD (SA Database)**
  - list of active SAs and their characteristics (algorithms, keys, parameters)
- **SPD (Security Policy Database)**
  - a-priori configured (e.g. manually) or connected to an automatic system (e.g. ISPS, Internet Security Policy System)
  - list of security rules to apply to the different packet flows, set by user. Each rule contains a set of selectors and an action. Rules are scanned by their order and a packet is processed according to the first match
    - Selectors: can be IP addresses, IP address ranges, TCP/UDP ports,..
    - An action, can be: Discard, Bypass IPSec, or apply IPSec (in this case a pointer to the entry SAD containing an SA is typically implemented)

How IPsec processing works (sending)



IPsec Security Association (SA): Example

- **Alice and Bob communicate securely using IPSec**
- **Alice's outgoing SAD and Bob's incoming SAD contain a SA that is used for traffic from Alice to Bob**
- **Bob's outgoing SAD and Alice's incoming SAD contain an SA that is used for traffic sent from Bob to Alice**



IPsec Security Association (SA): Example

Alice Outgoing SAD

Selectors	SA
…	
Alice, Bob	SPI=23, Alice, Bob, DES3, HMAC-SHA256, xxx..

Bob Incoming SAD

SPI	SA
…	
23	Alice, Bob, DES3, HMAC-SHA256, xxx..

IPsec Security Association (SA): Example

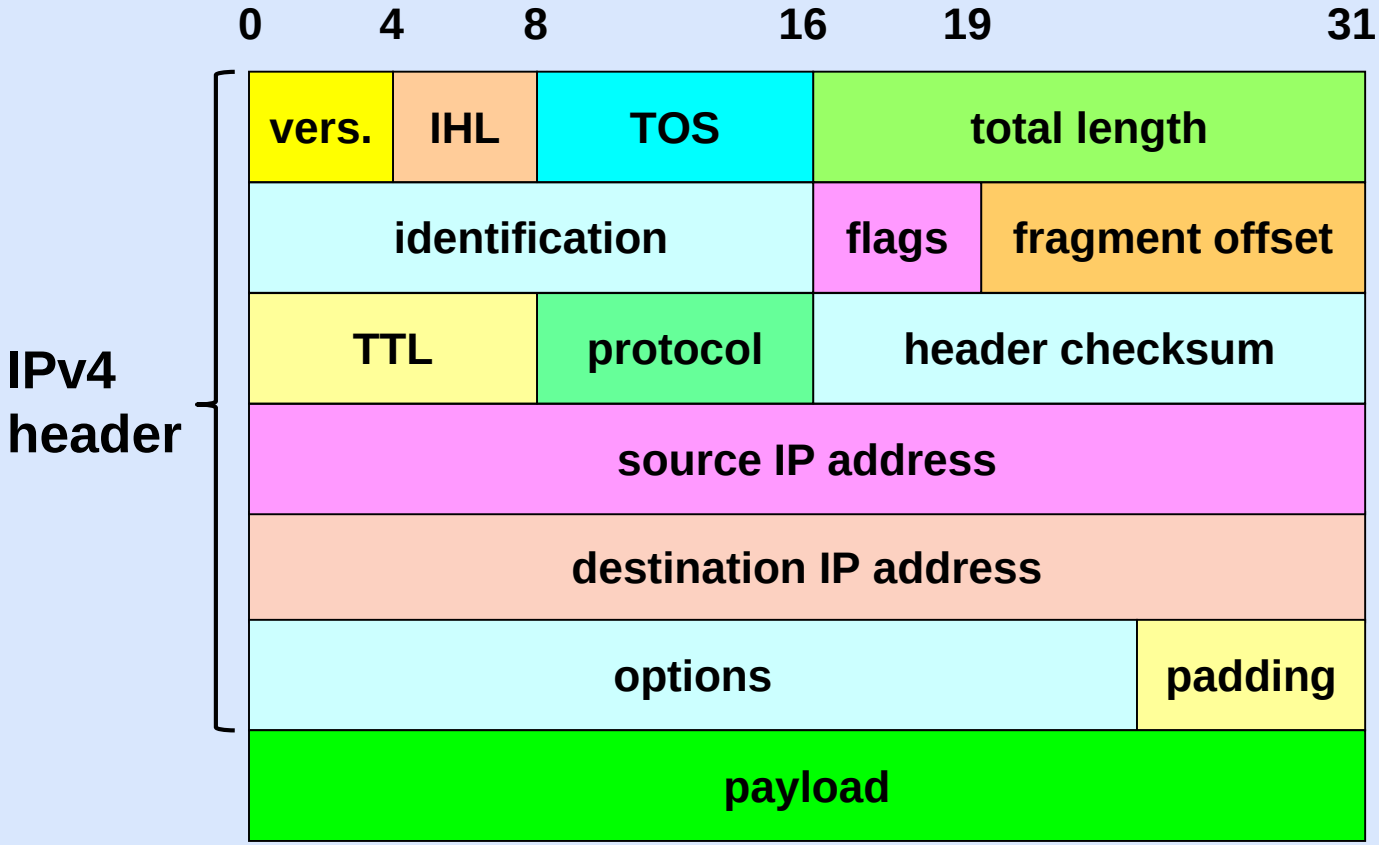
Alice Incoming SAD

SPI	SA
…	
14	Bob, Alice, AES, HMAC-SHA512, yyyy..

Bob Outgoing SAD

Selectors	SA
Bob, Eve	
Bob, any, HTTP	
…	
Bob, Alice	SPI=14, Bob, Alice, AES, HMAC-SHA512, yyyy…

IPv4 packet


0
4
8
16
19
31
vers.
IHL
TOS
total length
identification
flags
fragment offset
TTL
protocol
header checksum
source IP address
destination IP address
options
padding
payload
IPv4 header

IPv4 header fields

- **Vers: version number**
- **IHL (Internet Header Length) in 32-bit words**
  - length of IP header (in units of 32 bits)
- **TOS (Type Of Service): nearly ever used (!), used by routers and gateways for quality of service**
- **length: total no. of bytes (header and payload) of this IP packet**
- **identification: ID number of this packet (to be used by all fragments of this packet)**
- **flags: may/don't fragment, last/more fragments**
- **TTL (Time To Live): max number of hops this packet can make before it is discarded**

IPv4 header fields (II)

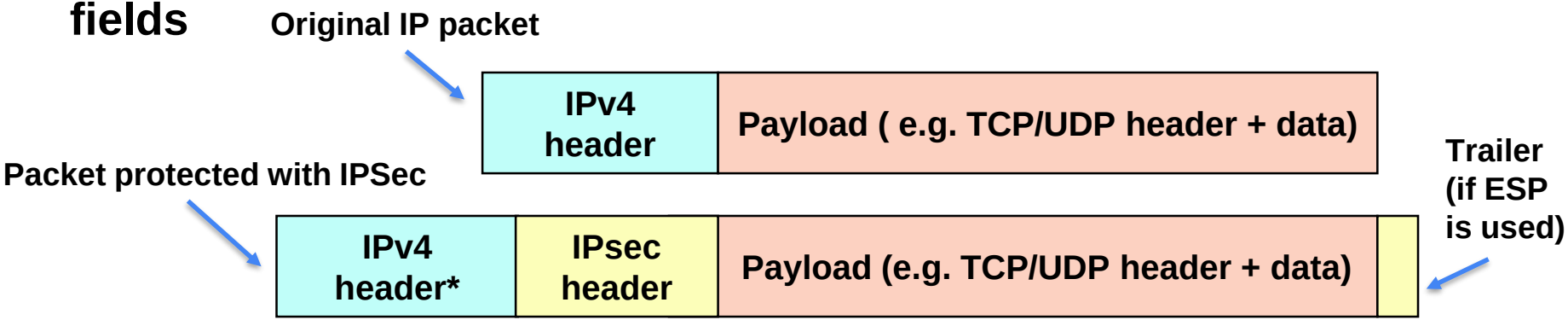
- protocol: protocol of the payload (i.e. which type of upper layer protocol is encapsulated in this IP packet)
- header checksum: non-cryptographic checksum of the IP packet header
- source IP address (32 bit) of the source
- destination IP address (32 bit) of the destination
- TTL (Time To Live): max number of hops this packet can make before it is discarded
- options: extensions of the IP header with features such as ‘source routing’
- padding: to align packet size to a multiple of 32 bits
- payload: the data contained in this IP packet, typically a packet of another protocol (TCP, UDP, IP, ICMP, etc..)

IPSec modes of operation

- **Transport mode:**
  - End-to-end: IPSec encapsulation is done by the source (sender) of the original IP packet, and the de-encapuslation is done by the destination (receiver) of the IP packet
- **Tunnel model:**
  - GW-to-GW: IPSec encapsulation and de-encapsulation is done by gateways along the route between the source (Sender) and destination (receiver)

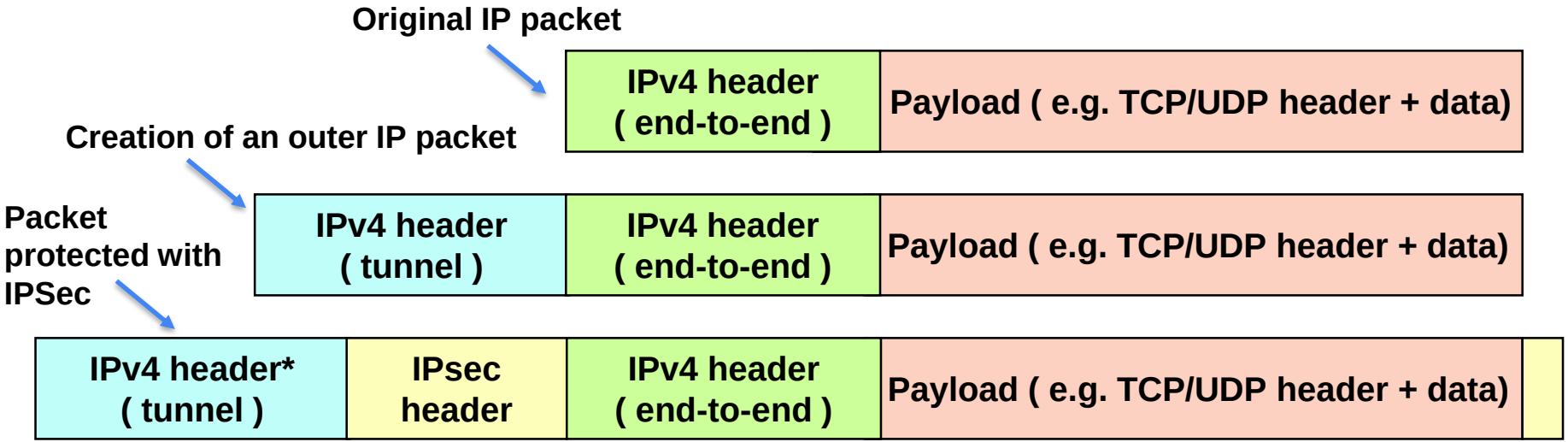
Transport mode IPsec

- **used for end-to-end security, that is used by hosts, not gateways (exception: traffic for the gateway itself, e.g. SNMP, ICMP)**
- **IPSec header (e.g. AH) is inserted between the original IP header and the original IP payload**
  - if IPSec header is ESP, a trailer is placed after the original IP payload
- **pro: computationally light ; con: no protection of header variable fields**



Tunnel mode IPsec

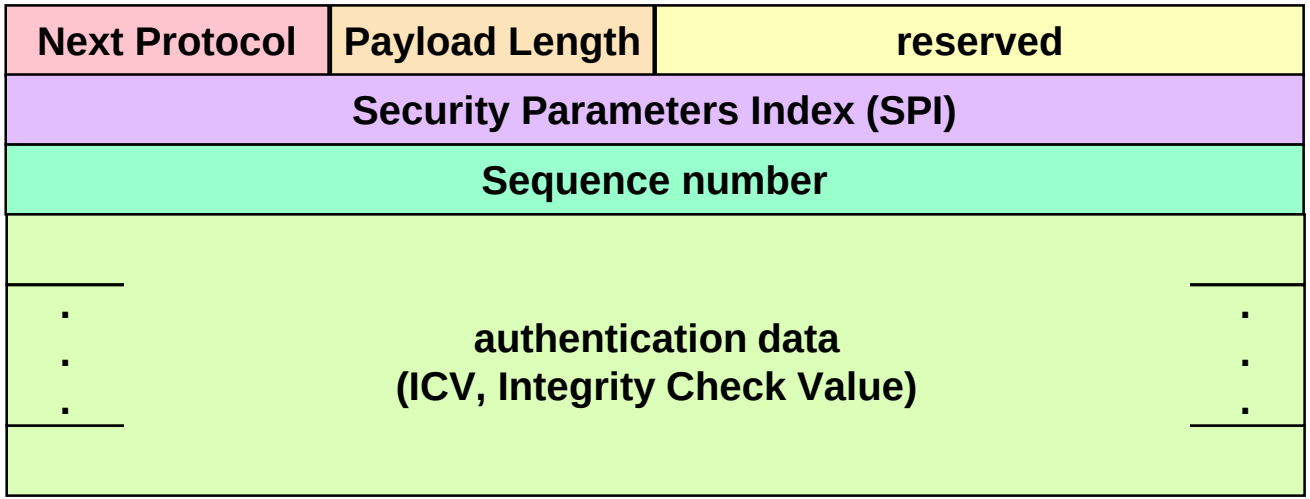
- used to create a VPN, (usually) by gateways
- pro: protection of header variable fields
- con: computationally heavy



AH

- **Authentication Header**
- **mechanism (first version, RFC-1826):**
  - data integrity and sender authentication
  - compulsory support of keyed-MD5 (RFC-1828)
  - optional support of keyed-SHA-1 (RFC-1852)
- **mechanism (second version, RFC-2402):**
  - data integrity, sender authentication and (partial) protection from replay attack
  - HMAC-MD5-96
  - HMAC-SHA-1-96

AH - format (RFC-4302)



AH format – description of fields

- **Next Protocol: value that indicates the protocol contained in the AH packet**
- **Payload Length: length of the AH header in 32-bit words (-2)**
- **SPI: Security Parameter Index, points to the SA in the receiver's SA database**
- **Sequence number: 32-bit counter used to prevent replay attacks**
- **authentication data (or ICV): result of the MAC (the length depends on the authentication algorithm)**

AH authentication

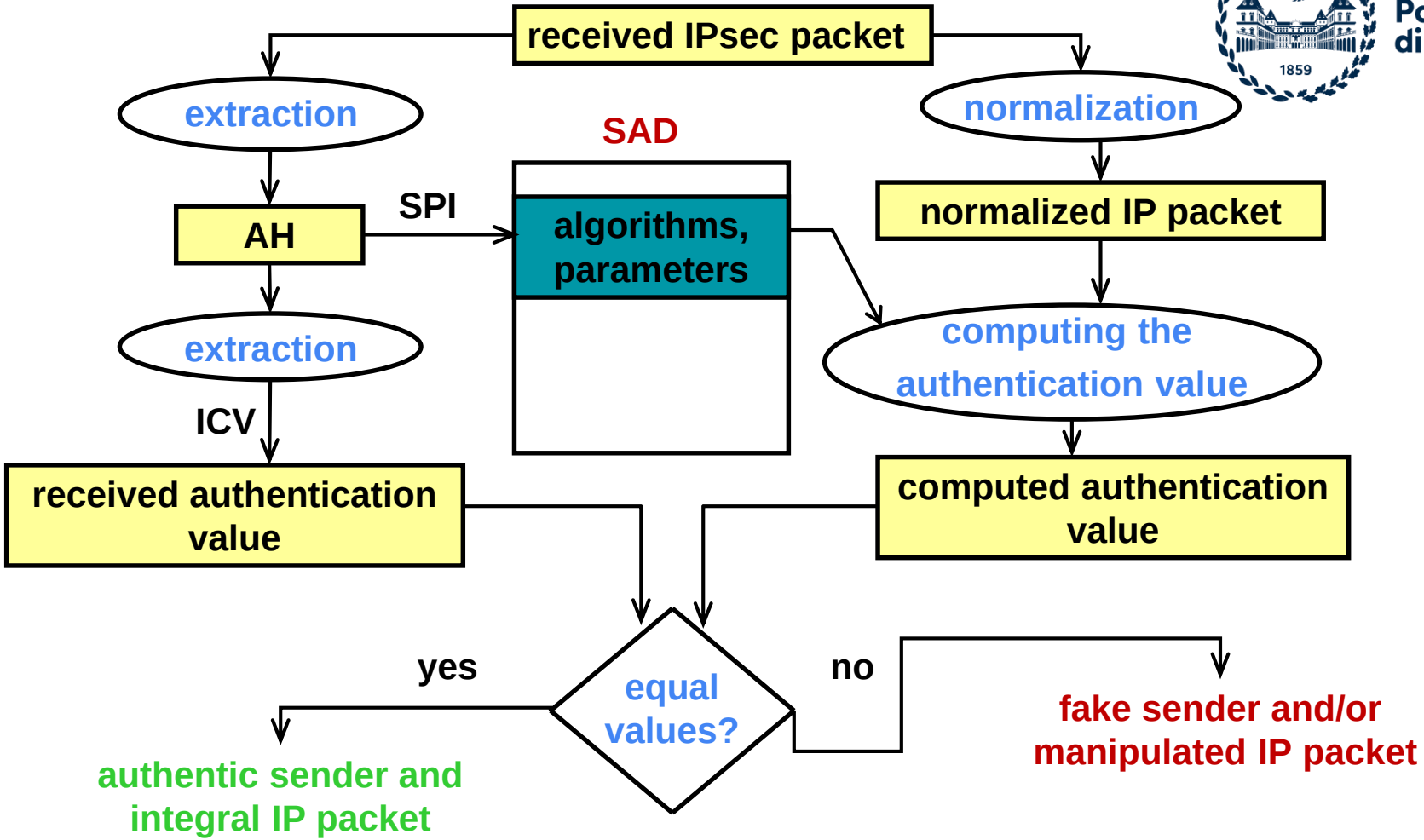
- **the authentication data (MAC) is computed over the all fixed fields in the IP packet (including fields in the IP header, fields in the AH header, and the payload, e.g. TCP/UDP header + data)**
- **normalization: before the MAC calculation, all the unpredictable or mutable fields (e.g. TTL) in the IP header are set to 0, while all the predictable fields (e.g. source routing) are set to their arrival value**
- **before the MAC calculation, the 'authentication data' field in the AH header is set to 0**

AH encapsulation

- **operations performed by the IPsec module (at sender):**
  - locate the outgoing SA in the SAD
    - if the relevant SA is not found – drop the IP packet
  - set the SPI in the AH header
  - increment the sequence number in the outgoing SA, and put the new sequence number in the AH header
  - calculate the authentication data (ICV)

AH decapsulation

- **operations performed by the IPsec module (at receiver):**
  - locate the incoming SA (according to the SPI)
    - if there is no SA that matches the SPI – drop the IP packet
  - check if the sequence number is correct
    - If sequence number is not correct – drop the IP packet
  - calculate the authentication data, and compare it against the authentication data in the packet
    - if authentication data does not match – drop the IP packet


received IPsec packet
extraction
AH
extraction
ICV
received authentication value
SPI
SAD
algorithms, parameters
normalization
normalized IP packet
computing the authentication value
computed authentication value
equal values?
yes
authentic sender and integral IP packet
no
fake sender and/or manipulated IP packet

Calculating authentication data: HMAC-SHA1-96

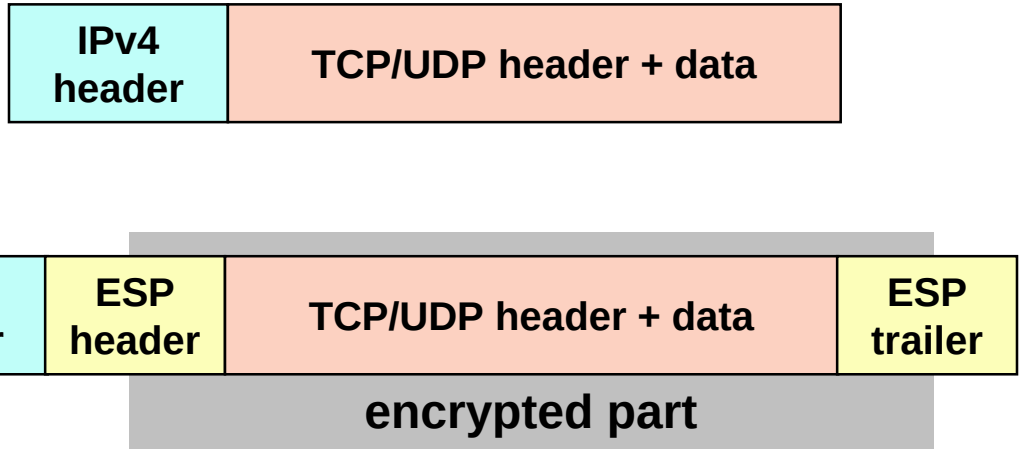
- given M normalize it to generate M’
- pad M’ to a multiple of 160 bit (by adding 0x00 bytes) to generate M’p
- compute the authentication base:
  B = HMAC-SHA1 (K, M’p)
- ICV = 96 leftmost bits of B

ESP

- **Encapsulating Security Payload**
- **first version (RFC-1827) gave only confidentiality**
- **base mechanism: DES-CBC (RFC-1829)**
- **other mechanisms possible**
- **second version (RFC-2406):**
  - provides also authentication (but the IP header, so the coverage is not equivalent to that of AH)
  - the packet dimension is reduced and one SA is saved

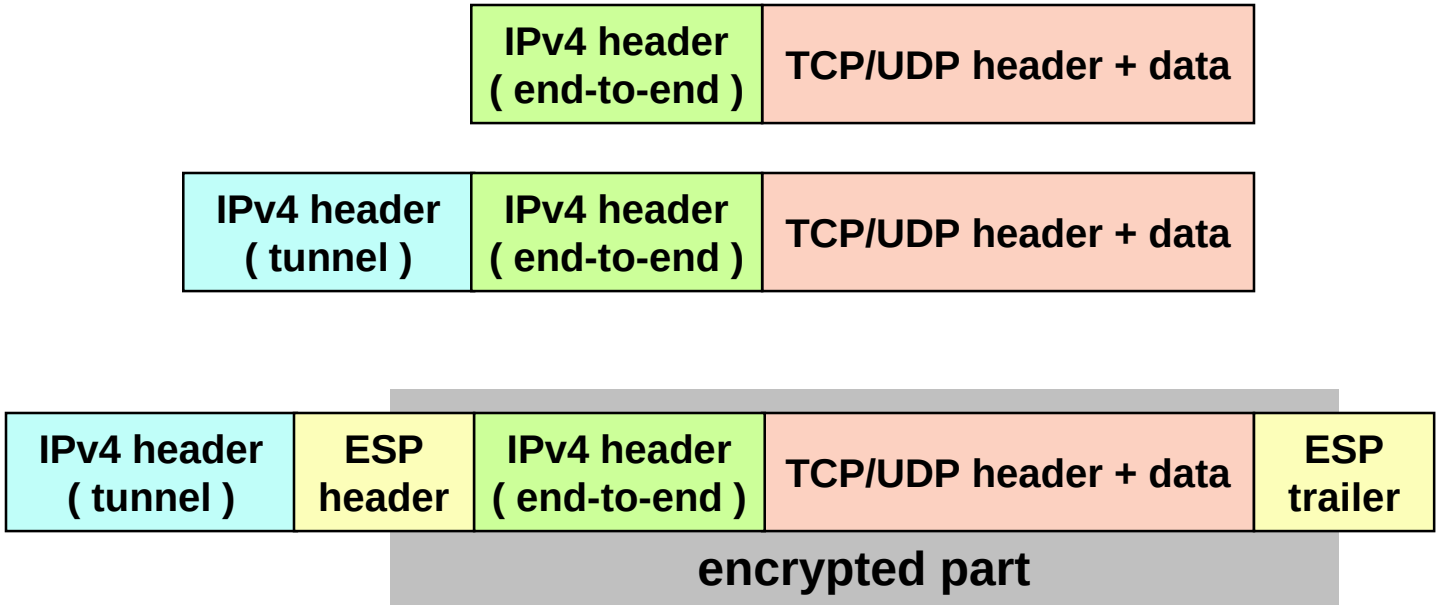
ESP in transport mode

- **pro: the payload is hidden (including info needed for QoS, filtering, or intrusion detection!)**
- **con: the header remains in clear**

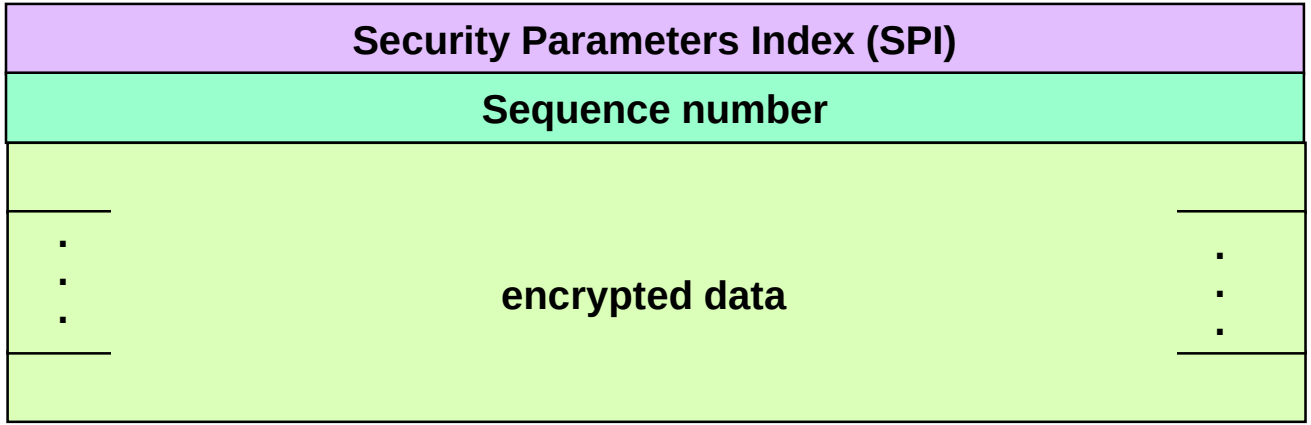


ESP in tunnel mode

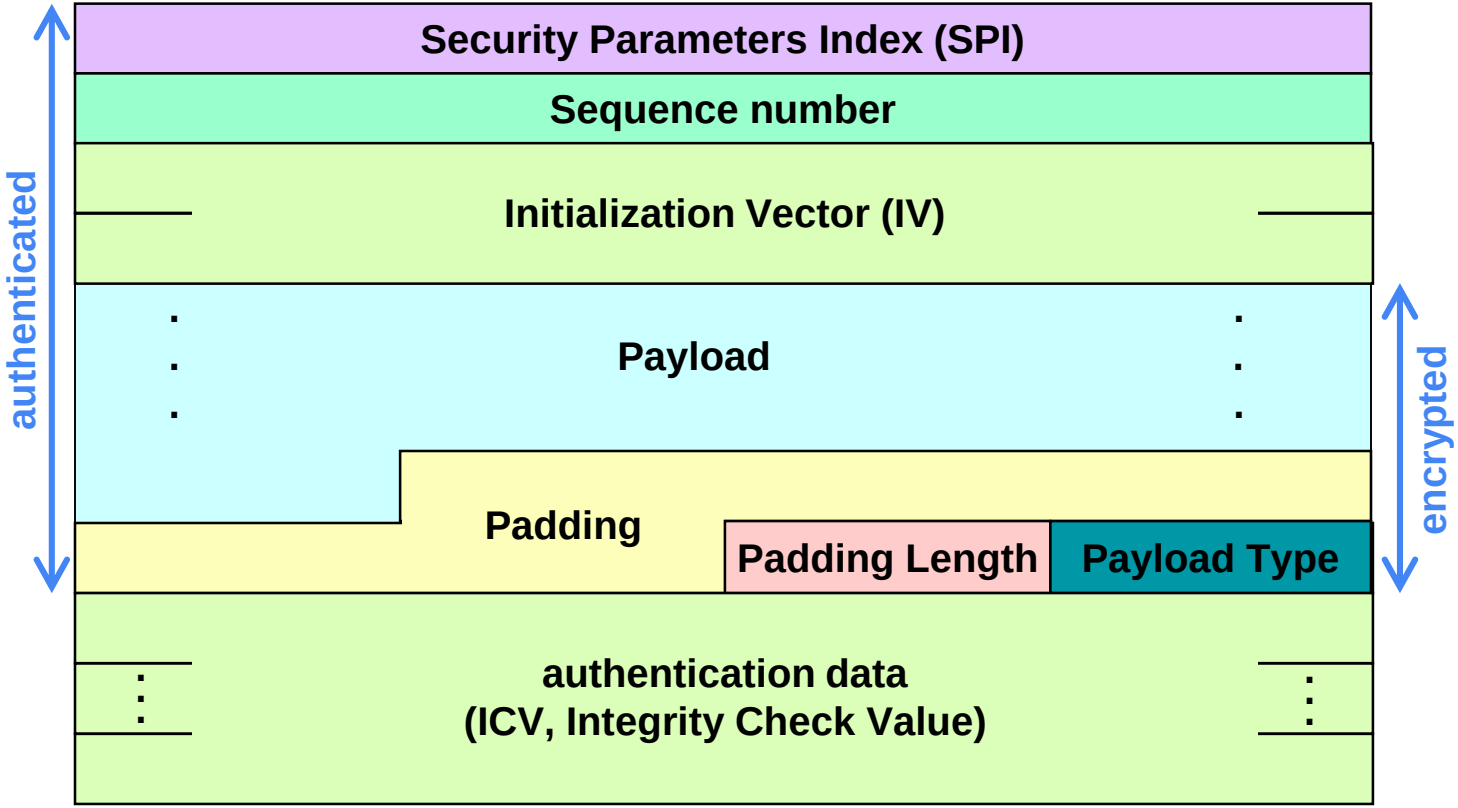
- **pro: hides both the payload and (original) header**
- **con: larger packet size**



ESP - format (RFC-2406)



ESP-DES-CBC - format (RFC-4306)



IPsec implementation details

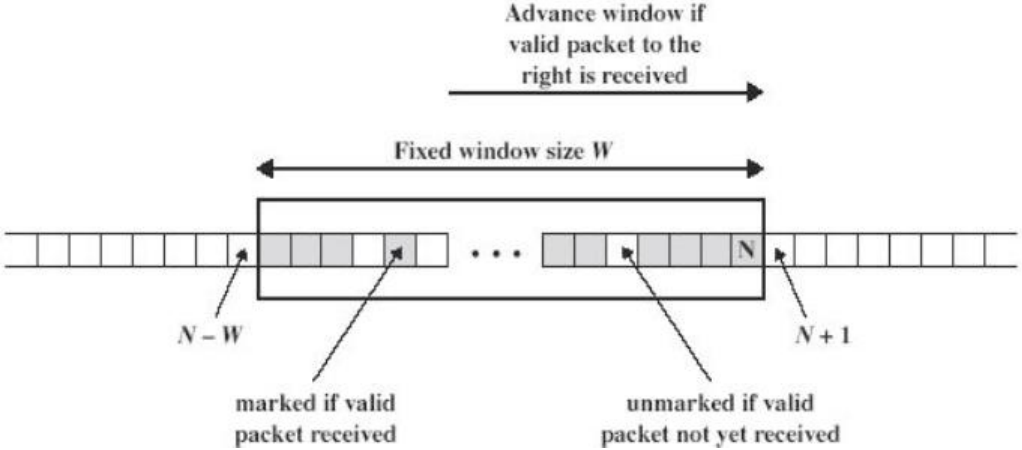
- **UI crypto-suites (RFC-4308) for interoperability**
  - VPN-A = ESP/3DES-CBC/HMAC-SHA1-96
  - VPN-B = ESP/AES-128-CBC/AES-XCBC-MAC-96
- **NULL algorithms for ESP:**
  - for authentication or confidentiality (but not simultaneously!)
  - protection vs. performance trade-off
- **sequence number:**
  - (partial) protection from replay
  - minimum window of 32 packets (64 suggested)

IPsec replay protection

- **when a SA is created, sender initializes the sequence number to 0**
- **when sending a packet, increment the sequence number per outgoing packet**
- **sequence number cannot be modified by an attacker (is protected by the MAC calculated for AH or by ESP)**
- **when the sequence number 2^{32}-1 is reached, a new SA should be negotiated**
  - AH keys must be changed after less than 2**32 packets

IPsec replay protection (II)

- **the receiver extracts the sequence number from the IPSec header and checks if he has already encountered this number**
- **to avoid dropping legal packets arriving out of order, the receiver maintains in the SA a sliding window (minimal size 32)**
  - moving window: outside it, no replay protection



IPsec v3

- **AH is optional, ESP mandatory**
- **support for single source multicast**
- **ESN (Extended Sequence Number):**
  - 64 bit (but only the 32 least significant ones are transmitted)
  - default when using IKEv2
- **support for authenticated encryption (AEAD)**
- **clarifications about SA and SPI (for faster lookup)**

IPsec v3 – algorithms (RFC-4305)

- **for integrity and authentication:**
  - (MAY) HMAC-MD5-96
  - (MUST) HMAC-SHA-1-96
  - (SHOULD+) AES-XCBC-MAC-96
  - (MUST) NULL (only for ESP)
- **for confidentiality:**
  - (MUST) NULL
  - (MUST–) TripleDES-CBC
  - (SHOULD+) AES-128-CBC
  - (SHOULD) AES-CTR
  - (SHOULD NOT) DES-CBC

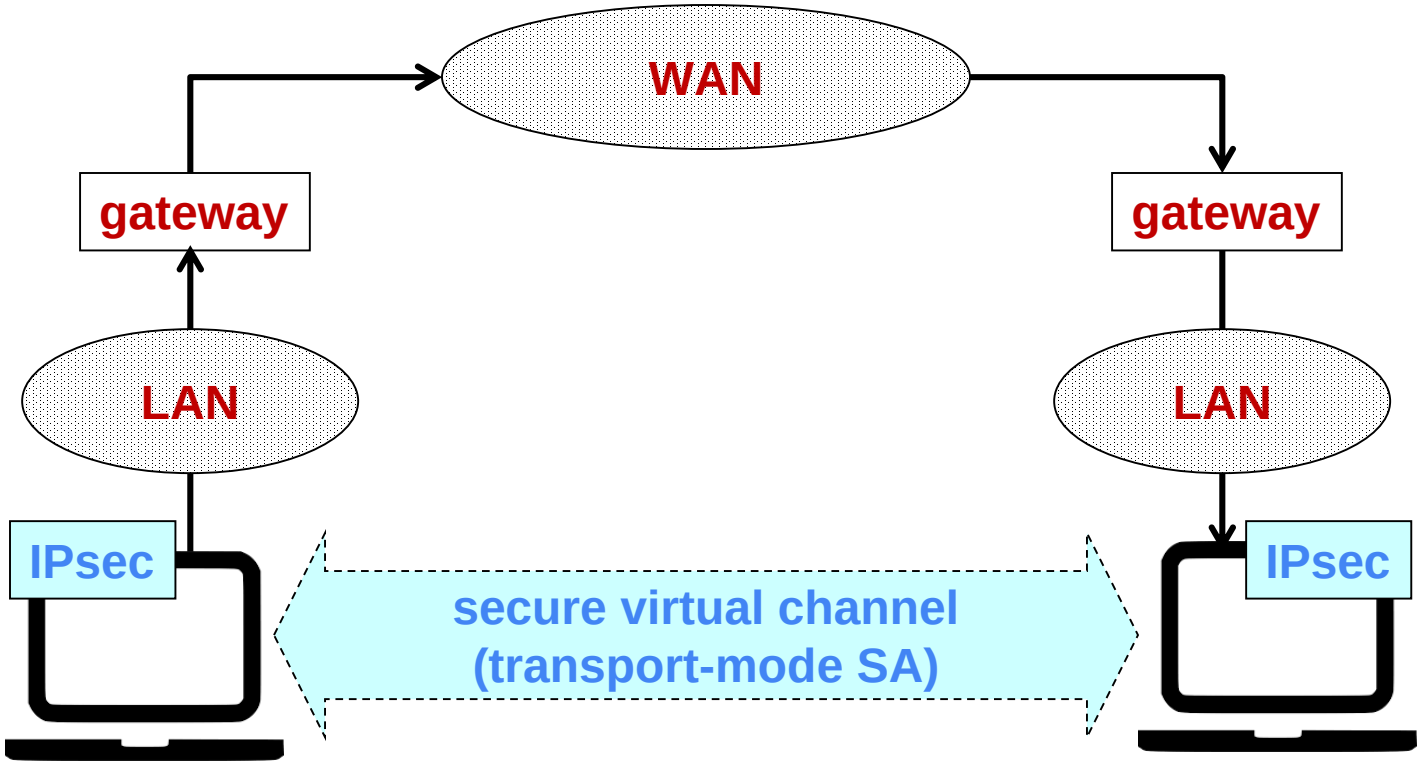
IPsec v3 – other algorithms

- **for authenticated encryption (AEAD mode):**
  - AES-CCM
  - AES-CMAC
  - ChaCha20 w/ Poly1305
- **for authentication and integrity:**
  - HMAC-SHA-256-128
  - HMAC-SHA-384-192
  - HMAC-SHA-512-256

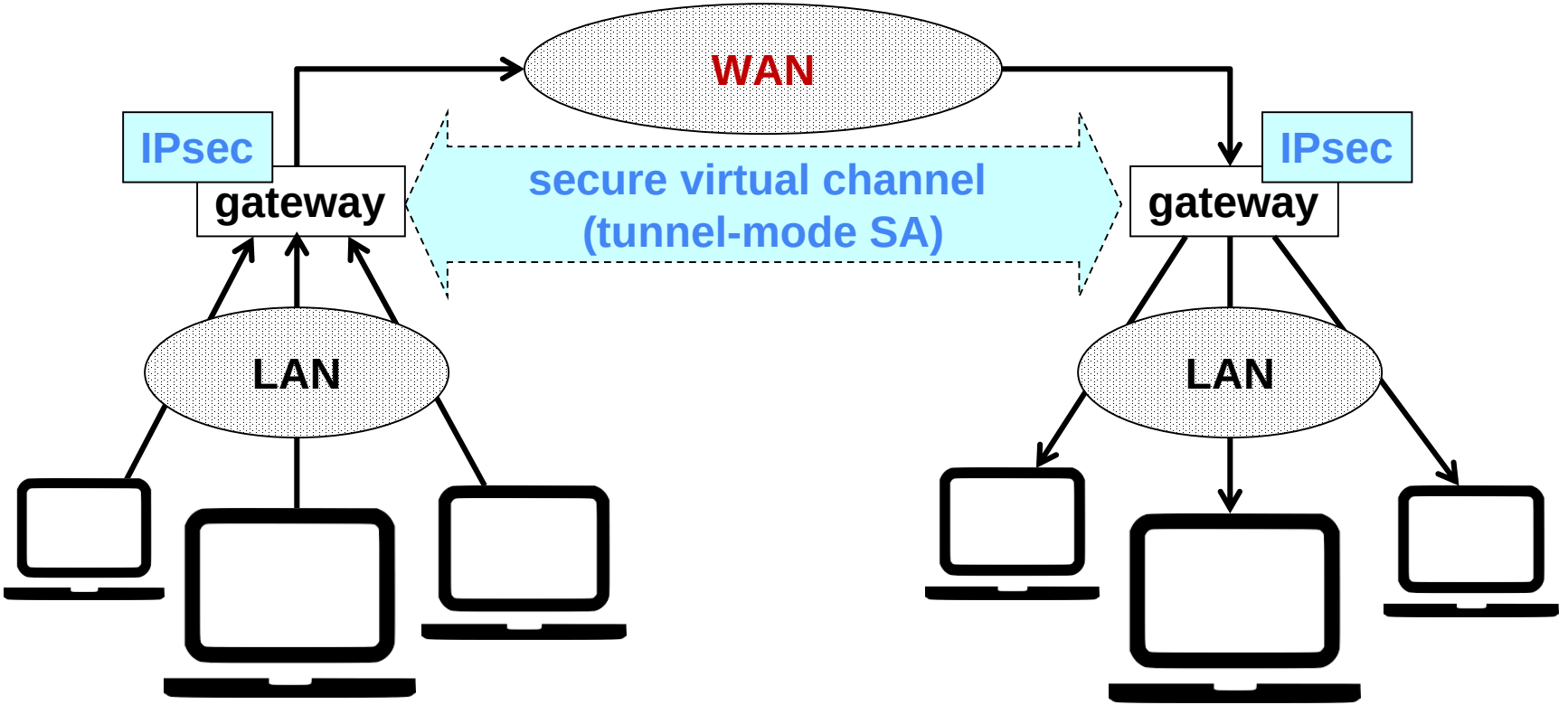
IPsec v3 – TFC

- **TFC (Traffic Flow Confidentiality) padding in ESP**
  - after the payload and before the normal padding
  - the receiver must be able to compute the original size of the payload (e.g. possible with IP, UDP, and ICMP payloads)
- **support for "dummy packets" (next header 59)**
  - useful only if encrypted …

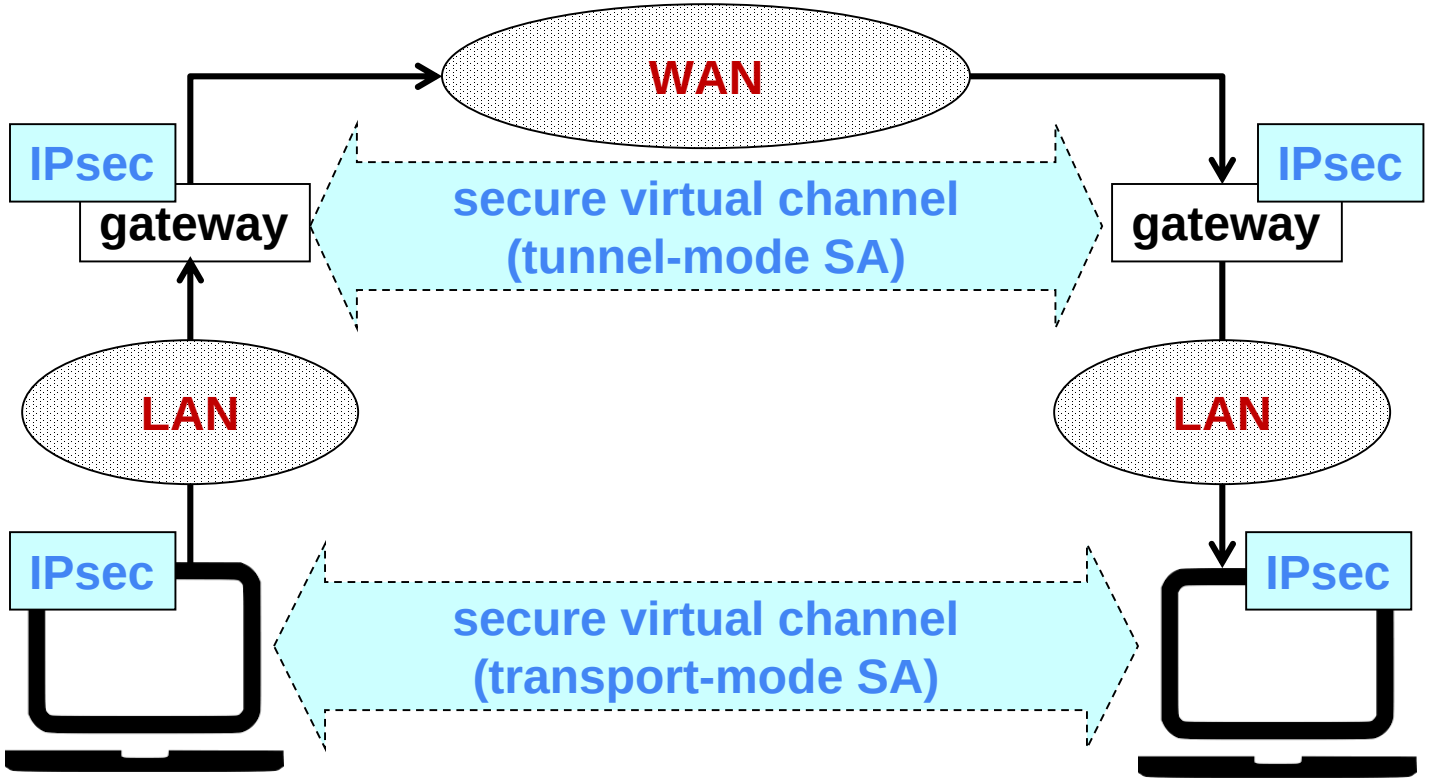
End-to-end security



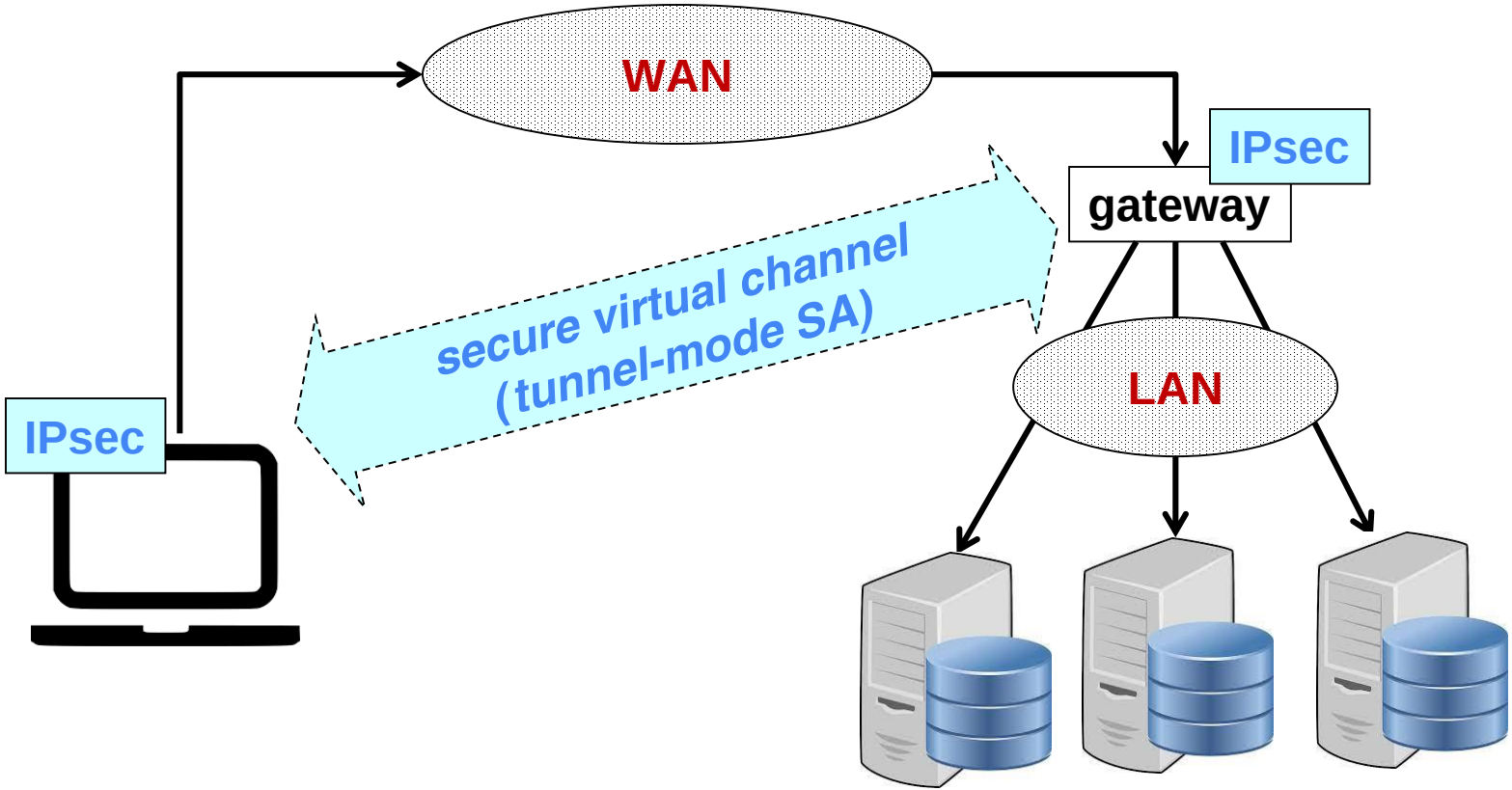
Basic VPN


WAN
IPsec
gateway
secure virtual channel
(tunnel-mode SA)
IPsec
gateway
LAN
LAN

End-to-end security with basic VPN



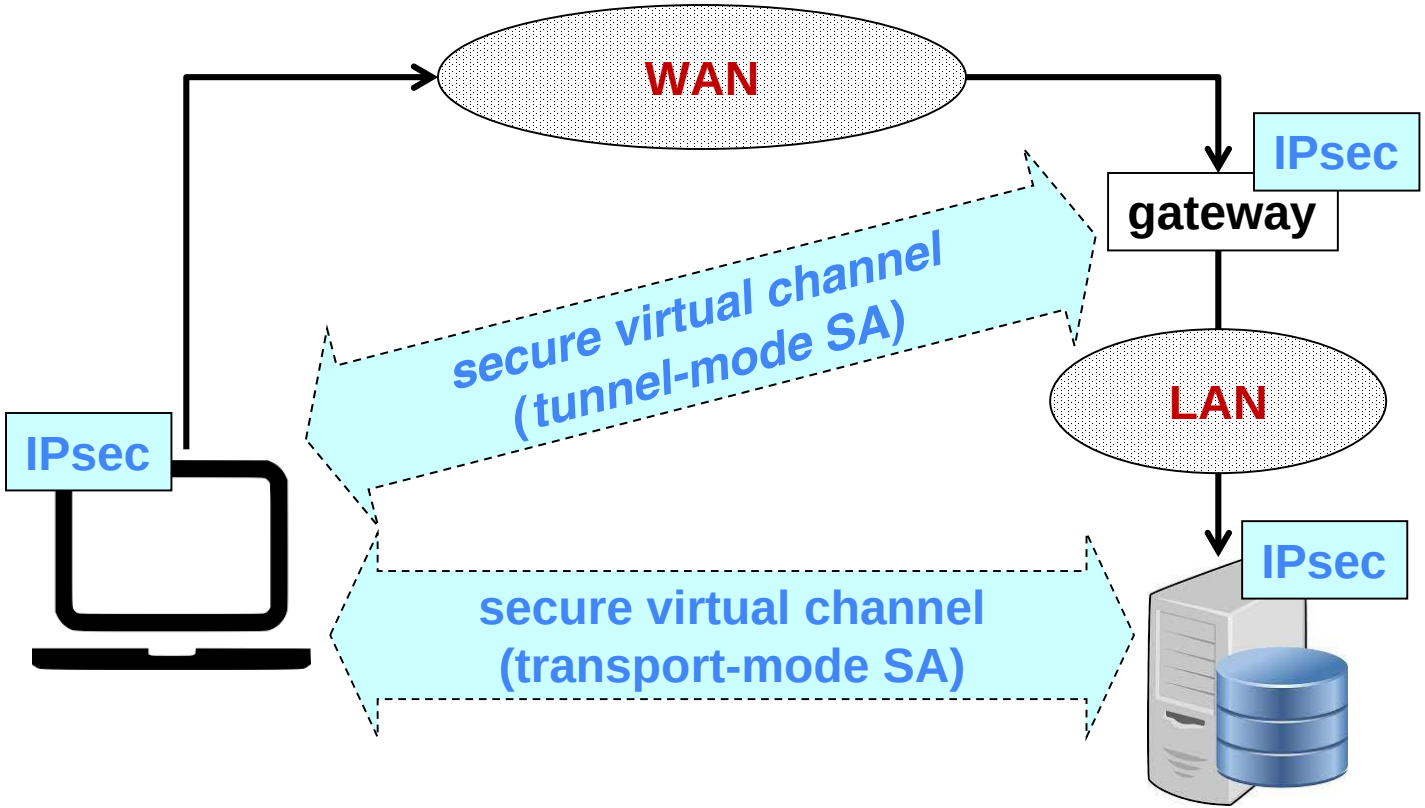
Secure gateway



Secure gateway variants

- **remote worker / nomadic user wishing to access the company network**
  - WAN = Internet, LAN = Intranet
  - beware of security issues if personal device is used also for other kind of connections
- **individual wishing to escape limitations (e.g. ISP, MNO, government, geo-protected content) or wishing to be anonymous**
  - WAN = Internet, LAN = Internet (!)
  - gateway = (personal) VPN provider
    - double-check their log / anonymity policy
  - if services to be accessed are all HTTP-based then an (anonymizing) HTTP proxy may be sufficient

Secure remote access



IPsec key management

- **very important component of IPsec**
- **provides to the IPsec parties the symmetric keys used for packet authentication and/or encryption**
- **what about key distribution?**
  - OOB (e.g. manual)
  - automatic in-band (which protocol?)

ISAKMP

- **Internet Security Association and Key Management Protocol**
- **RFC-2408**
- **procedures needed to negotiate, set-up, modify and delete a SA**
- **key exchange method not fixed:**
  - OAKLEY (RFC-2412): protocol for authenticated exchange of symmetric keys

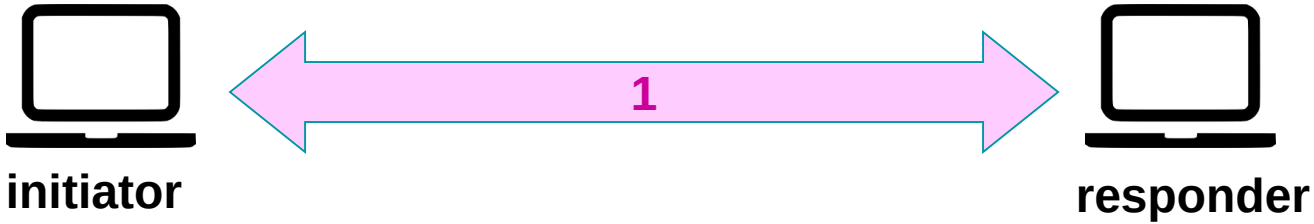
IKE

- **Internet Key Exchange (RFC-2409)**
  - standard protocol for SA establishment and management
  - is a merge of two protocols: ISAKMP + OAKLEY
  - can be executed between
    - two hosts
    - two security gateways
    - a host and a security gateway
  - exploits Diffie-Hellman protocol for authenticated key exchange

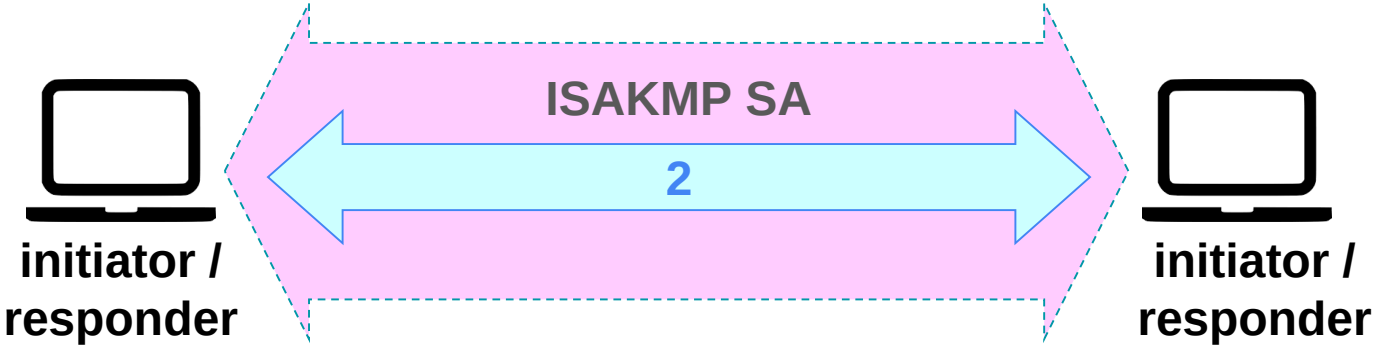
IKE overview

- **IKE is a two phase protocol**
  - Phase 1
    - parties establish a SA to protect the ISAKMP exchange (called ISAKMP SA)
      - performs mutual authentication of parties
      - negotiate cryptographic algorithms, authentication method and keys
    - this SA is used to protect the messages exchanged in Phase 2
  - Phase 2
    - parties establish IPSec SAs that will be used to protect the traffic
    - Phase 2 keys can be derived from phase 1 keys
- **the same ISAKMP SA may be reused several times to negotiate other IPsec SA**

IKE: operations



IKE phase 1 - negotiation of a bidirectional ISAKMP SA: "main mode" or "aggressive mode"



IKE phase 2 - negotiation of the IPsec SA: "quick mode"

IKE: “modes” of operation

- **Main Mode:**
  - 6 messages
  - protects the parties identities
- **Aggressive Mode:**
  - 3 messages (but doesn’t protect the parties identities)
- **Quick Mode:**
  - 3 messages
  - negotiation only of the IPsec SA
- **New Group Mode:**
  - 2 messages

IKE: authentication methods

- **Digital Signature**
  - non-repudiation of the IKE negotiation
- **Public Key Encryption**
  - identity protection in the aggressive mode
- **Revised Public Key Encryption**
  - less expensive, only 2 public-key operations
- **Pre-Shared Key**
  - the party ID may only be its IP address (problem with mobile users)

VPN concentrator

- **special-purpose appliance that acts as a terminator of IPsec tunnel:**
  - for remote access of single clients
  - to create site-to-site VPN
- **very high performance with respect to the costs (low)**

Applicability of IPsec

- **only unicast packets (no broadcast, no multicast, no anycast)**
- **between parties that activated a SA:**
  - by shared keys
  - by X.509v3 certificates
- **… therefore in “closed” groups**