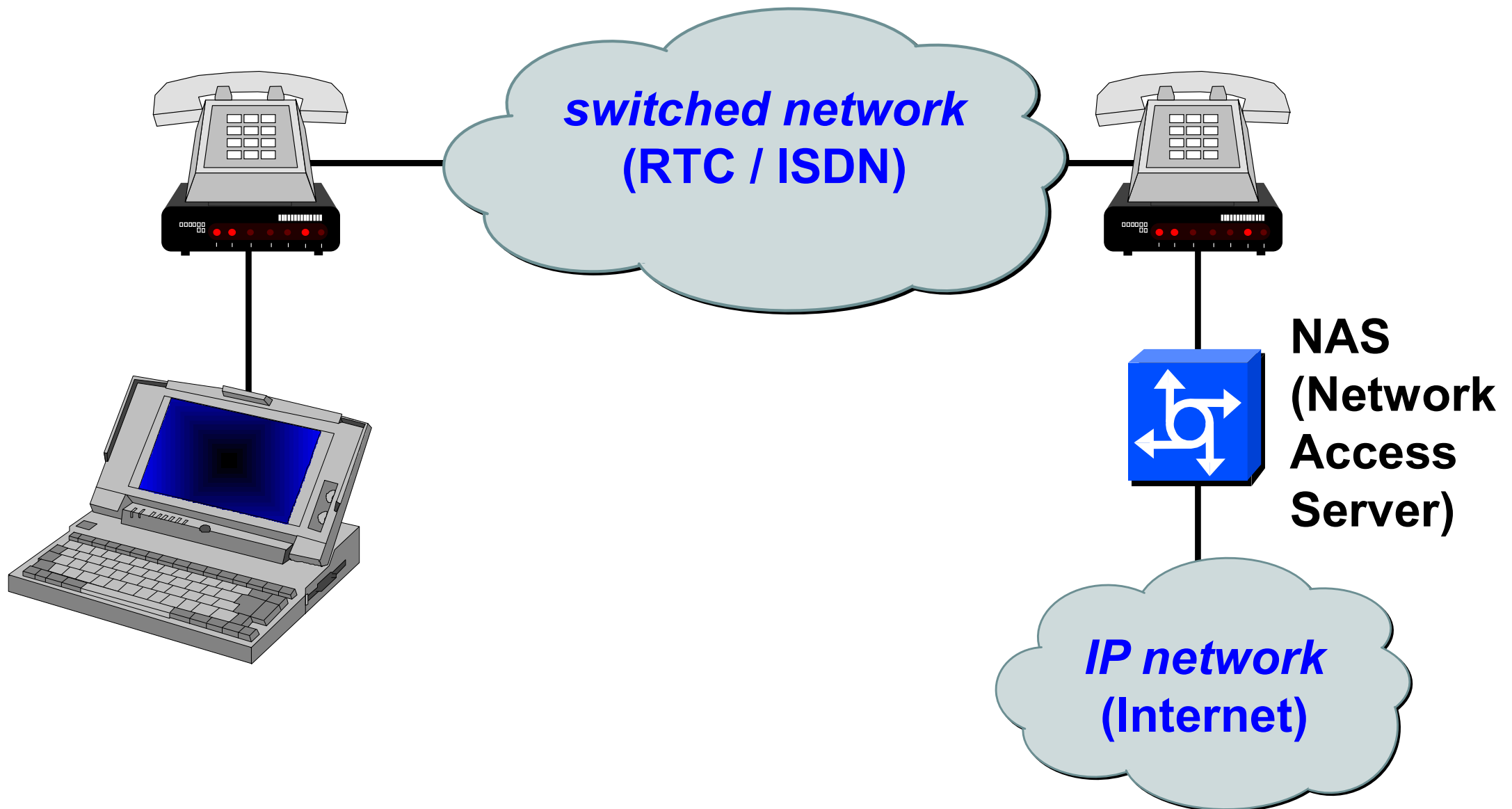


Security of IP networks

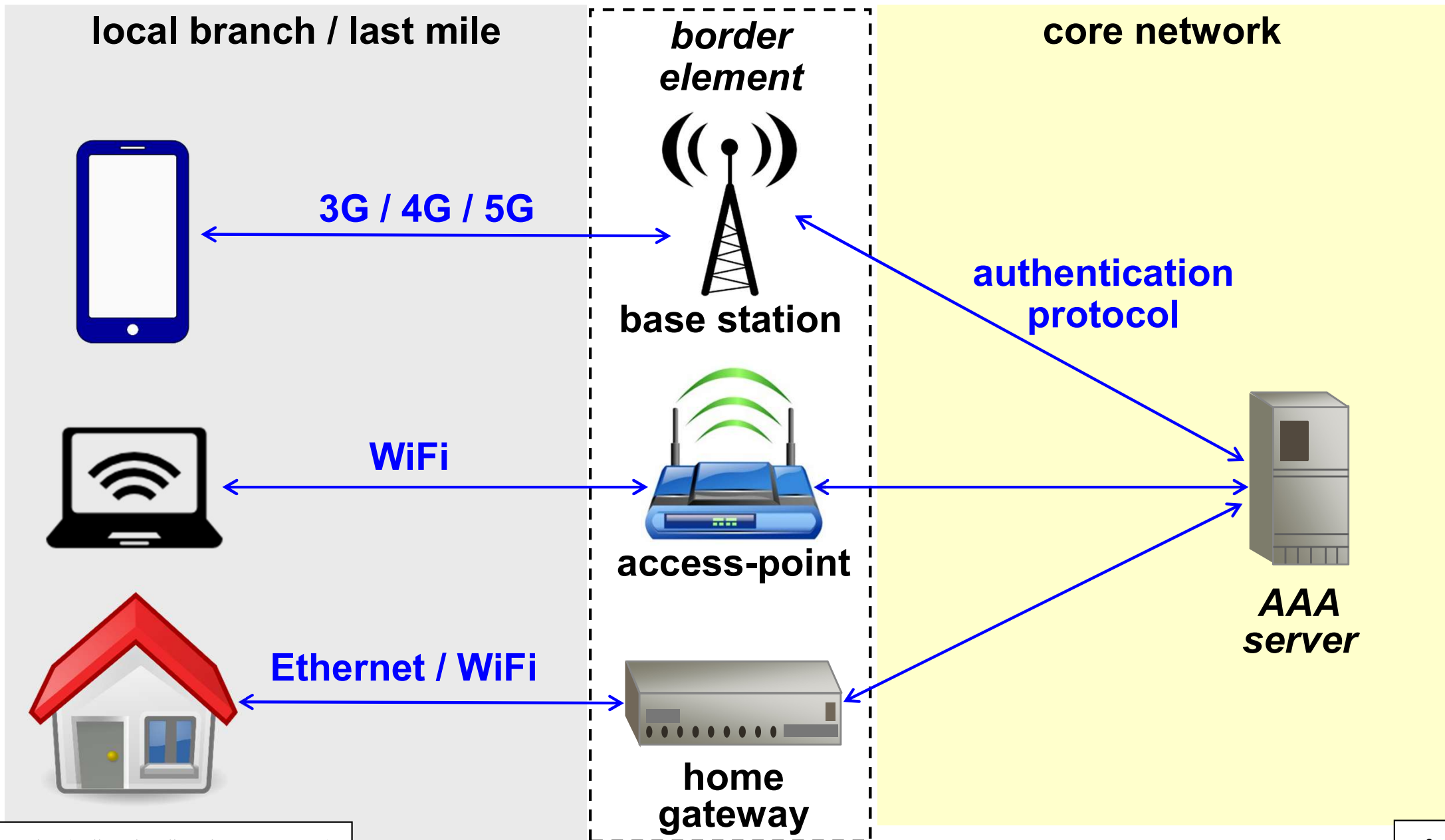
Antonio Lioy
< lioy @ polito.it >

Politecnico di Torino
Dip. Automatica e Informatica

Remote access via dial-up lines



Network access (modern way)



Authentication of PPP channels

- **PPP is a protocol ...**
 - ... to encapsulate network packets (L3, e.g. IP) ...
 - ... and carry them over a point-to-point link
 - physical (e.g. RTC, ISDN)
 - virtual L2 (e.g. xDSL with PPPoE)
 - virtual L3 (e.g. L2TP over UDP/IP)
- **activated in three sequential steps:**
 - LCP (Link Control Protocol)
 - authentication (optional; PAP, CHAP or EAP)
 - L3 encapsulation (e.g. IPCP, IP Control Protocol)

Authentication of network access

- for dial-up and for wireless and virtual links
- **PAP**
 - Password Authentication Protocol
 - user password sent in clear
- **CHAP**
 - Challenge Handshake Authentication Protocol
 - symmetric challenge (based on user password)
- **EAP**
 - Extensible Authentication Protocol
 - external techniques (challenge, OTP, TLS)

EAP

- **RFC-3748 (extended by RFC-5247)**
“PPP Extensible Authentication Protocol (EAP)”
- **a flexible L2 authentication framework**
- **authentication predefined mechanisms:**
 - MD5-challenge (similar to CHAP)
 - OTP
 - generic token card
- **other mechanisms may be added:**
 - RFC-2716 “PPP EAP TLS authentication protocol”
 - RFC-3579 “RADIUS support for EAP”

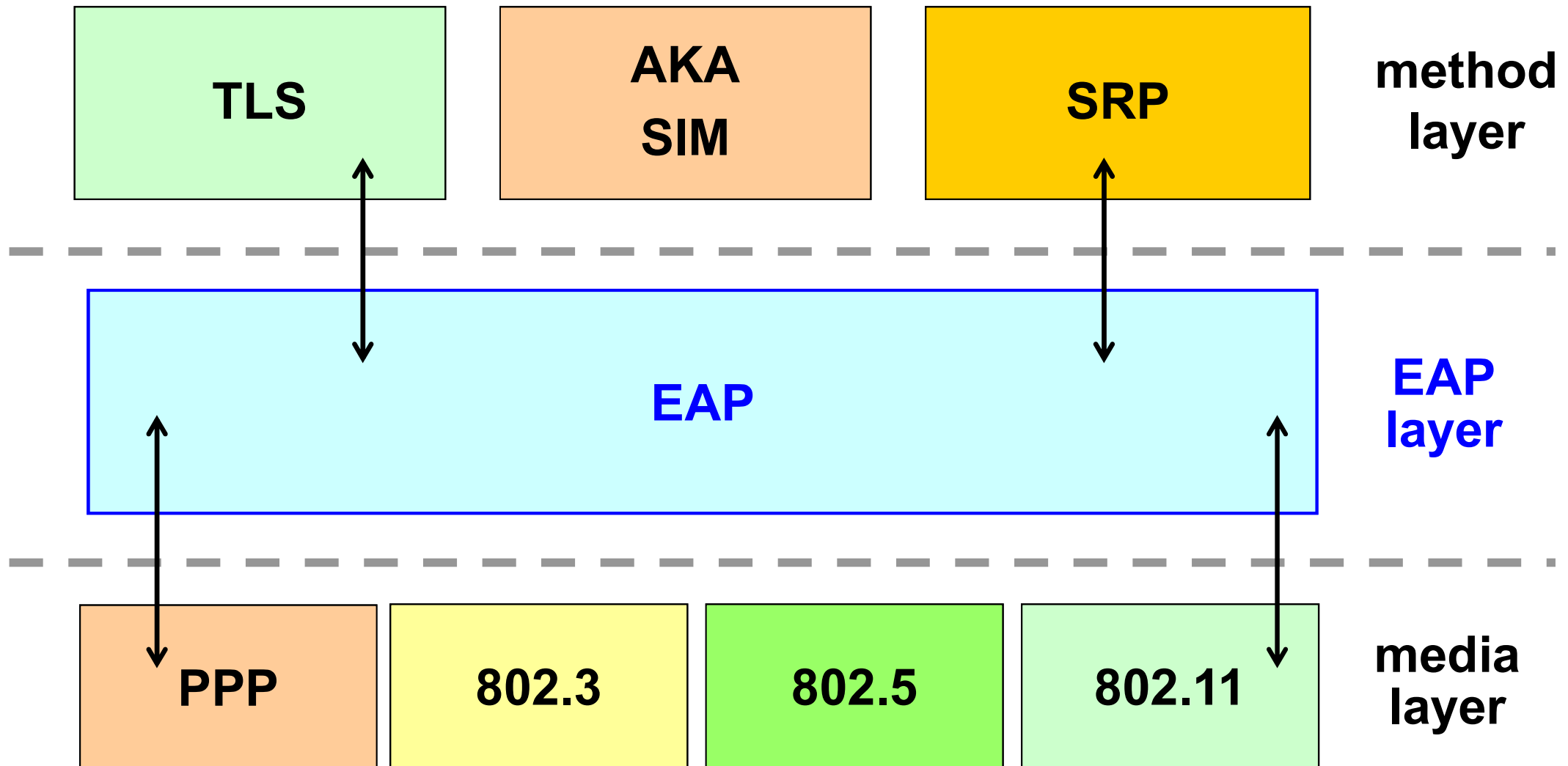
EAP - encapsulation

- authentication data are transported via its own encapsulation protocol (because L3 packets are not yet available ...)
- features of EAP encapsulation:
 - independent of IP
 - supports any link layer (e.g. PPP, 802, ...)
 - explicit ACK/NAK (no windowing)
 - assumes no reordering (PPP guarantees ordering, UDP and raw IP do not!)
 - retransmission (max 3-5 retransmissions)
 - no fragmentation (must be provided by EAP methods for payload greater than min EAP MTU)

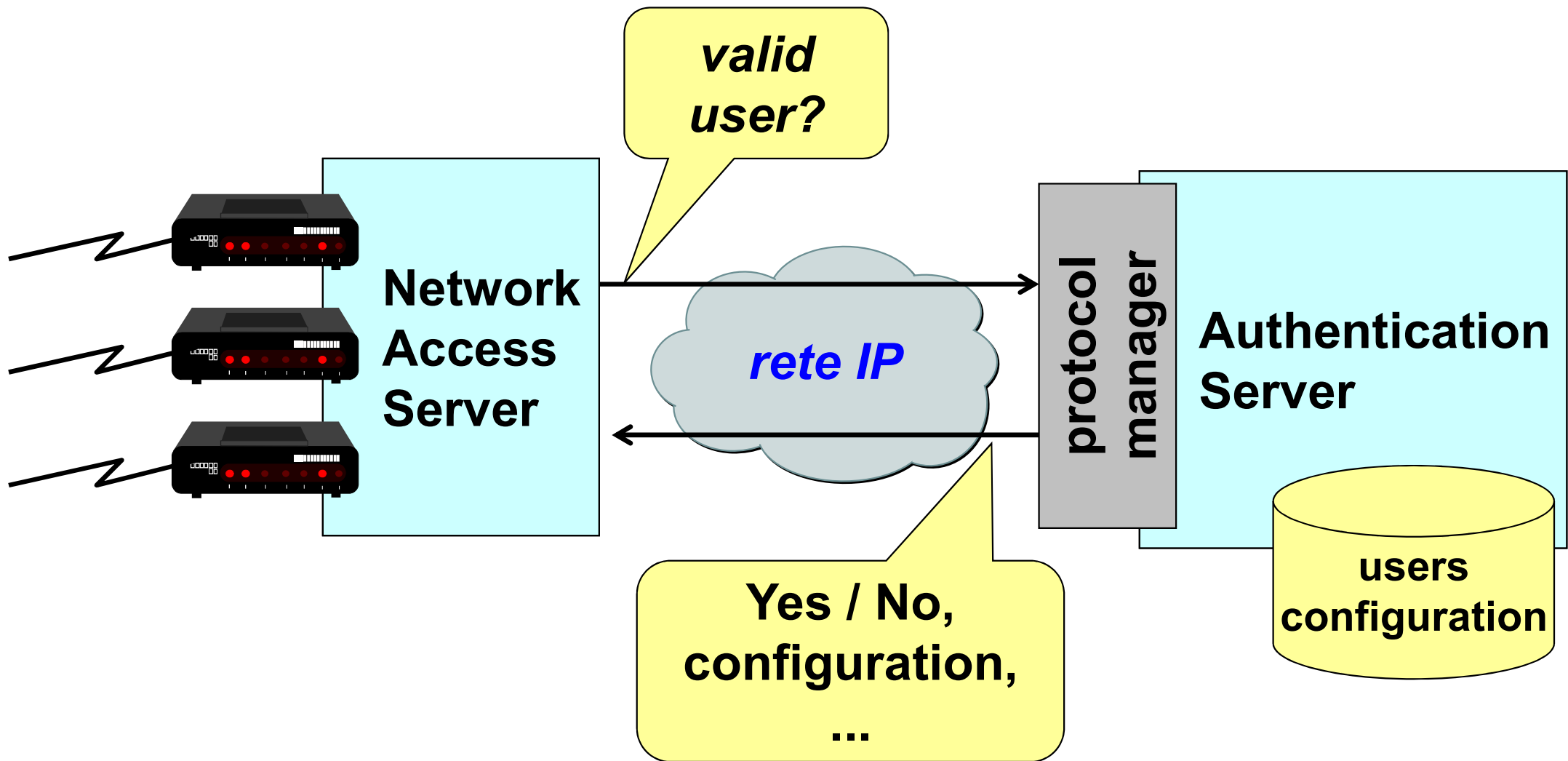
EAP

- **the link is not assumed to be physically secure**
 - EAP methods must provide security on their own
- **some EAP methods:**
 - EAP-TLS (RFC-5216)
 - EAP-MD5 (RFC-3748) – only EAP peer authentication, no mutual authentication
 - EAP-TTLS = tunnelled TLS (to operate any EAP method protected by TLS)
 - EAP-SRP (Secure Remote Password)
 - GSS_API (includes Kerberos)
 - AKA-SIM (RFC-4186, RFC-4187)

EAP - architecture



Authentication for network access



AAA

- **the NAS manufacturers claim that security needs three functions:**
 - *Authentication* – entity's identity is authenticated based on credentials (e.g. password, OTP)
 - *Authorization* – determining whether an entity is authorized to perform a given activity or gain access to resources/services
 - *Accounting* – tracking network resource usage for audit support, capacity analysis or cost billing
- **the AS performs exactly these three functions talking with one or more NAS via one or more protocols**

Network authentication protocols

■ RADIUS

- the de-facto standard
- proxy towards other AS

■ DIAMETER

- evolution of RADIUS
- emphasis on roaming among different ISP
- takes care of security

■ TACACS+ (TACACS, XTACACS)

- originally technically better than RADIUS, achieved smaller acceptance because it was a proprietary solution (Cisco)

RADIUS

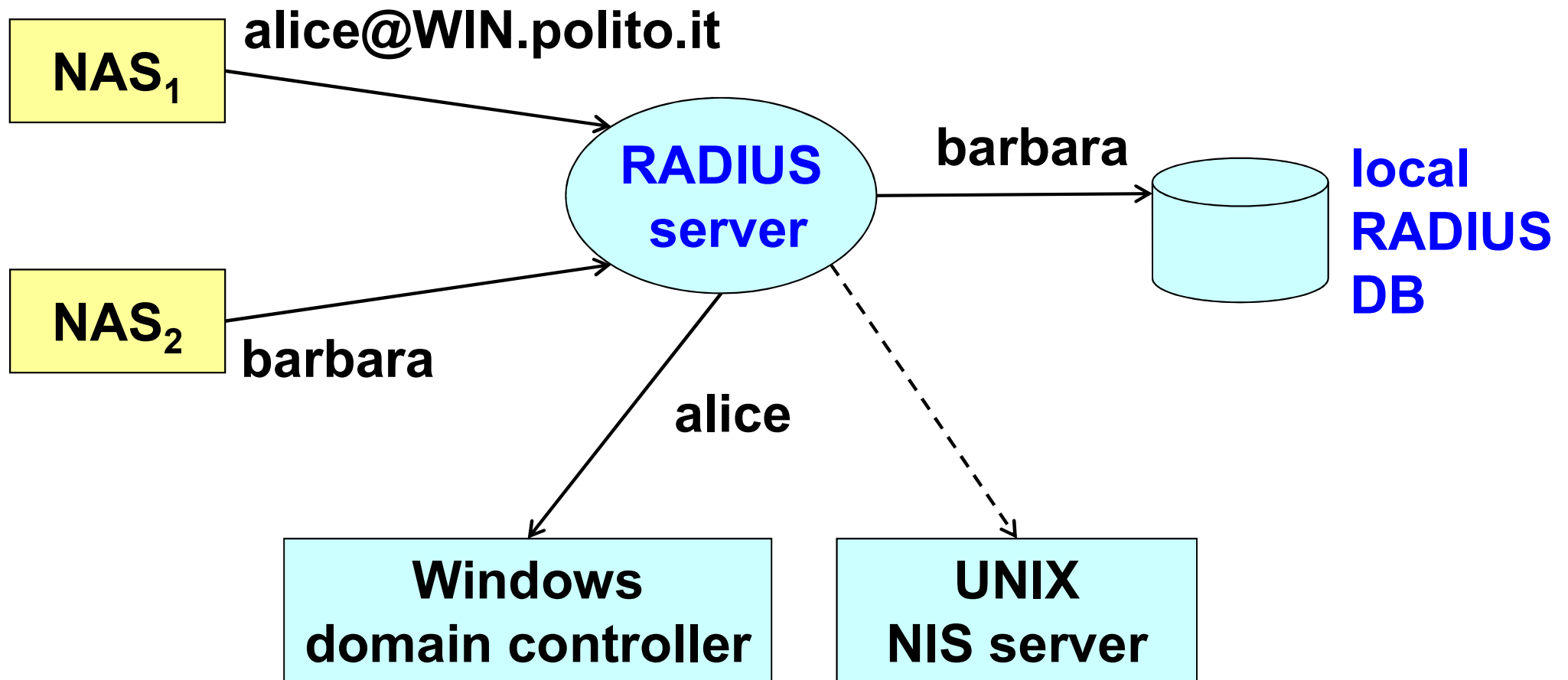
- **Remote Authentication Dial-In User Service**
- **Livingston Technologies (1991) then IETF**
- **supports authentication, authorization and accounting to control network access:**
 - physical ports (analogical, ISDN, IEEE 802)
 - virtual ports (tunnel, wireless access)
- **centralized administration and accounting**
- **client-server schema between NAS and AS**
 - port 1812/UDP (authentication) and 1813/UDP (accounting) ; unofficial ports: 1645 & 1646/UDP
 - timeout + retransmission
 - secondary server

RADIUS - RFC

- **RFC-2865 (protocol)**
- **RFC-2866 (accounting)**
- **RFC-2867/2868 (tunnel accounting and attributes)**
- **RFC-2869 (extensions)**
- **RFC-3579 (RADIUS support for EAP)**
- **RFC-3580 (guidelines for 802.1X with RADIUS)**

RADIUS proxy

- the RADIUS server may act as a proxy towards other authentication servers



Which security functionalities for Radius?

- **sniffing NAS req (if contains pwd)**
 - confidentiality, privacy
- **fake AS resp (to block valid or allow invalid user)**
- **changing AS resp ($Y > N$ or $N > Y$)**
 - authN & integrity of AS resp
- **replay of AS resp (if not properly tied to NAS req)**
 - anti-replay of AS resp
- **pwd enumeration (from fake NAS)**
 - authN of NAS req
- **DoS (many NAS req from fake NAS)**
 - server scalability

RADIUS: data protection

- **packet integrity and authentication via keyed-MD5:**
 - key = shared-secret
 - client without key are ignored
- **password transmitted “encrypted” with MD5 (after padding with NUL bytes to a multiple of 128 bit):**

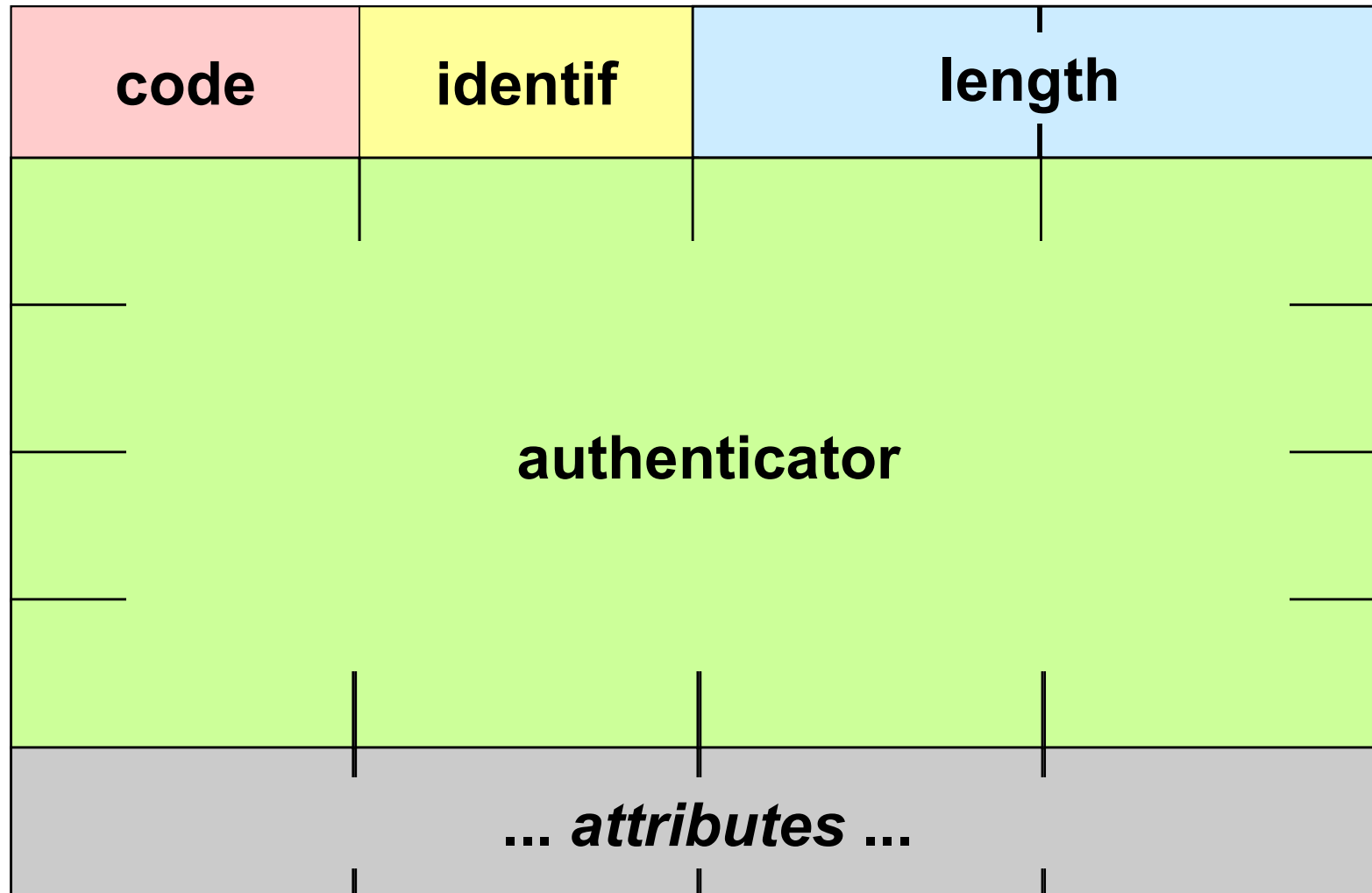
password \oplus md5(key+authenticator)

RADIUS

- **user authentication via PAP, CHAP, token-card and EAP**
 - CISCO provides a free server for CryptoCard
 - others support SecurID
- **attributes in TLV form, easily extensible without modification to installed base (by ignoring any unknown Type):**

attribute type – length – value

RADIUS - format



RADIUS – packet types

■ ACCESS-REQUEST

- contains access credentials (e.g. username + pwd)

■ ACCESS-REJECT

- access is denied (e.g. due to bad username/pwd)

■ ACCESS-CHALLENGE

- requests additional info from the user (e.g. a PIN, token code, secondary password)

■ ACCESS-ACCEPT (*parameters*):

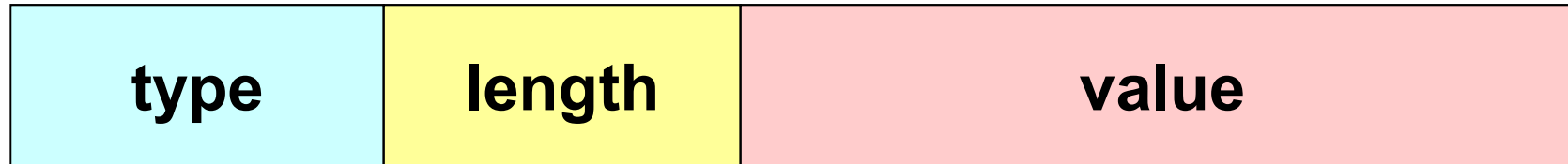
- access is granted + network parameters are given
 - for SLIP/PPP: IPaddr, netmask, MTU, ...
 - for terminal: host, port

RADIUS - authenticator

- **double purpose:**
 - server reply authentication and no replay
 - masking the password
- **in Access-Request:**
 - it is named Request Authenticator
 - 16 byte randomly generated by the NAS
- **in Access-Accept / Reject / Challenge**
 - it is named Response Authenticator
 - it is computed via a keyed-digest:

md5 (code || ID || length || RequestAuth || attributes || secret)

RADIUS - some attributes

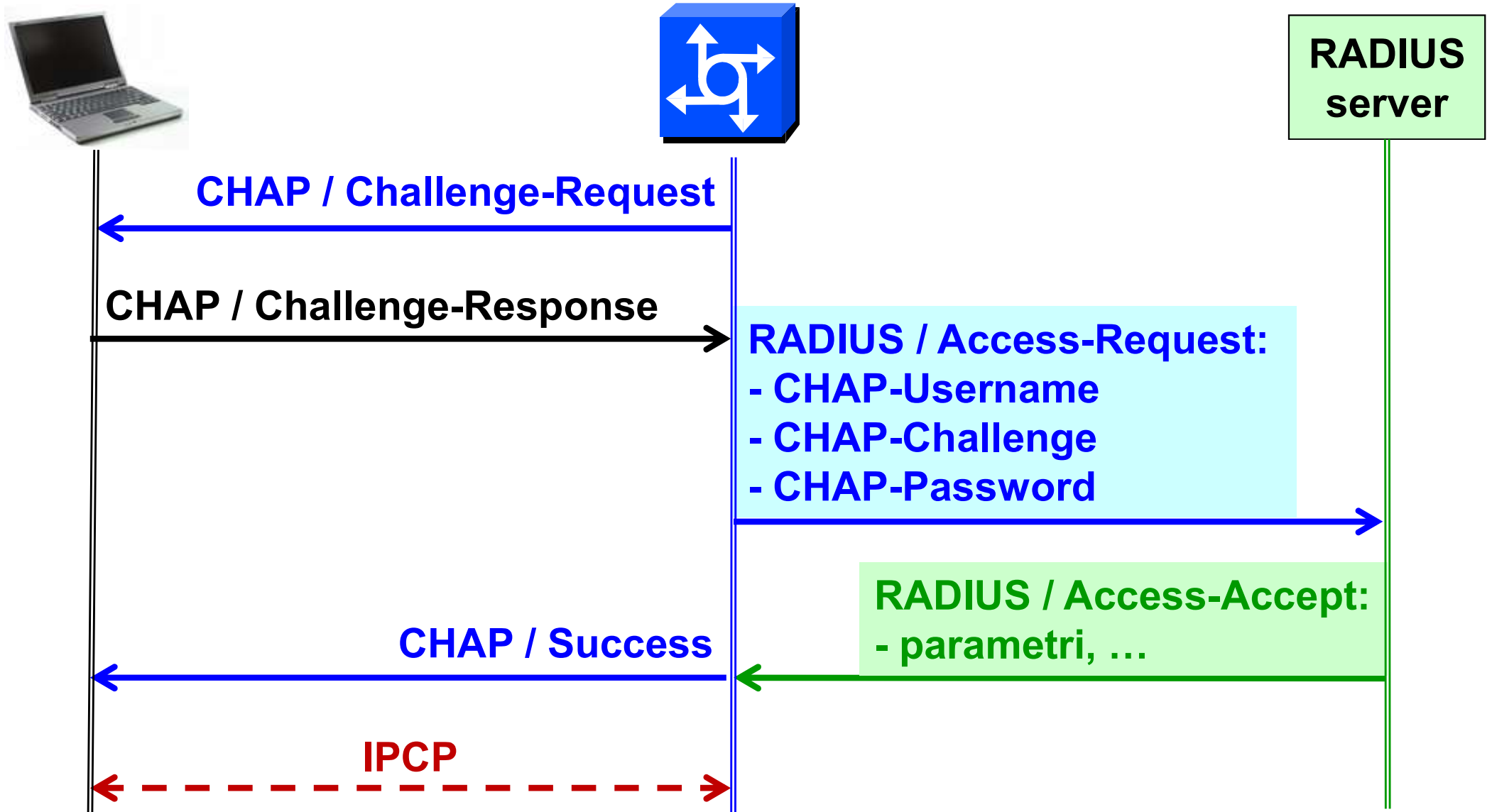


- **type = 1 (User-Name)**
 - value = text, network access identifier (NAI), DN
- **type = 2 (User-Password)**
 - value = password \oplus md5 (key || RequestAuthent.)
- **type = 3 (Chap-Password)**
 - value = user CHAP response (128 bit)
- **type = 60 (CHAP-Challenge)**
 - value = challenge from the NAS to the user

NAI (Network Access Identifier)

- **RFC-2486**
- **NAI = username [@ realm]**
- **all devices must support NAI up to 72 byte long**
- **the exact syntax for username and realm is in the RFC (note that only ASCII characters < 128 are allowed, but all of them are allowed)**
- **note that the username is the one used in the PPP authentication phase (does not necessarily match the application username)**

Example - CHAP + RADIUS



DIAMETER

- evolution of RADIUS
- special emphasis on roaming between ISP
- RFC-3588 “Diameter base protocol”
- RFC-3589 “Commands for the 3GPP”
- RFC-3539 “AAA transport profile”
- RFC-4004 “Diameter mobile IPv4 application”
- RFC-4005 “Diameter network access server application”
- RFC-4006 “Diameter credit-control application”
- RFC-4072 “Diameter EAP application”

Security of DIAMETER

- **compulsory protection via IPsec or TLS:**
 - Diameter client MUST support IPsec and MAY support TLS
 - Diameter server MUST support IPsec and TLS
- **compulsory configurations:**
 - (IPsec) ESP with non null algo for both authentication and privacy
 - (TLS) mutual authentication (client MUST have a public-key certificate)
 - (TLS) MUST support RSA+RC4_128/3DES+ MD5/SHA1 e MAY support RSA+AES_128+SHA1

IEEE 802.1x

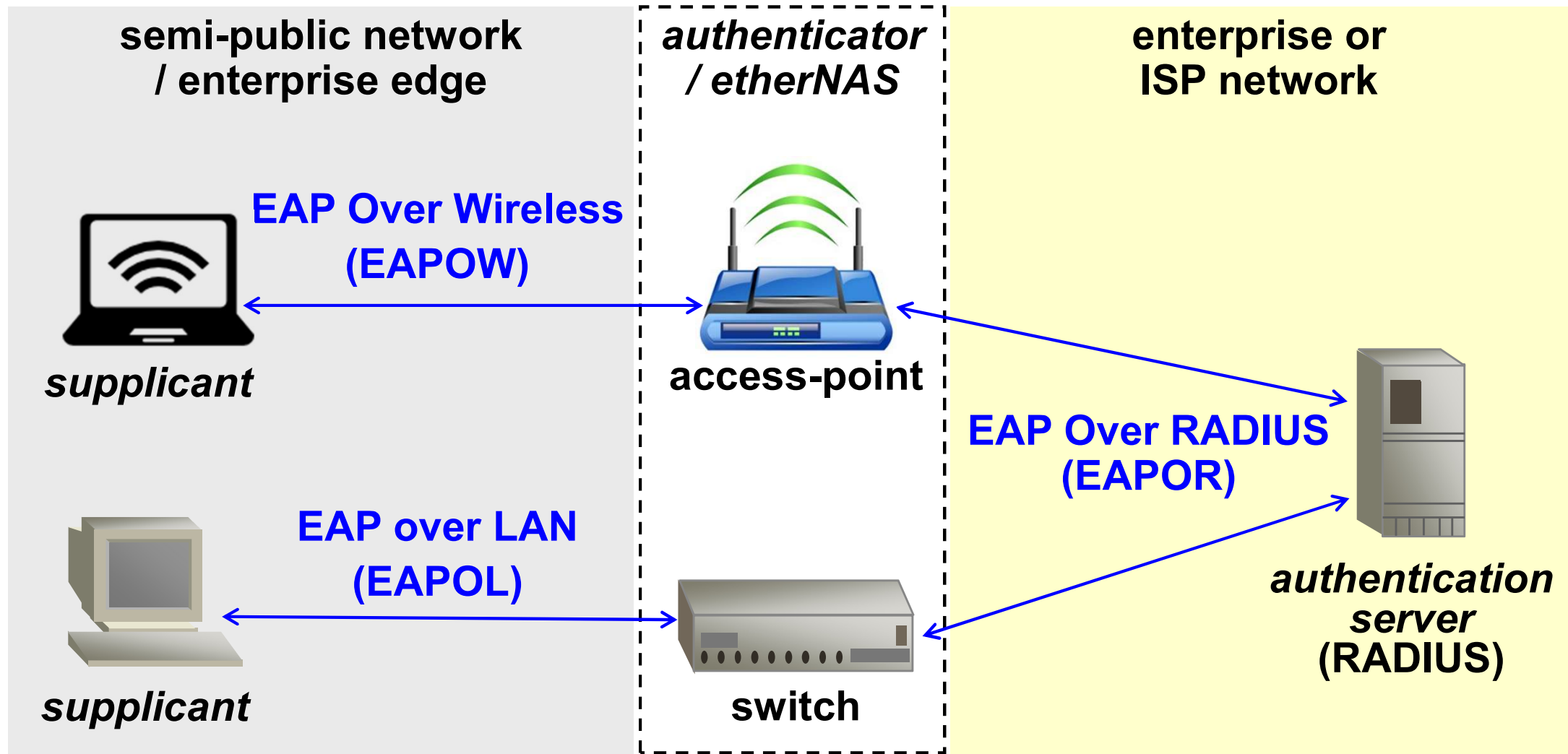
- **Port-Based Network Access Control:**
 - L2 authentication architecture
 - useful in a wired network to block access
 - absolutely needed in wireless networks
- **first implementations:**
 - Windows-XP and Cisco wireless access-points

<http://standards.ieee.org/getieee802/download/802.1X-2001.pdf>

IEEE 802.1x

- **authentication and key-management framework:**
 - may derive session keys for use in packet authentication, integrity and confidentiality
 - standard algorithms for key derivation (e.g. TLS, SRP, ...)
 - optional security services (authentication or authentication+encryption)

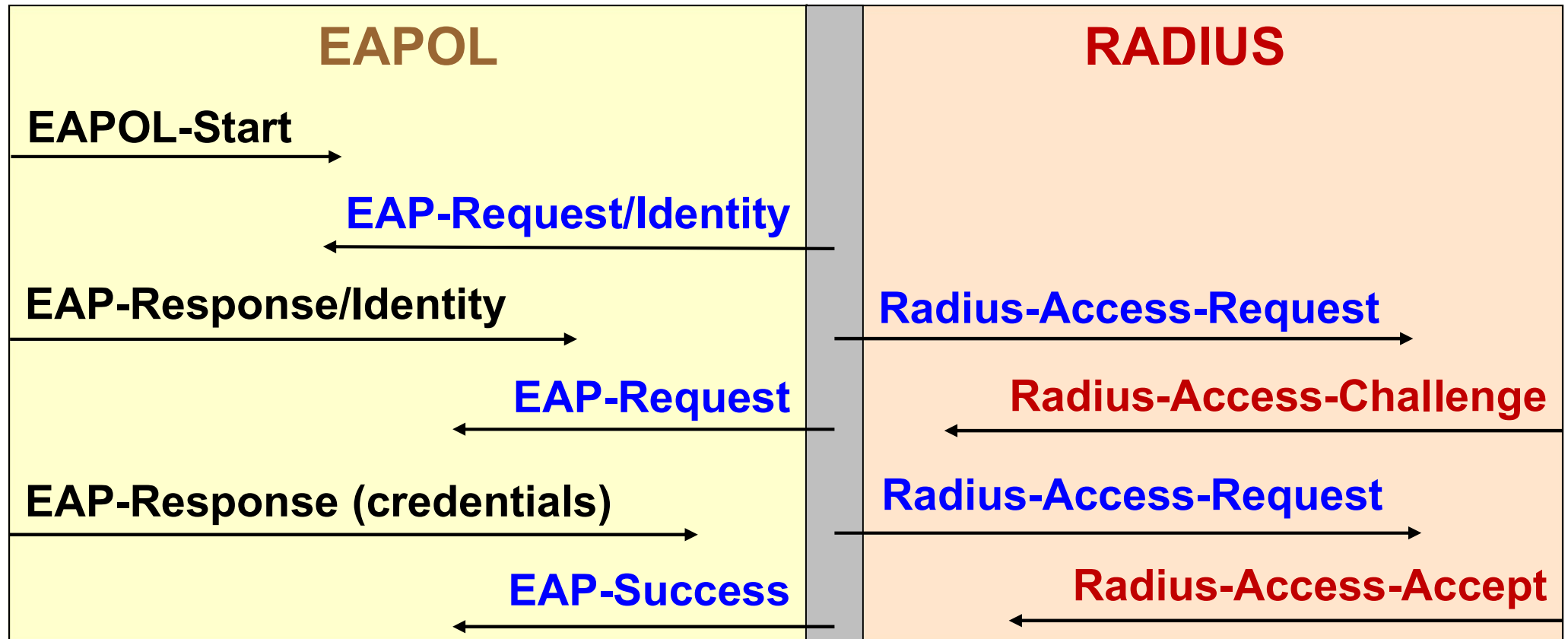
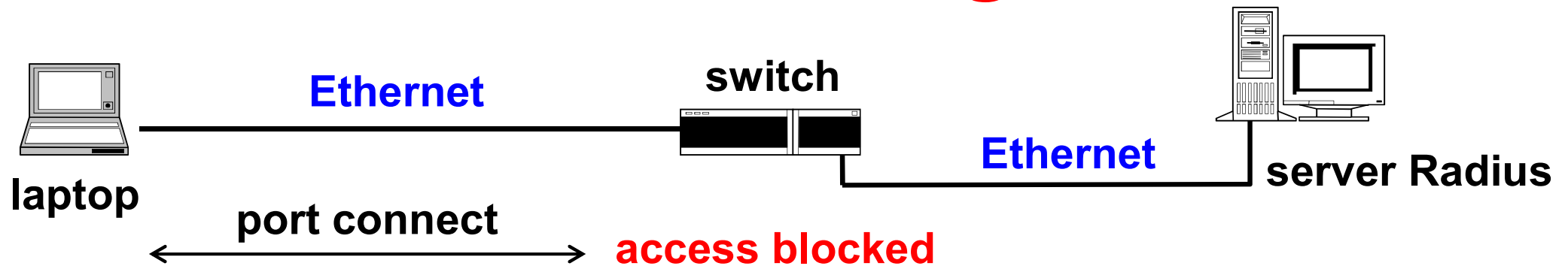
802.1x - architecture



802.1x - advantages

- **exploits the application level for the actual implementation of the security mechanisms**
 - direct dialogue between supplicant and AS
 - NIC and NAS operate as “pass-through device”
 - no change needed on NIC and NAS to implement new mechanisms
 - perfect integration in AAA

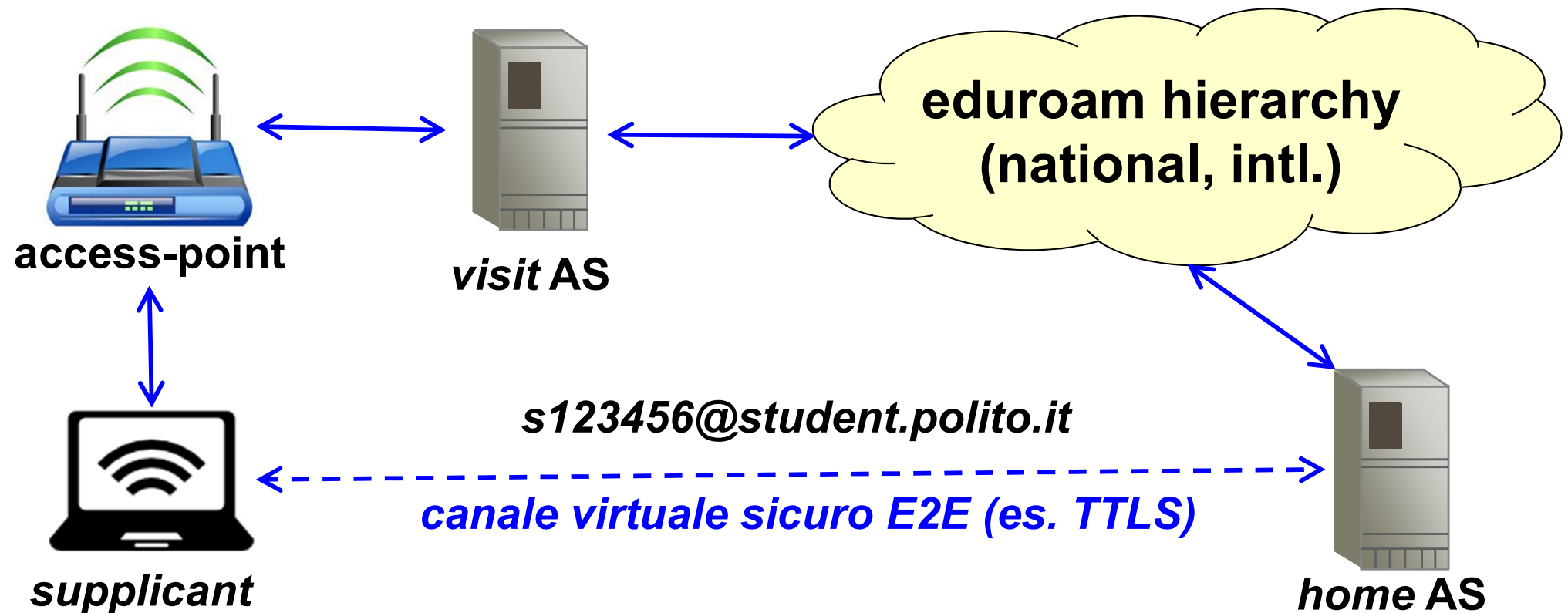
802.1x - messages



access allowed

eduroam

- **accesso WiFi in istituzioni di ricerca (Italia, Europea, ...)**
 - (2/11/2019) 101 paesi più 18 in test
 - usa 802.1x + federazione RADIUS



Which is the best OSI level to implement security?

Application
Presentation
Session
Transport
Network
Data Link
Physical



firewall? IPSEC?
smart-card?
encryption box?
guards?

Optimal level?

- 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 we leave more room for DoS attacks
- 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)

DHCP security

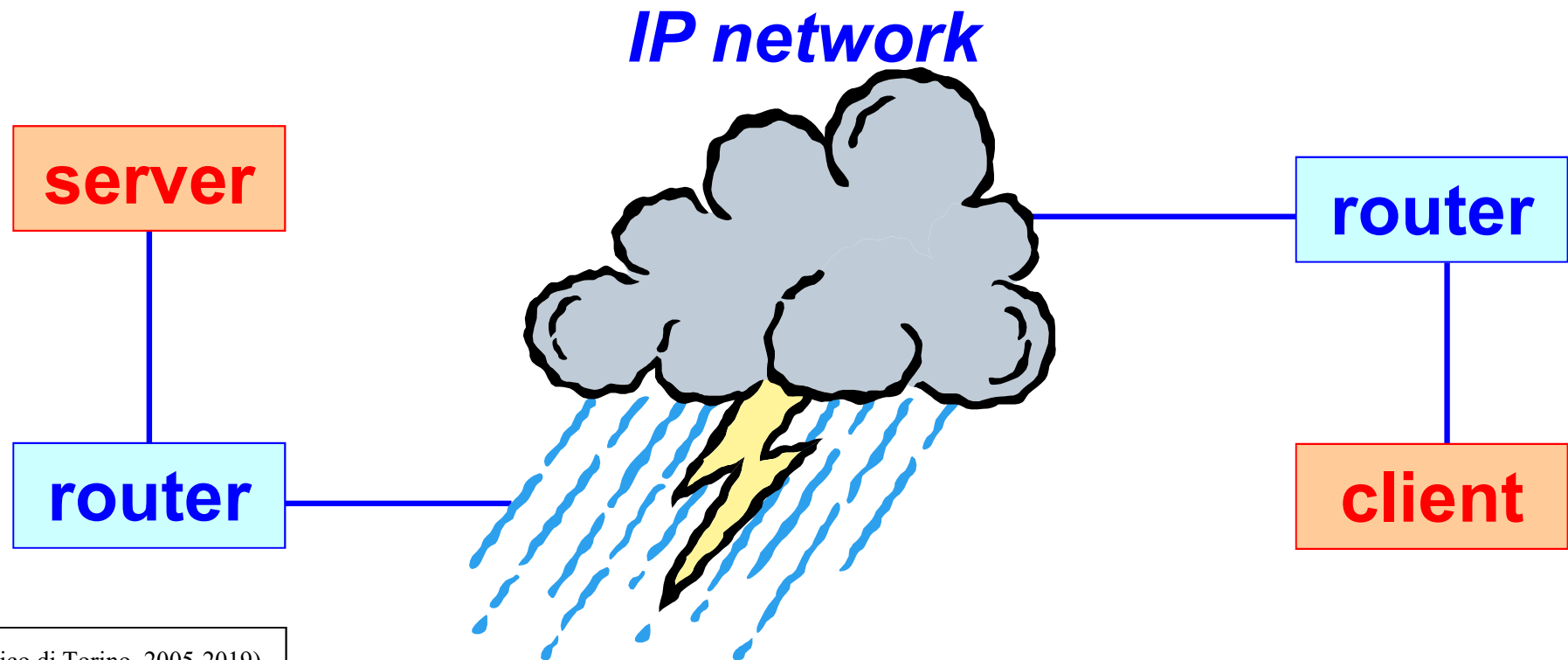
- **non-authenticated protocol**
- **activation of a shadow server is trivial**
- **possible attacks from the fake server:**
 - **denial-of-service**
 - provides a wrong network configuration
 - **MITM**
 - provides a configuration with a 2-bit subnet + default gateway equal to an attacker host
 - if we additionally activate NAT we can intercept the replies too

DHCP protection

- **some switch (e.g. Cisco) offers:**
 - DHCPsnooping = only replies from “trusted ports”
 - IP guard = only IP got from a DHCP server (but there is a limit on the number of recognized addresses)
- **RFC-3118 “Authentication for DHCP messages”**
 - use of HMAC-MD5 to authenticate the messages
 - rarely adopted

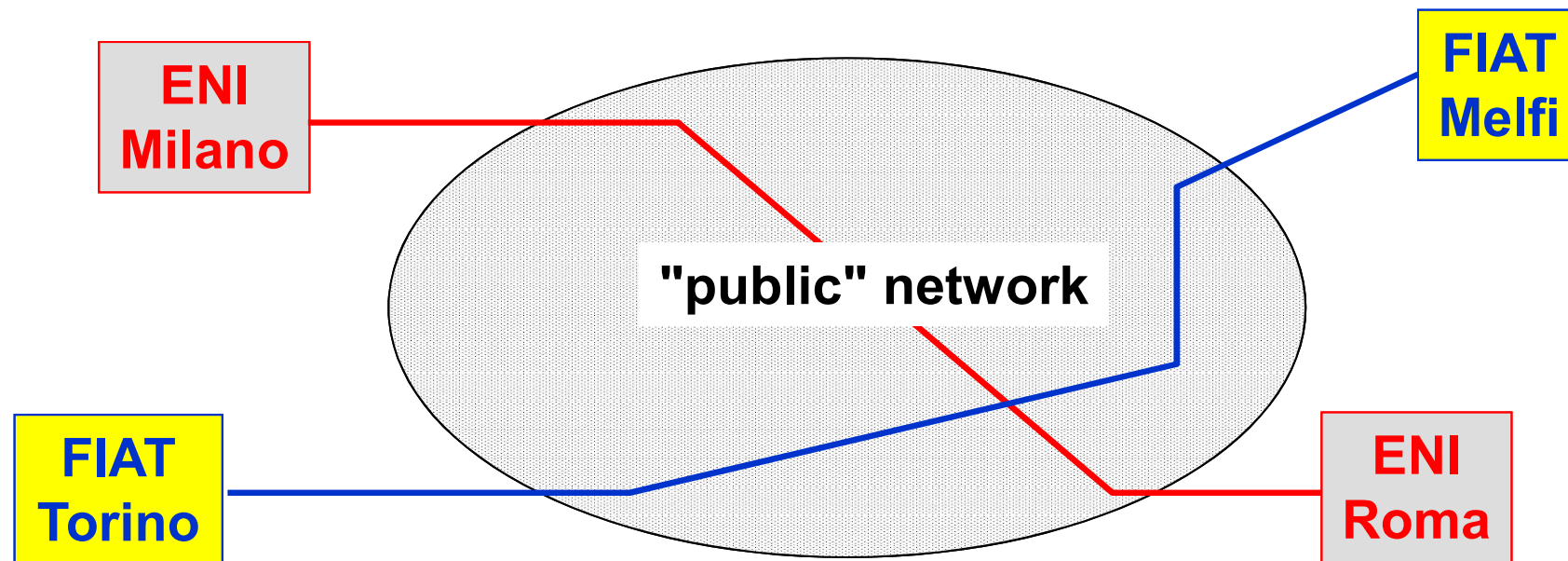
Security at network level (L3)

- 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



Techniques to create a VPN

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

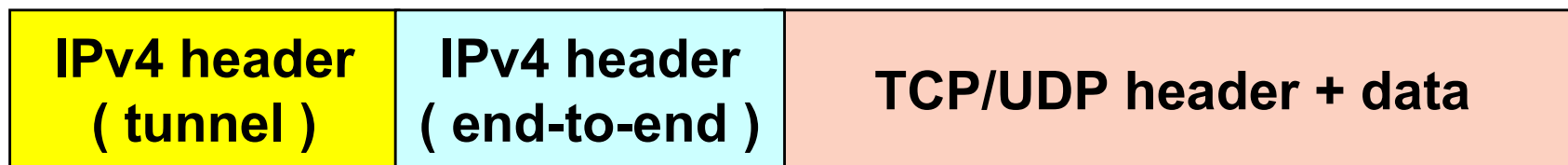
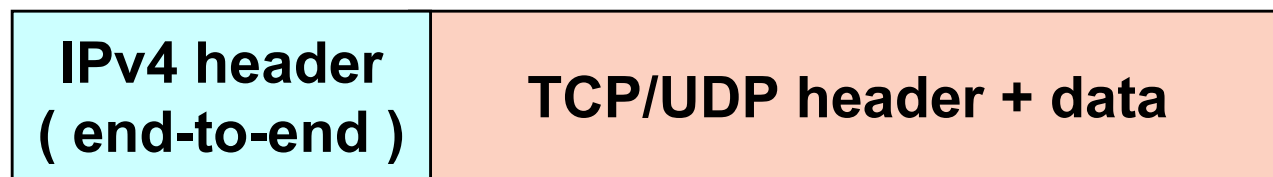
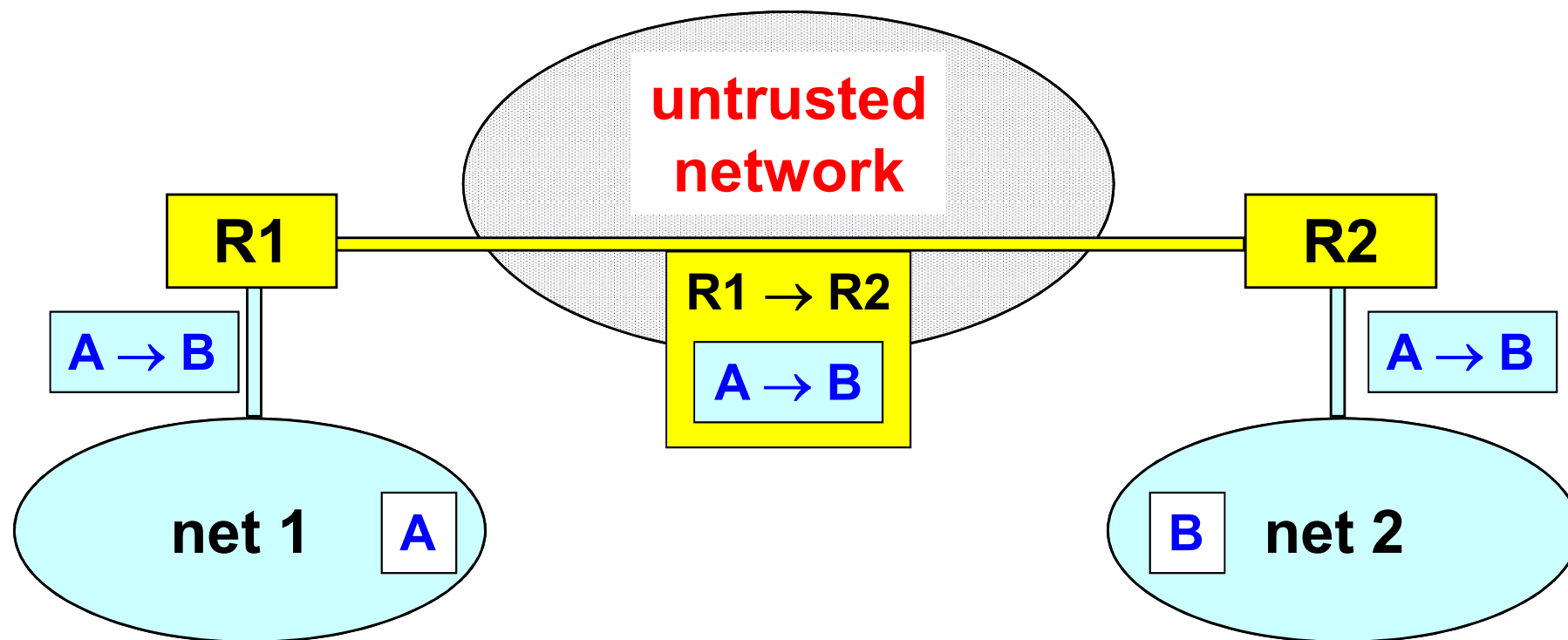
1. VPN via private addresses

- the networks to be part of the VPN use non-public 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 somebody:
 - guesses or discovers the addresses
 - can sniff the packets during transmission
 - has access to the communication devices

2. VPN via tunnel

- **the routers encapsulate whole L3 packets as a payload inside another packet**
 - IP in IP
 - IP over MPLS
 - other
- **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



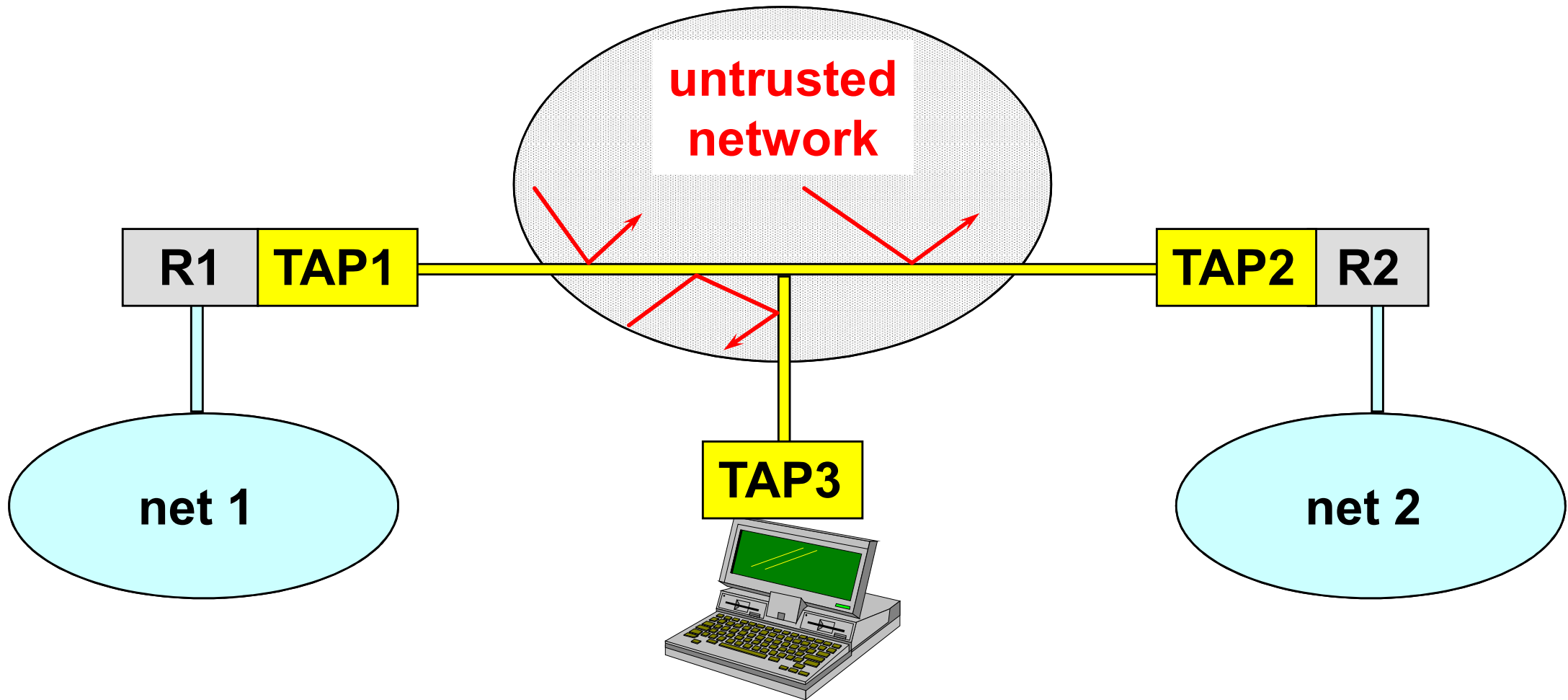
IP tunnel: fragmentation

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

VPN via secure IP tunnel



IPsec

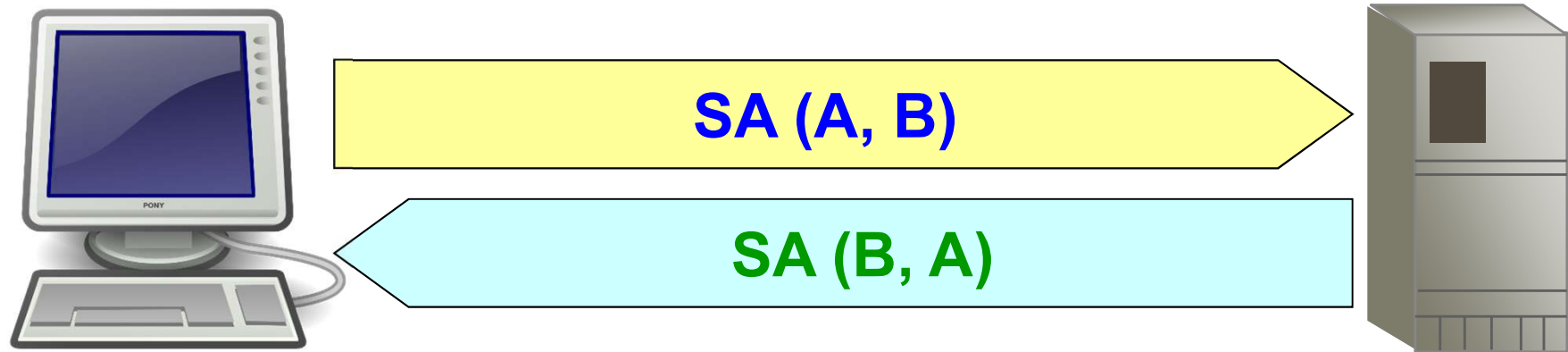
- **IETF architecture for L3 security in IPv4 / IPv6:**
 - to create S-VPN over untrusted networks
 - to create end-to-end secure packet flows
- **definition of two specific packet types:**
 - AH (Authentication Header)
for integrity, authentication, no replay
 - ESP (Encapsulating Security Payload)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

IPsec security services

- **authentication of IP packets:**
 - data integrity
 - sender authentication
 - (partial) protection against “replay” attacks
- **confidentiality of IP packets:**
 - data encryption

IPsec Security Association (SA)

- unidirectional logic connection between two IPsec systems
- each SA has associated different security services
- two SA are needed to get complete protection of a bidirectional packet flow



IPsec local database

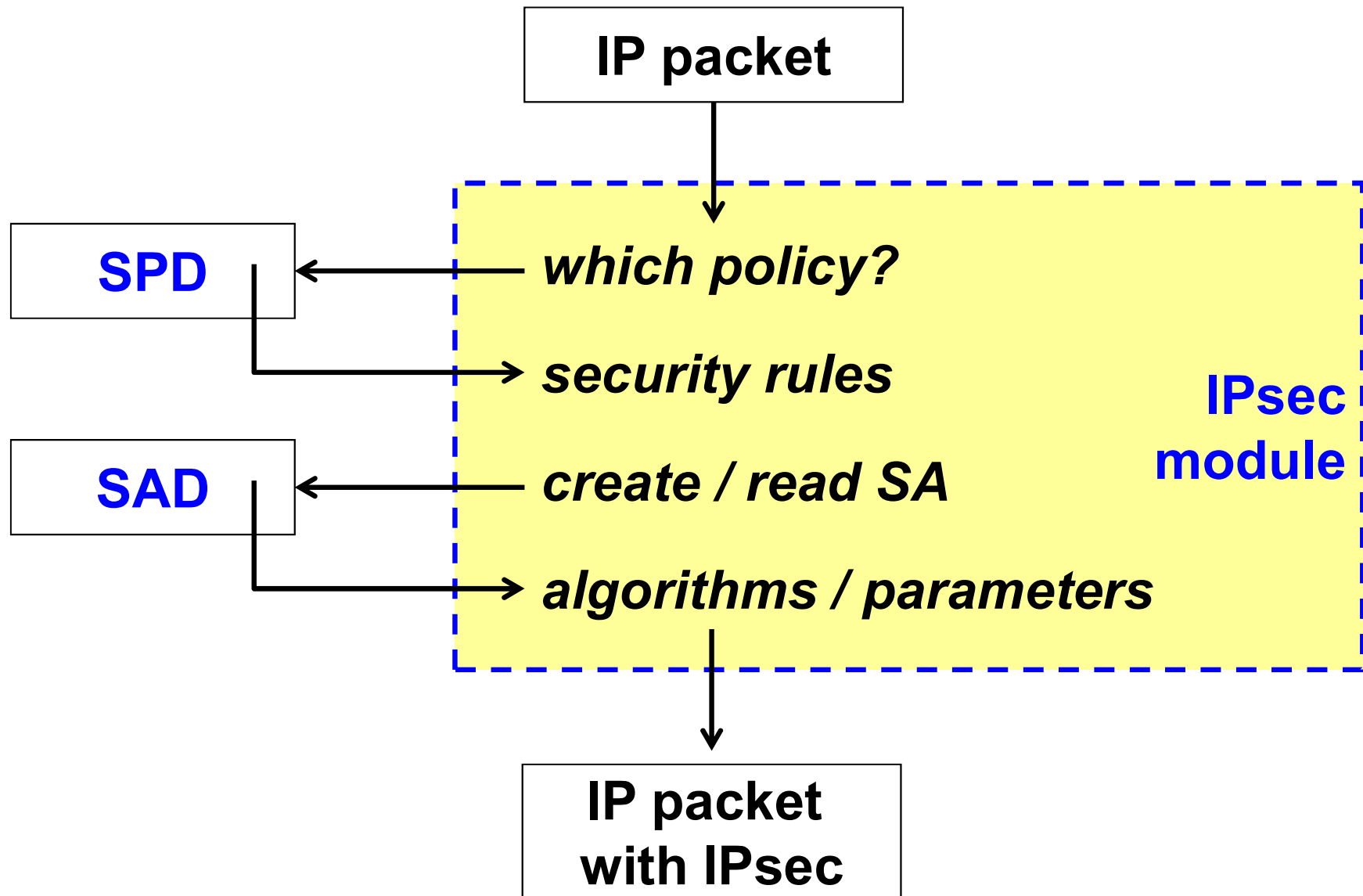
- **SAD (SA Database)**

- list of active SA and their characteristics (algorithms, keys, parameters)

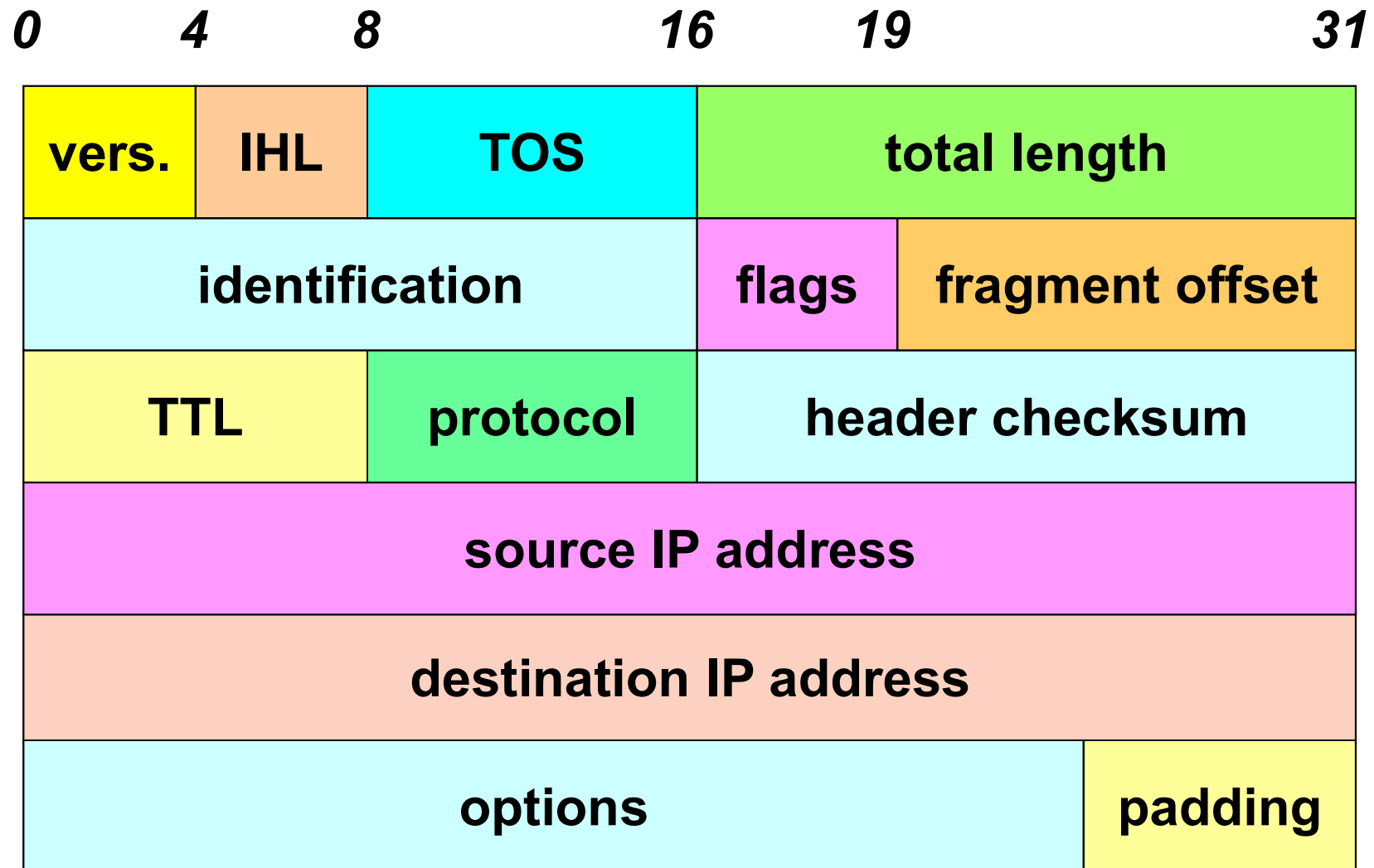
- **SPD (Security Policy Database)**

- list of security policy to apply to the different packet flows
- a-priori configured (e.g. manually) or connected to an automatic system (e.g. ISPS, Internet Security Policy System)

How IPsec works (sending)



IPv4 header

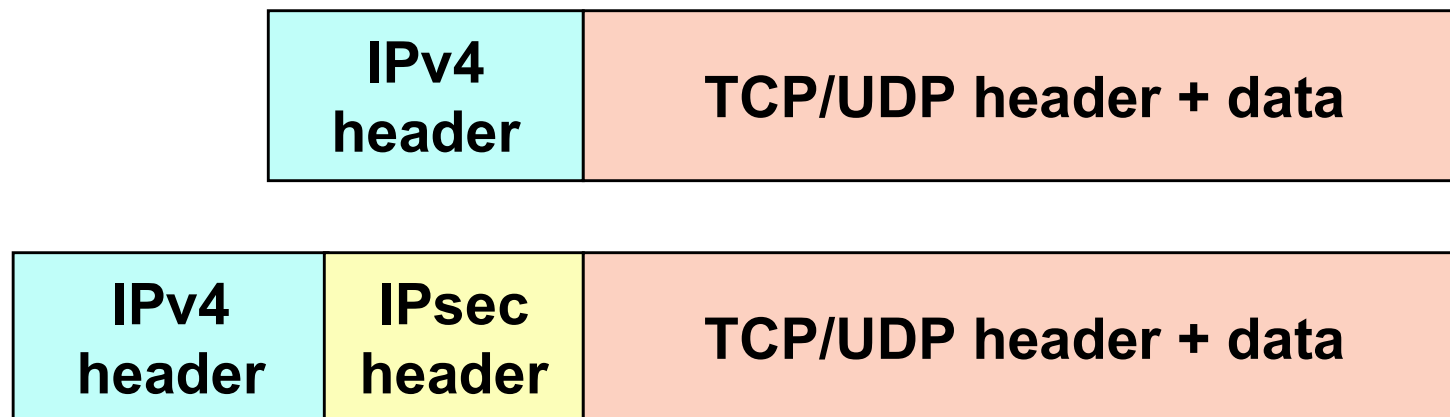


IPv4 header fields

- **IP addresses** (32 bit) of sender and receiver
- **IHL** (Internet Header Length) in 32-bit words
- **TOS** (Type Of Service): nearly ever used (!)
- **length**: no. of bytes of the IP packet
- **identification**: ID of the packet (for fragments)
- **flags**: may/don't fragment, last/more fragments
- **TTL** (Time To Live): max number of hops
- **protocol**: protocol of the payload

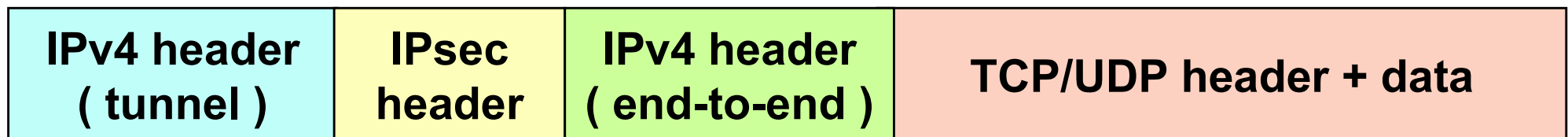
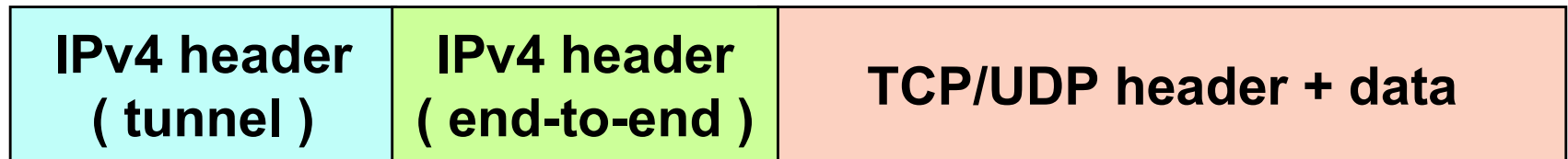
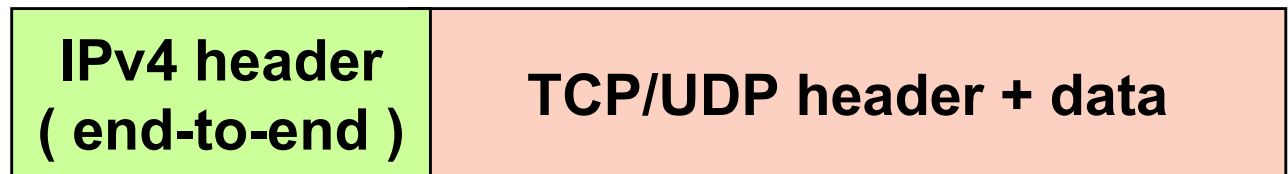
Transport mode IPsec

- used for end-to-end security, that is used by hosts, not gateways (exception: traffic for the gateway itself, e.g. SNMP, ICMP)
- 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

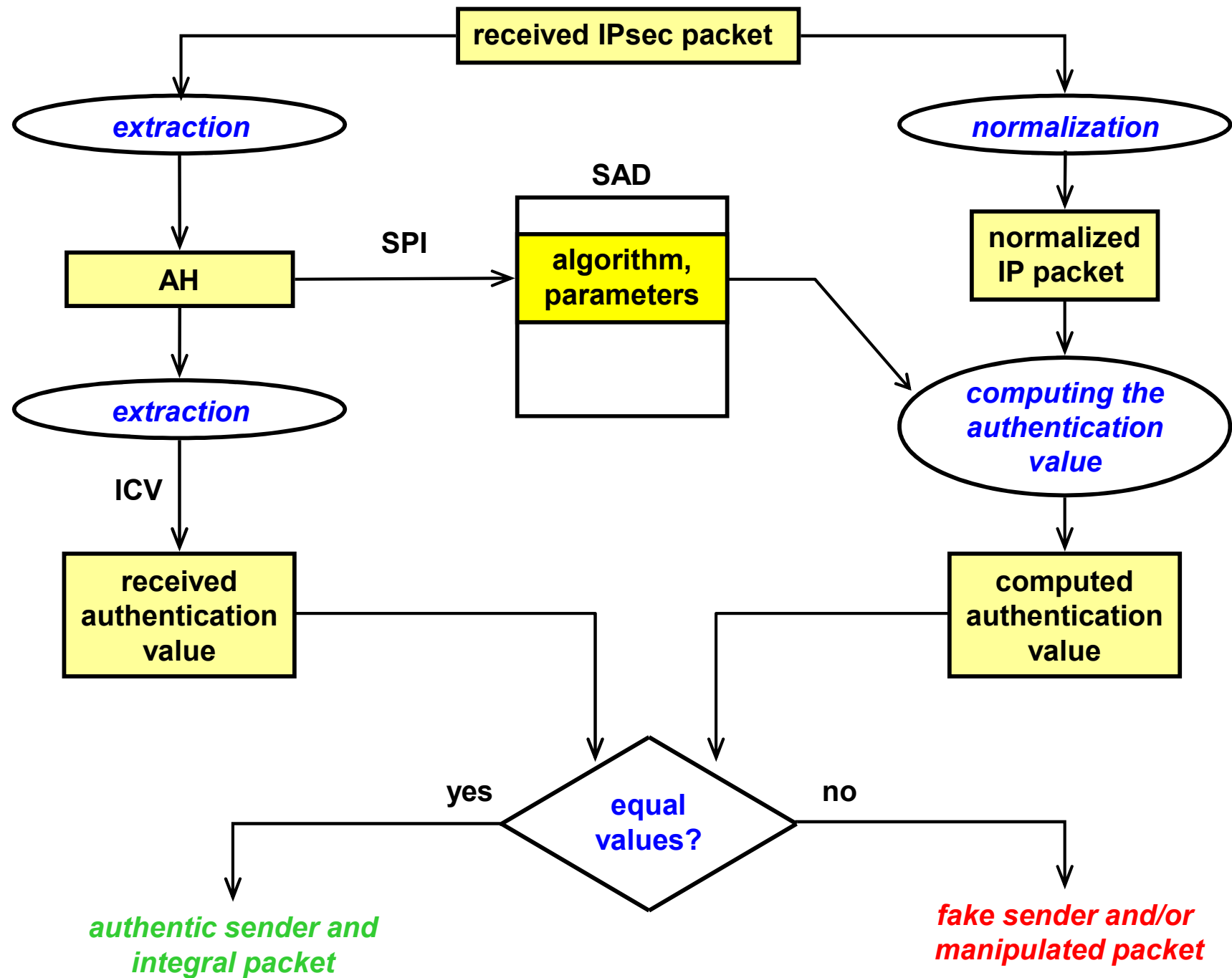


AH

- **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)

Next Header	Length	<i>reserved</i>
Security Parameters Index (SPI)		
Sequence number		
▪	<i>authentication data</i>	▪
▪	(ICV, Integrity Check Value)	▪
▪		▪



Normalization for AH

- **reset the TTL / Hop Limit field**
- **if the packet contains a Routing Header, then:**
 - set the destination field to the address of the final destination
 - set the content of the routing header to the value that it will have at destination
 - set the Address Index field at the value that it will have at destination
- **reset all options with the C bit (*change en route*) set**

HMAC-SHA1-96

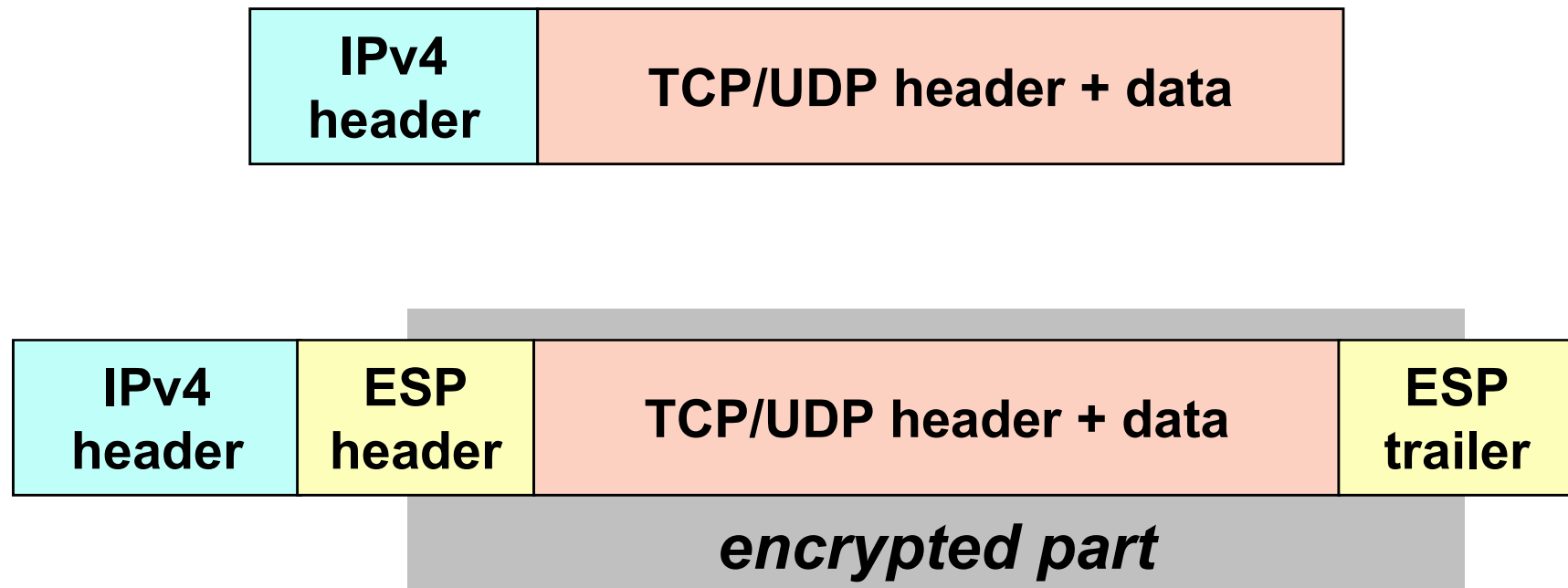
- given **M** normalize it to generate **M'**
- pad **M'** to a multiple of 160 bit (by adding 0x00 bytes) to generate **M'p**
- compute the authentication base:
B = HMAC-SHA1 (K, M'p)
- **ICV = 96 leftmost bits of B**

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

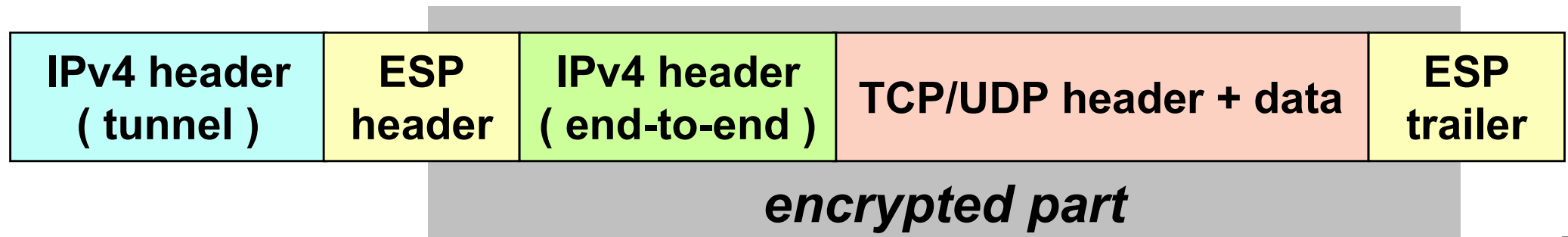
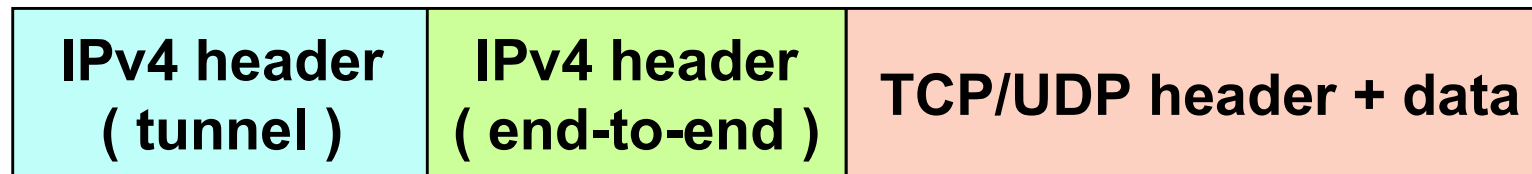
ESP in transport mode

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

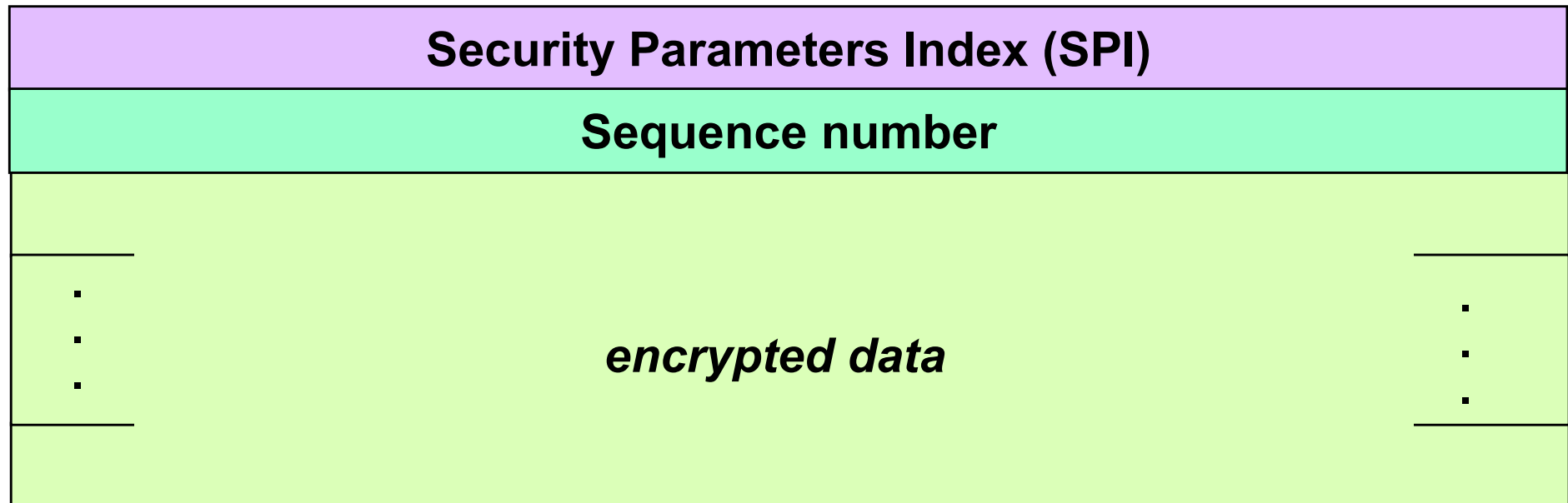


ESP in tunnel mode

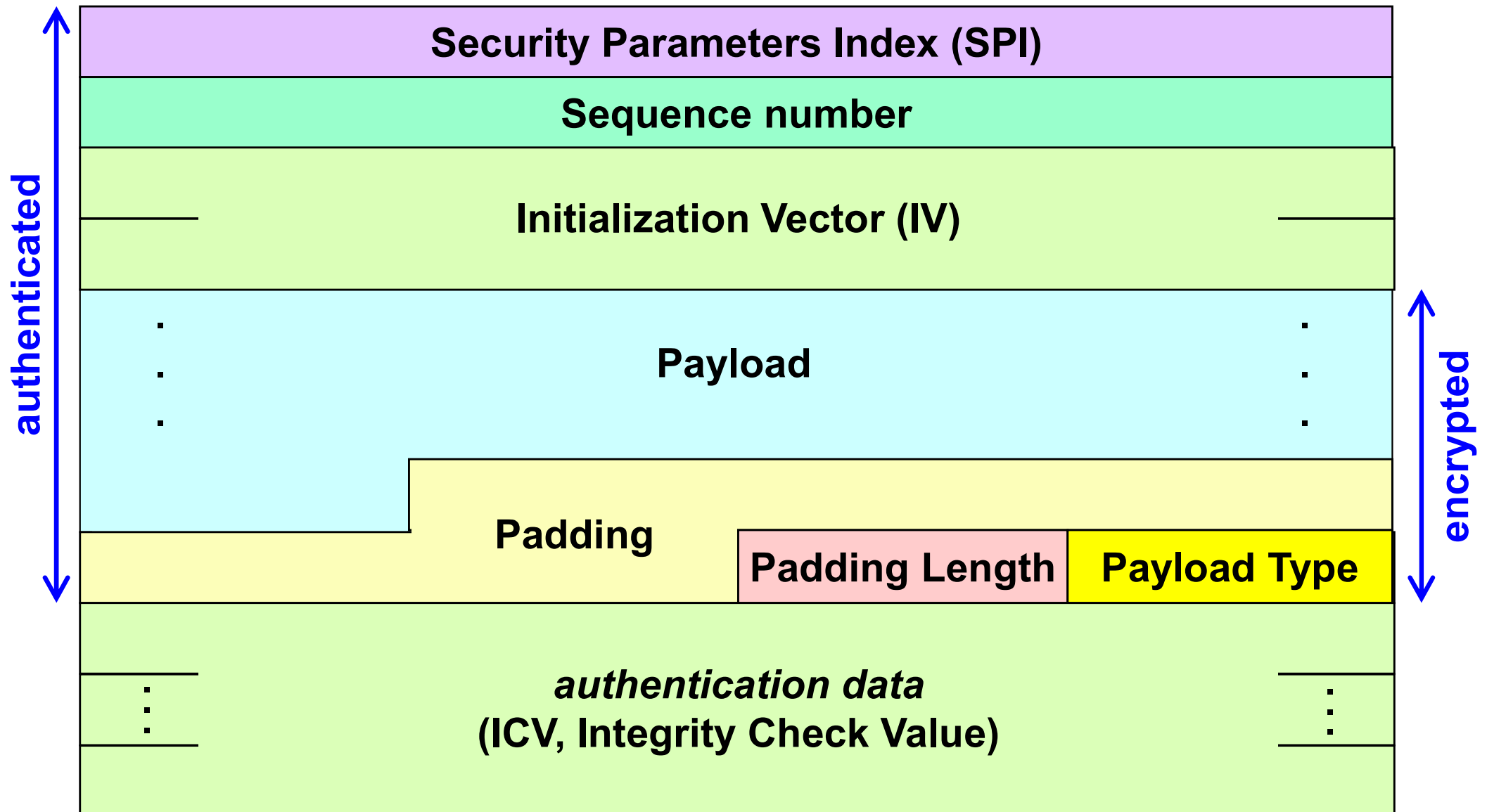
- pro: hides both the payload and (original) header
- con: larger packet size



ESP - format (RFC-2406)



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

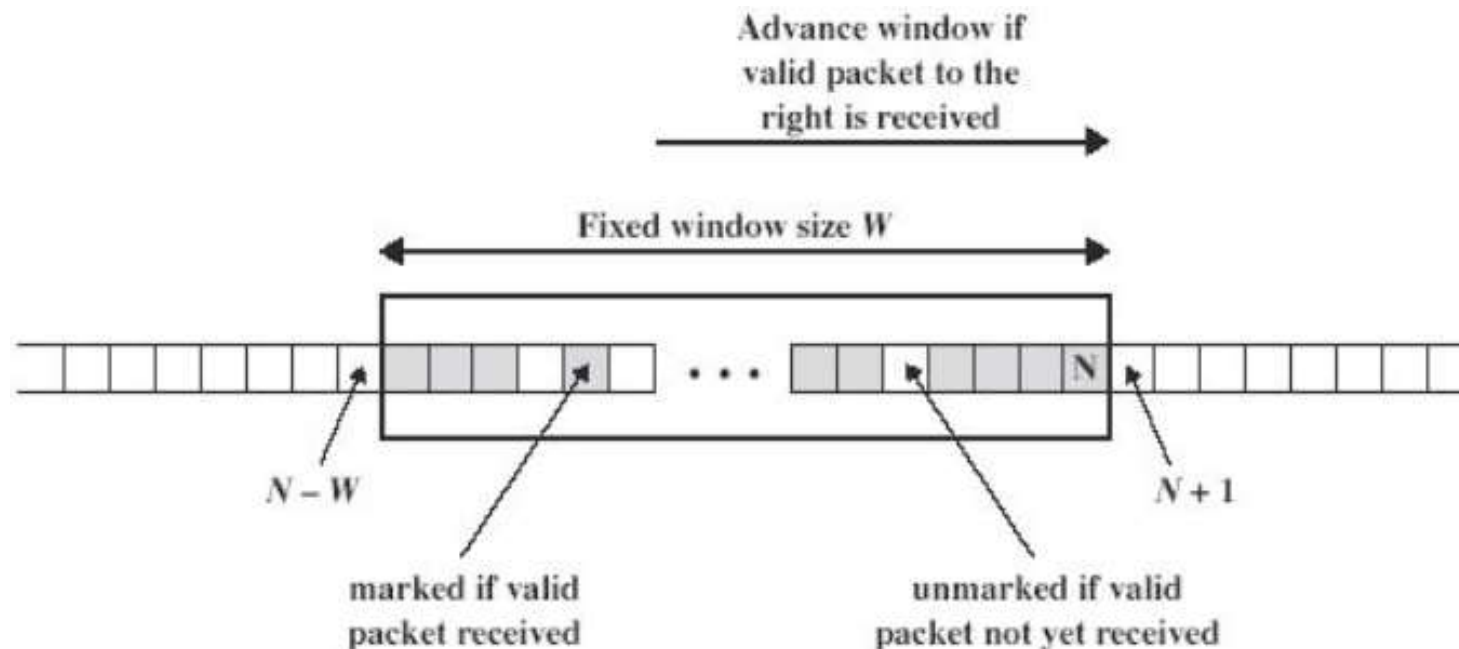


IPsec implementation details

- **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 privacy (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 established, sender initializes sequence number to 0
- when sending a packet, increment the sequence number
- when the sequence number $2^{32}-1$ is reached, a new SA should be negotiated



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 privacy:**
 - (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

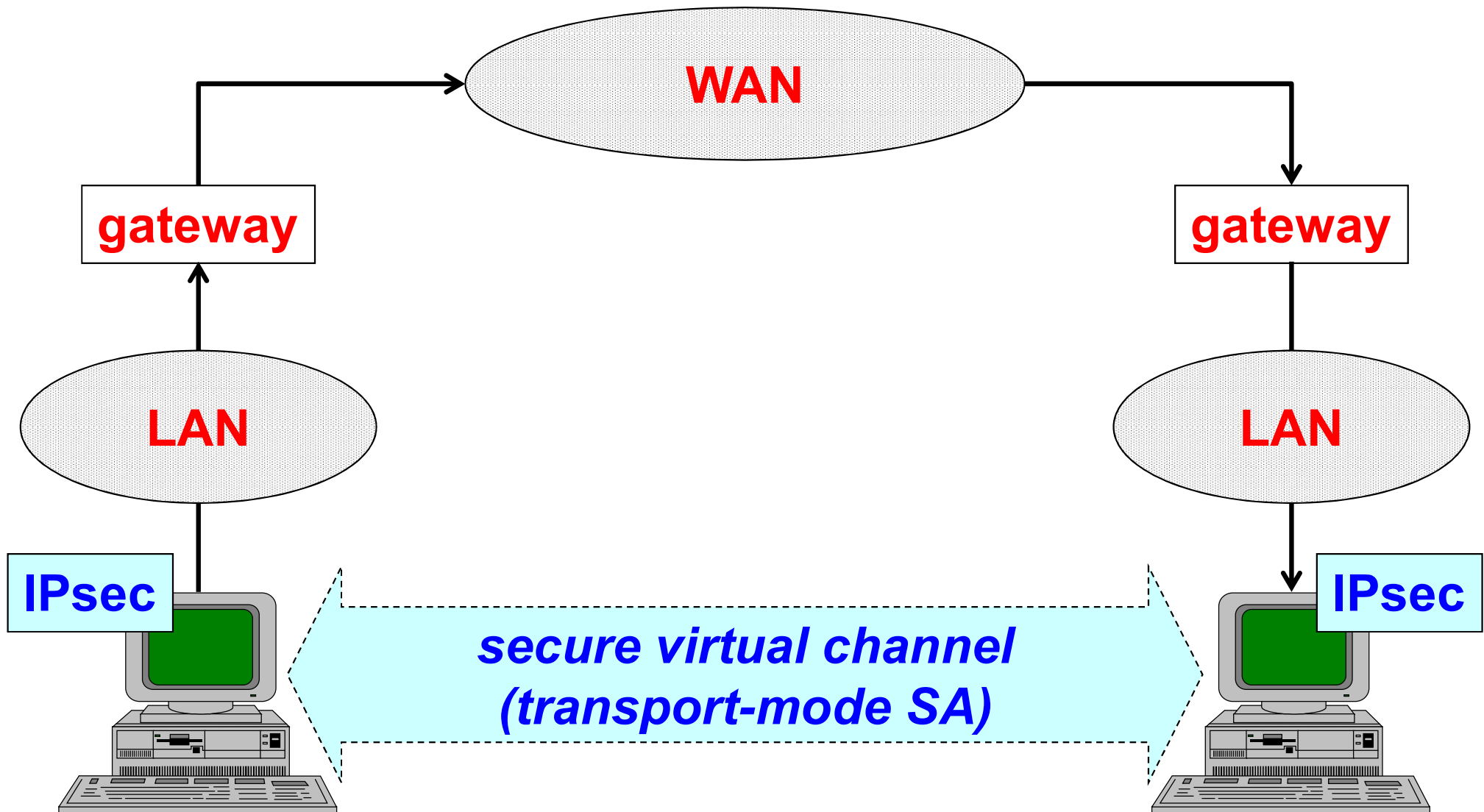
- **for authentication and integrity:**

- HMAC-SHA-256-128
- HMAC-SHA-384-192
- HMAC-SHA-512-256

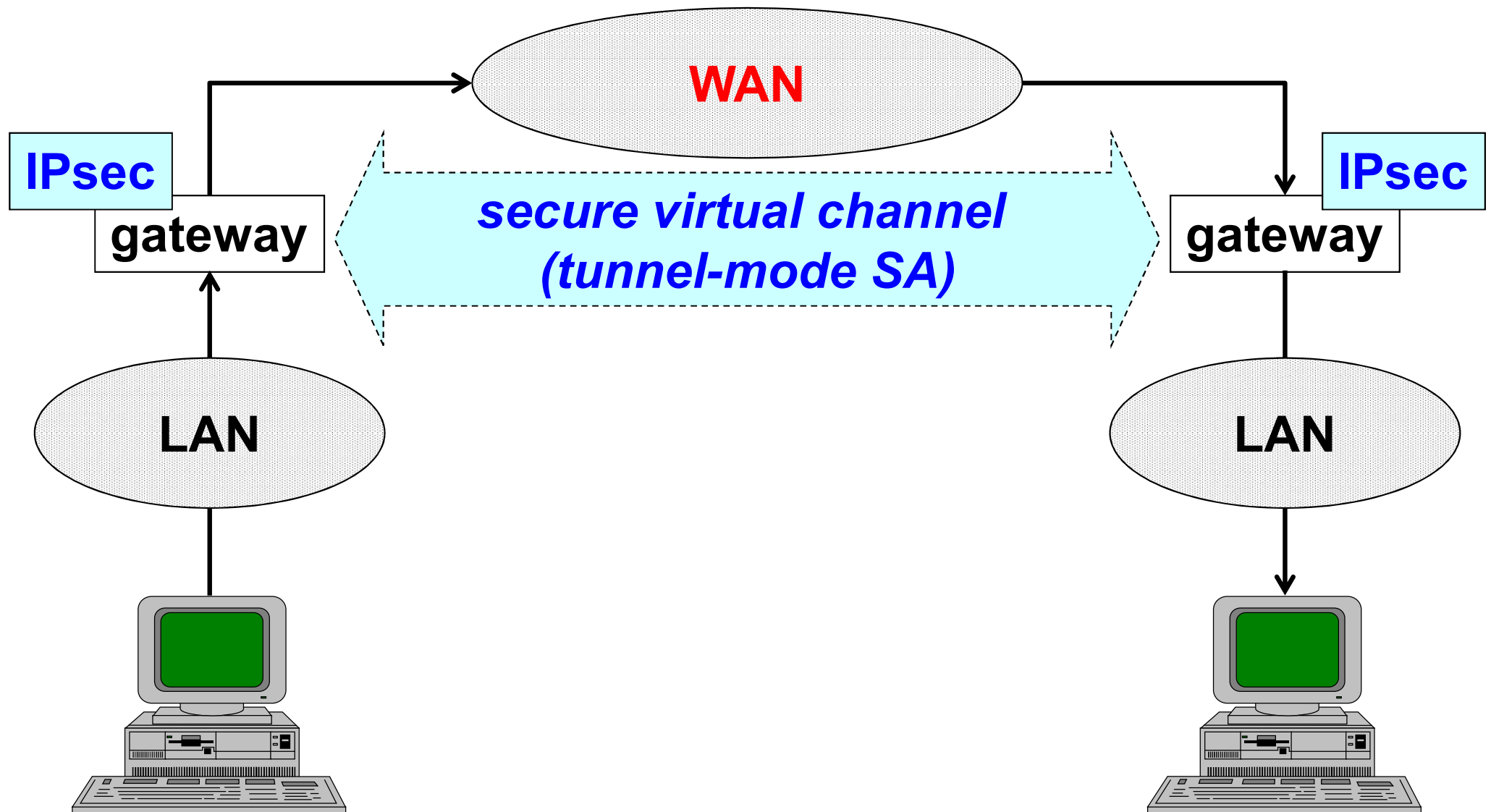
IPsec v3 – TFC

- **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 ...

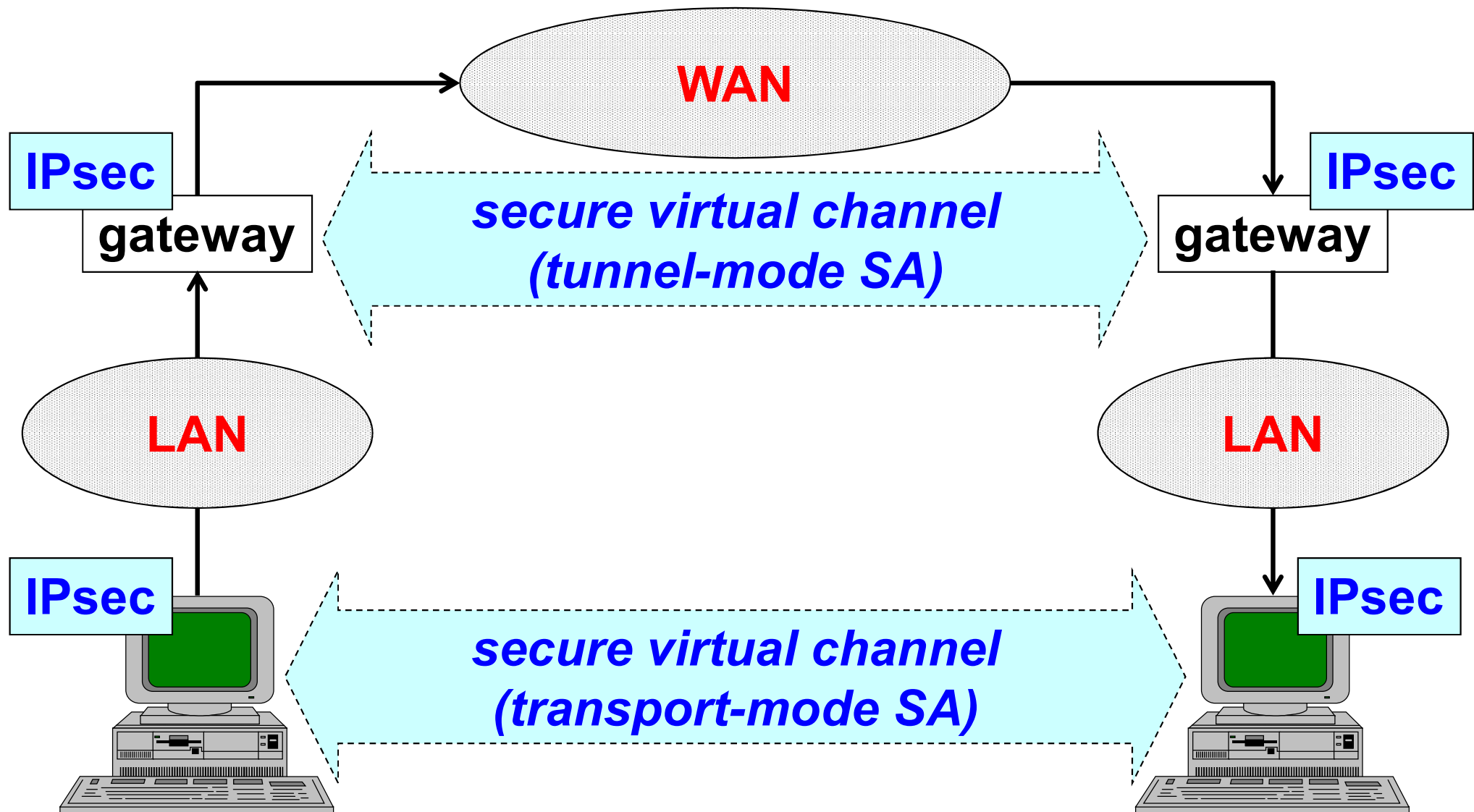
End-to-end security



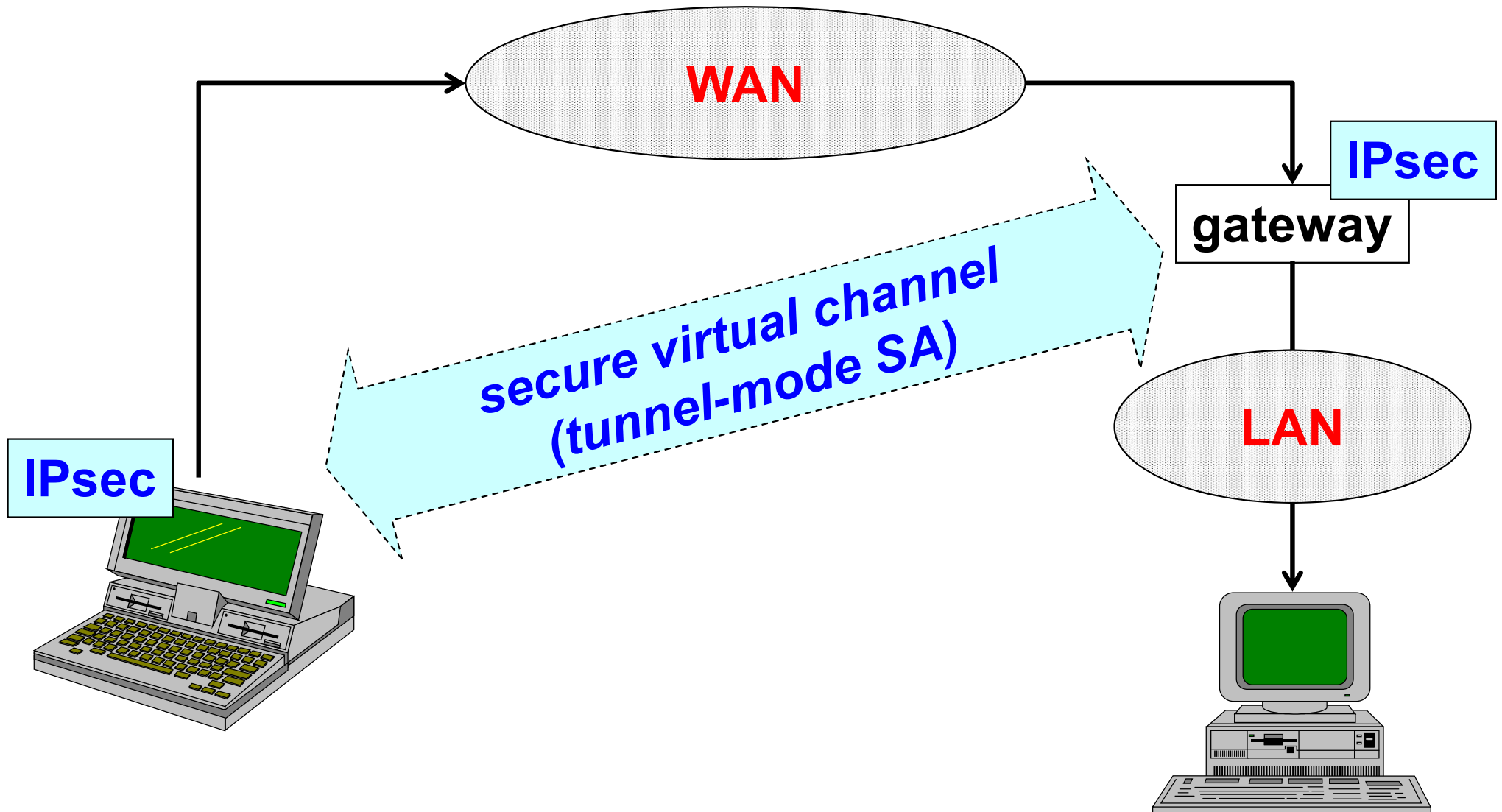
Basic VPN



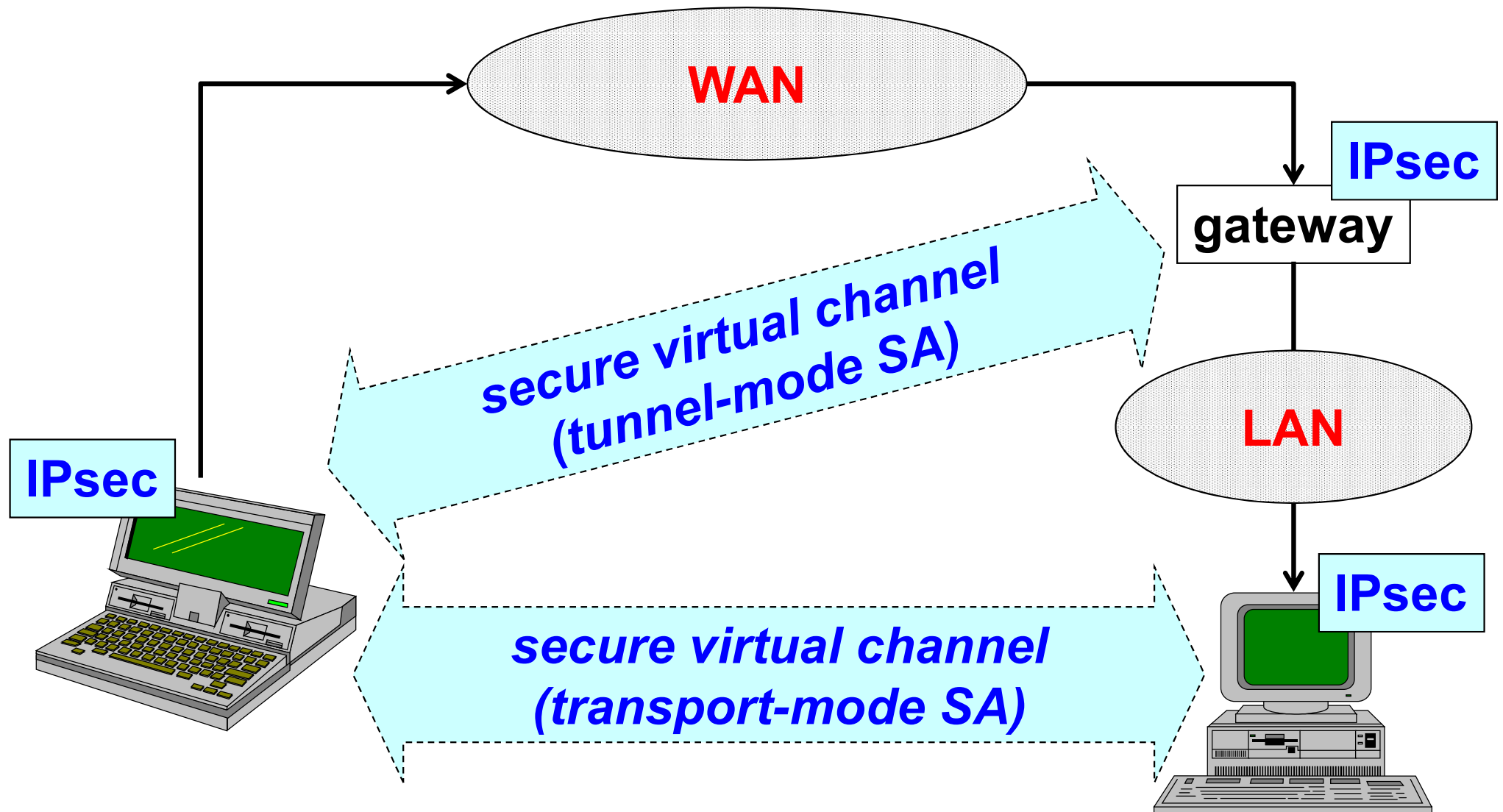
End-to-end security with basic VPN



Secure gateway



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?)

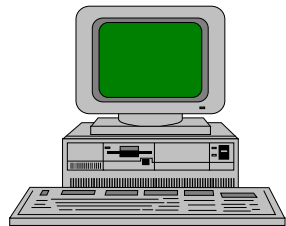
ISAKMP

- **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**

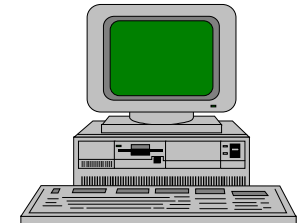
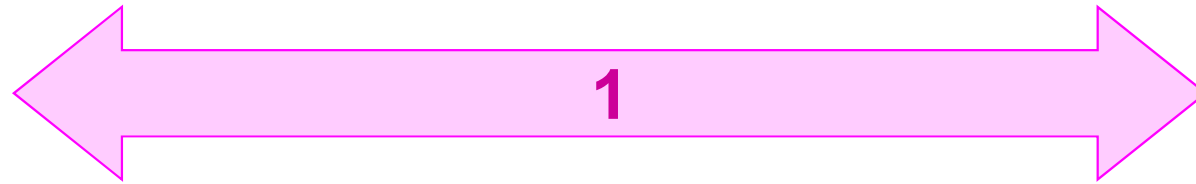
IKE

- Internet Key Exchange (RFC-2409)
- = ISAKMP + OAKLEY
- creation of a SA to protect the ISAKMP exchange
- this SA is used to protect the negotiation of the SA needed by IPsec traffic
- the same ISAKMP SA may be reused several times to negotiate other IPsec SA

IKE: operations

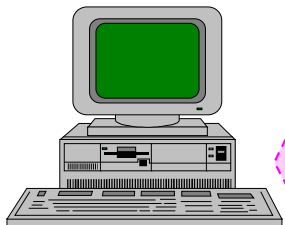


initiator

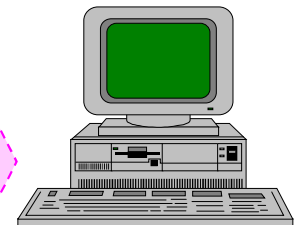
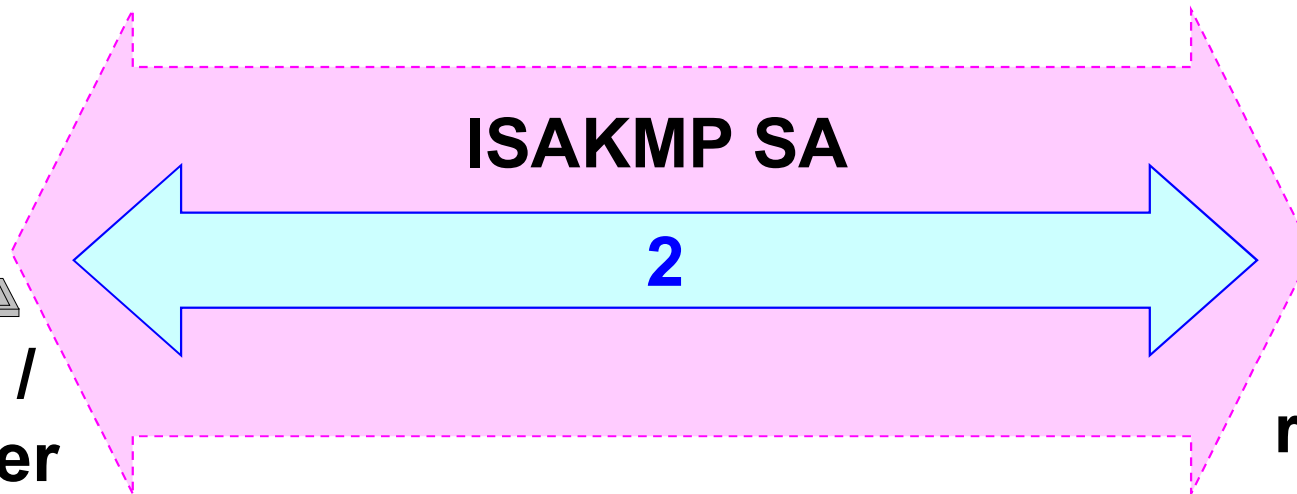


responder

**IKE phase 1 - negotiation of a bidirectional ISAKMP SA:
“main mode” or “aggressive mode”**



**initiator /
responder**



**initiator /
responder**

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)**

System requirements for IPsec

- **on router:**
 - powerful CPU or crypto accelerator
 - not managed in outsource
- **on firewall:**
 - powerful CPU
- **on VPN concentrator:**
 - maximum independence from the other security measures

IPsec influence on performance

- **network throughput is reduced:**
 - larger packet size
 - transport mode AH: +24 bytes
 - transport mode ESP-DES-CBC: ≥ 32 bytes
 - larger number of packets (for SA activation)
- **usually reduction is not very large**
- **exception: point-to-point link that used L2 compression that now becomes useless or counterproductive when applied to ESP packets**
- **possible compensation via IPComp (RFC-3173) or application-level compression**

Applicability of IPsec

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

IP (in)security

- **addresses are not authenticated**
- **packets are not protected:**
 - integrity
 - authentication
 - confidentiality
 - replay
- **therefore all protocols using IP as carrier can be attacked, mainly relevant for the “service” protocols (i.e. the non-application ones, such as ICMP, IGMP, DNS, RIP, ...)**

ICMP security

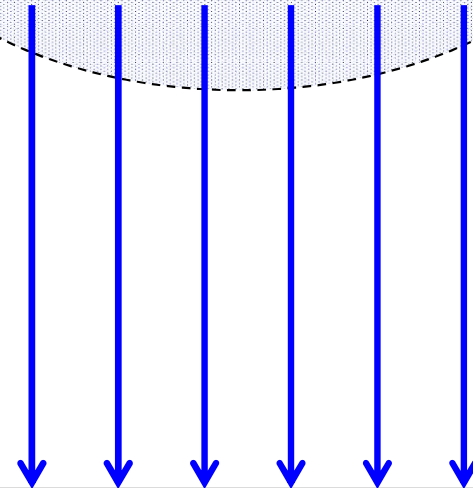
- **Internet Control and Management Protocol**
- **vital for network management**
- **many attacks are possible because it has no authentication**
- **ICMP functions:**
 - echo request / reply
 - destination unreachable (network / host / protocol / port unreachable)
 - source quence
 - redirect
 - time exceeded for a datagram

Smurfing attack

src = A
dst = X.Y.255.255
ICMP echo request



reflector
(network X.Y)



ultimate target
(A)

Anti-smurfing countermeasures

- for external attacks: reject IP broadcast packets at your border

```
interface serial0  
no ip directed-broadcast
```

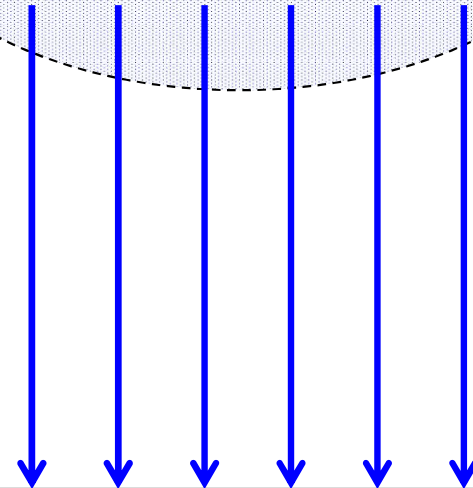
- for internal attacks: identify the attacker via network management tools

Fraggle attack

src = A
dst = X.Y.255.255
UDP echo request



reflector
(network X.Y)

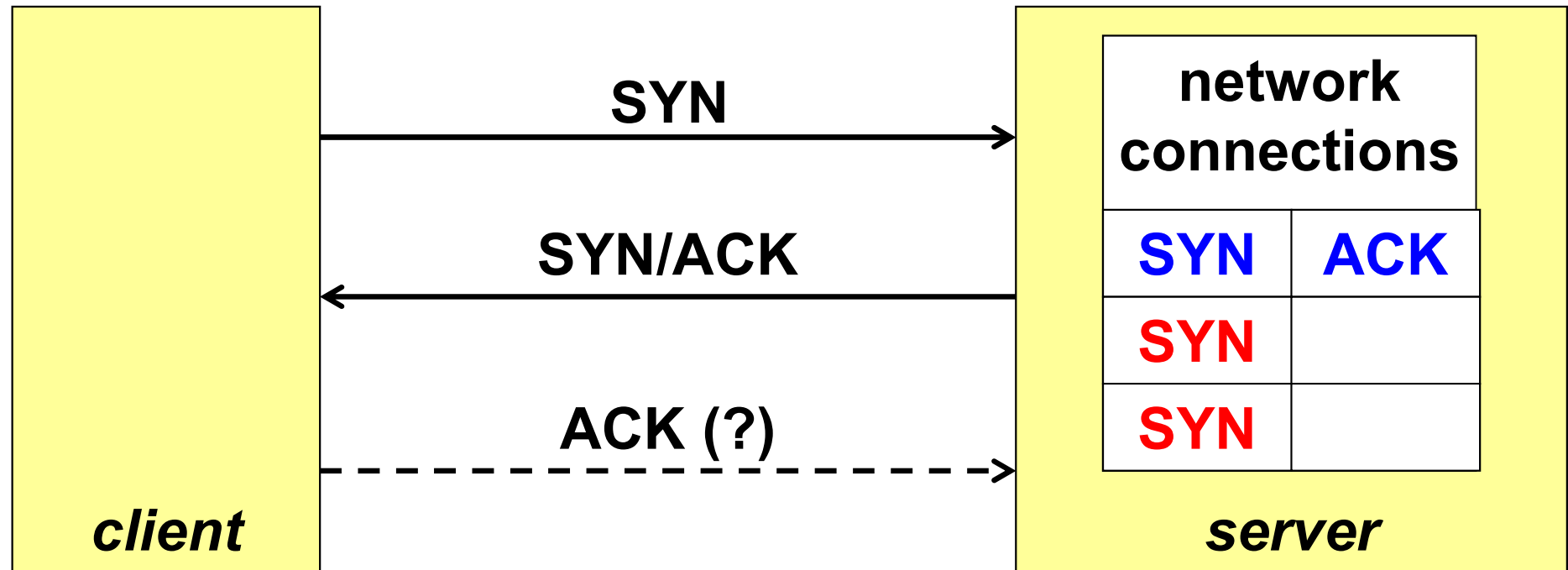


ultimate target
(A)

ARP poisoning

- **ARP = Address Resolution Protocol (RFC-826)**
 - used to discover the L2 address of a node when knowing its L3 address
 - result stored in the ARP table
- **ARP poisoning:**
 - nodes accept ARP reply without ARP request
 - nodes overwrite static ARP entries with the dynamic ones (obtained from ARP reply)
 - the “ar\$sha” ARP field (sender hw address) may differ from the src field in the 802.3 packet
 - used by attack tools (e.g. Ettercap)

TCP SYN flooding

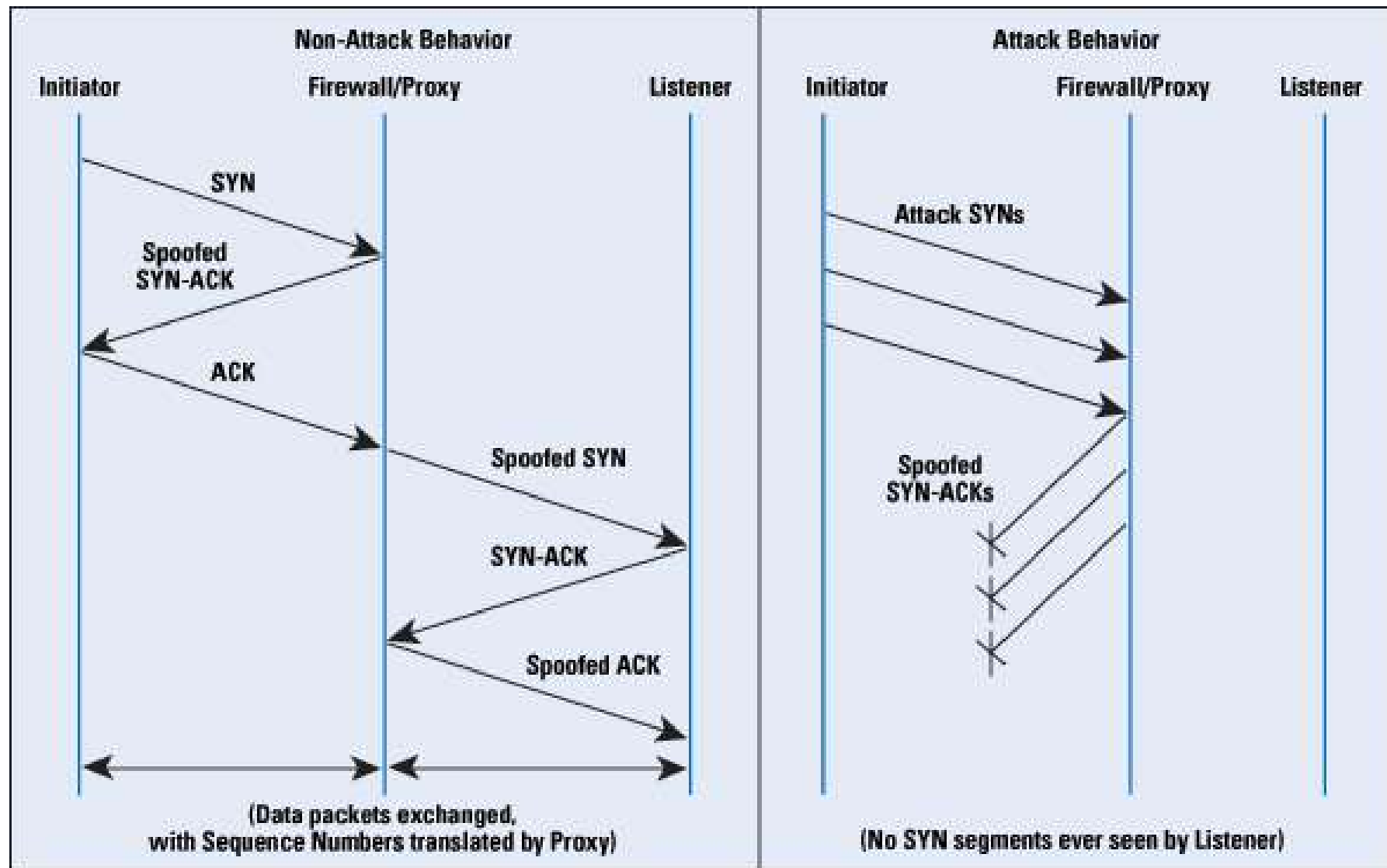


- multiple requests with IP spoofing
- the connection table is saturated until half-open connections timeout (typical value: 75")

Protection against SYN flooding

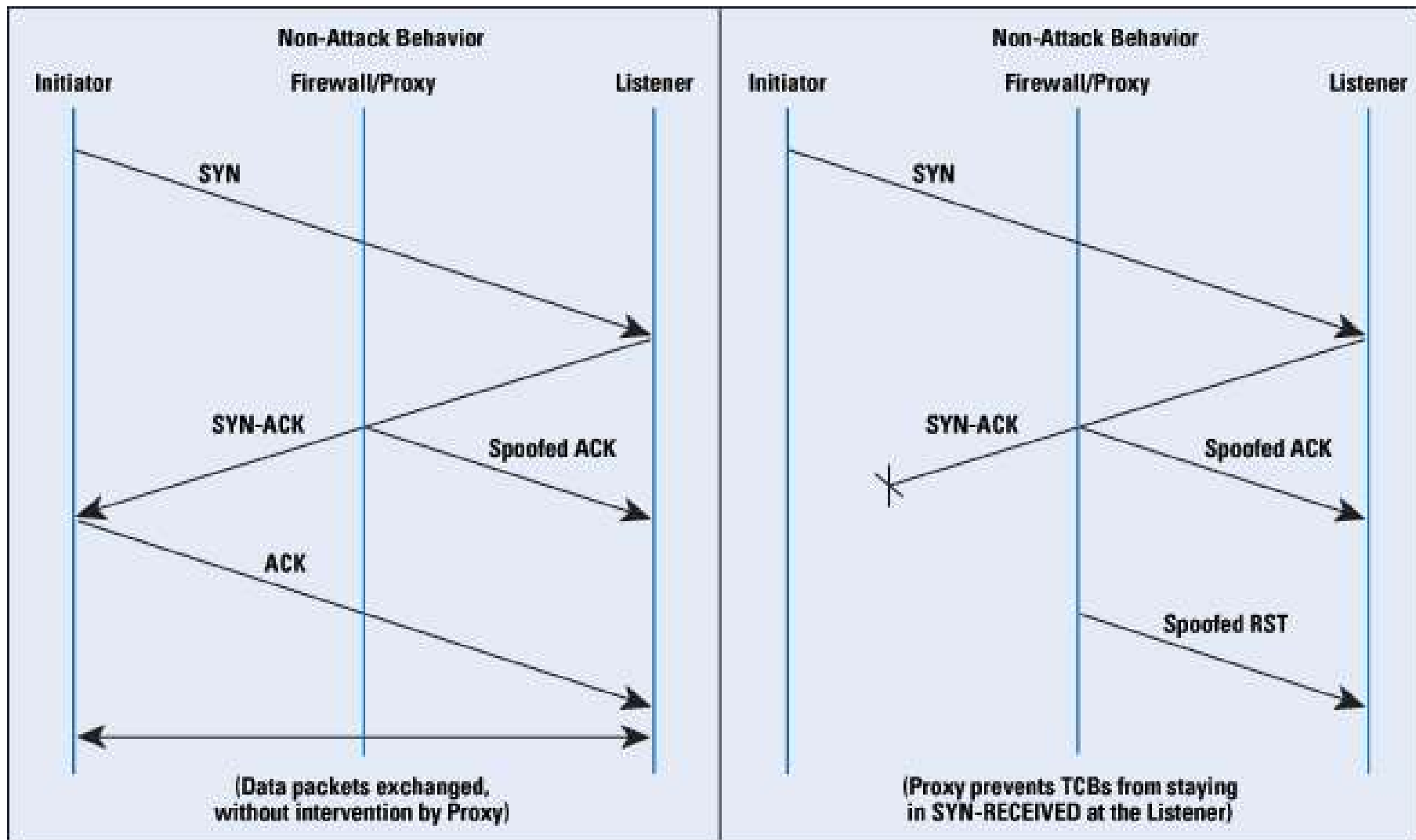
- **decrease the timeout**
 - risk to delete requests from valid but slow clients
- **increase the table size**
 - can be circumvented by sending more requests
- **use a router as “SYN interceptor”:**
 - substitutes the server in the first phase
 - if the handshake completes successfully, then transfers the channel to the server
 - “aggressive” timeout (risky!)
- **use a router as “SYN monitor”:**
 - kills the pending connection requests (RST)

SYN interceptor (or firewall relay)



http://www.cisco.com/web/about/ac123/ac147/archived_issues_syn_flood_attacks.html

SYN monitor (or firewall gateway)



http://www.cisco.com/web/about/ac123/ac147/archived_issues_syn_flood_attack.html

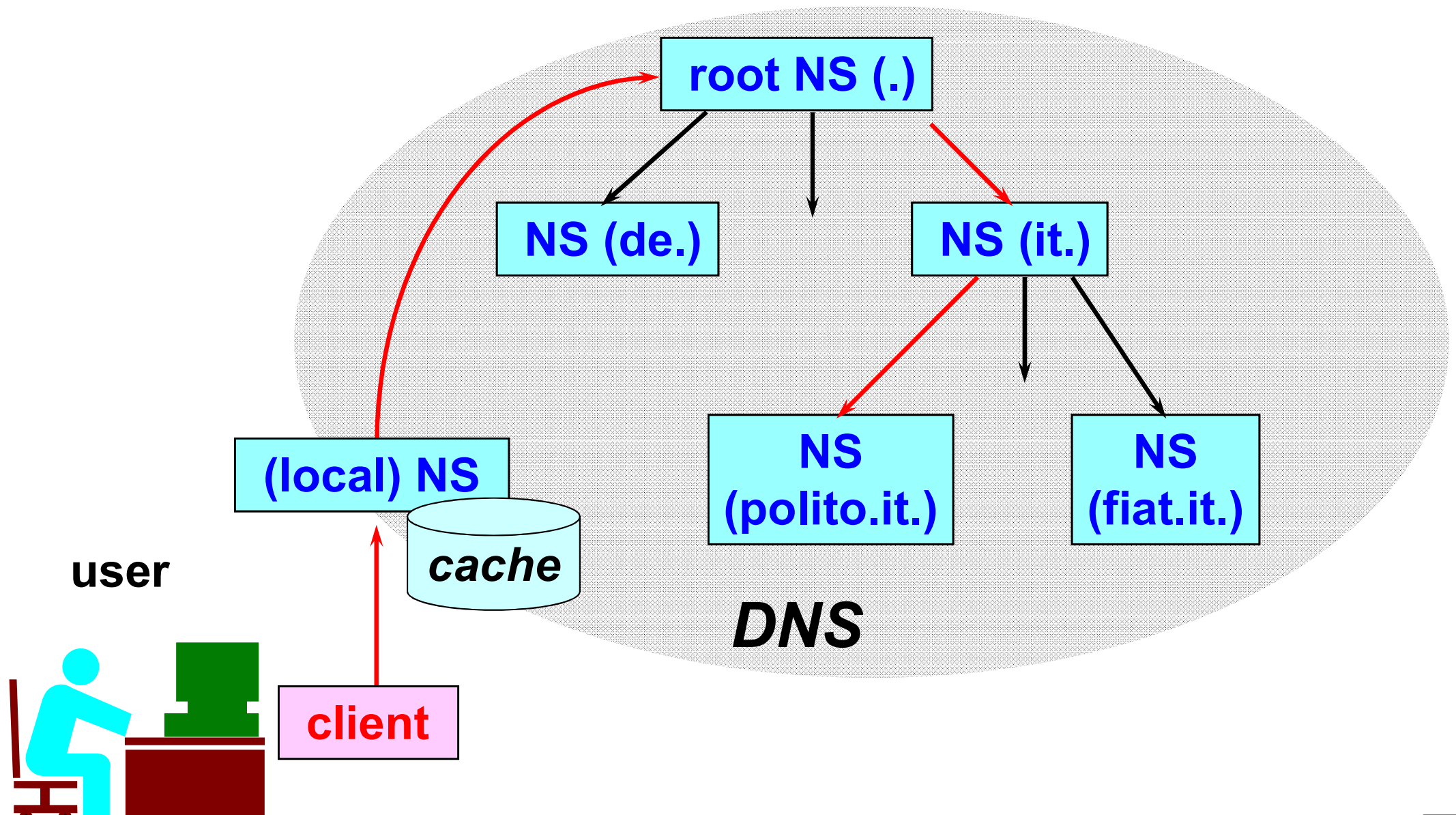
SYN cookie

- idea of D.J.Bernstein (<http://cr.yp.to>)
- the only approach really effective to completely avoid the SYN flooding attack
- uses the TCP sequence number of the SYN-ACK packet to transmit a cookie to the client and later recognize the clients that already sent the SYN without storing any info about them on the server
- available on Linux and Solaris

DNS security

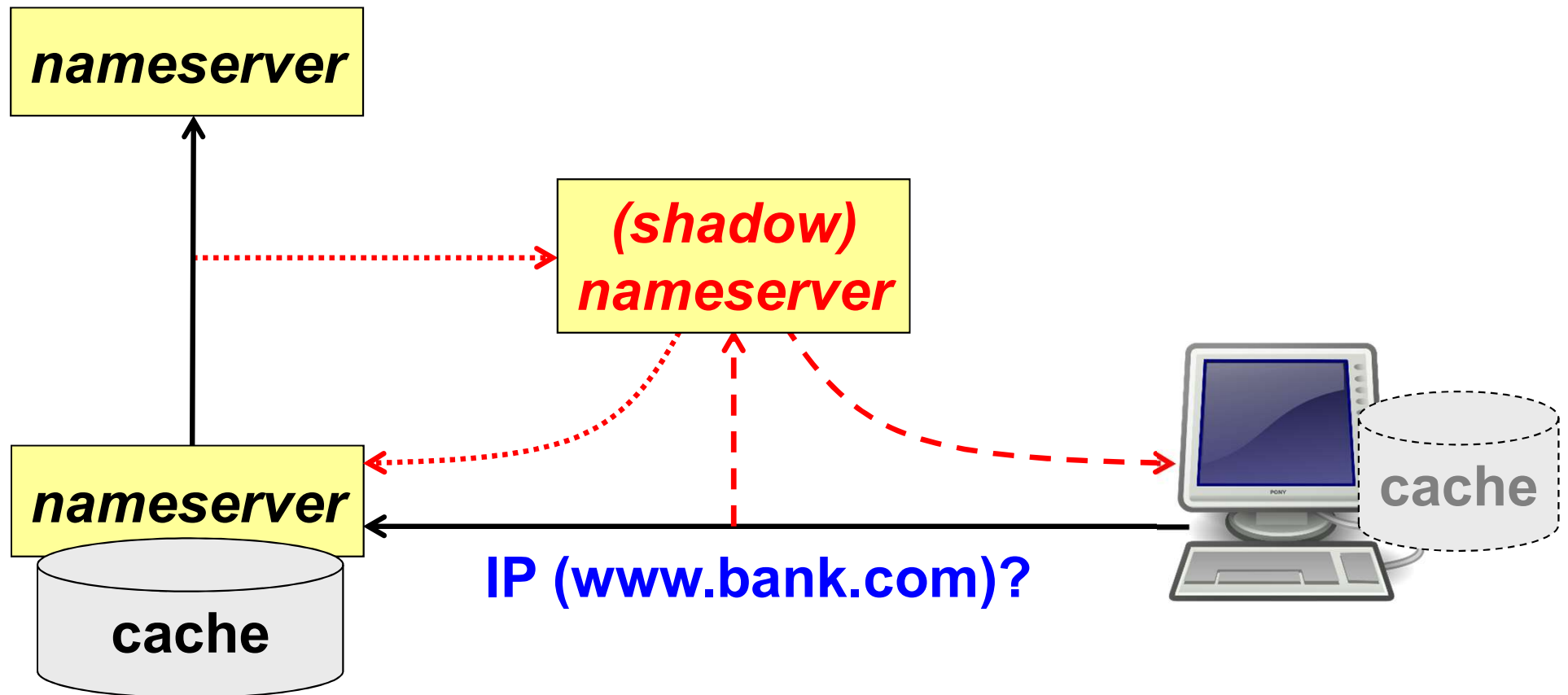
- **DNS (Domain Name System)**
- **translation:**
 - from names to IP addresses
 - from IP addresses to names
- **vital service**
- **queries over port 53/UDP**
- **zone transfers over port 53/TCP**
- **no security**
- **DNS-SEC under development**

DNS architecture



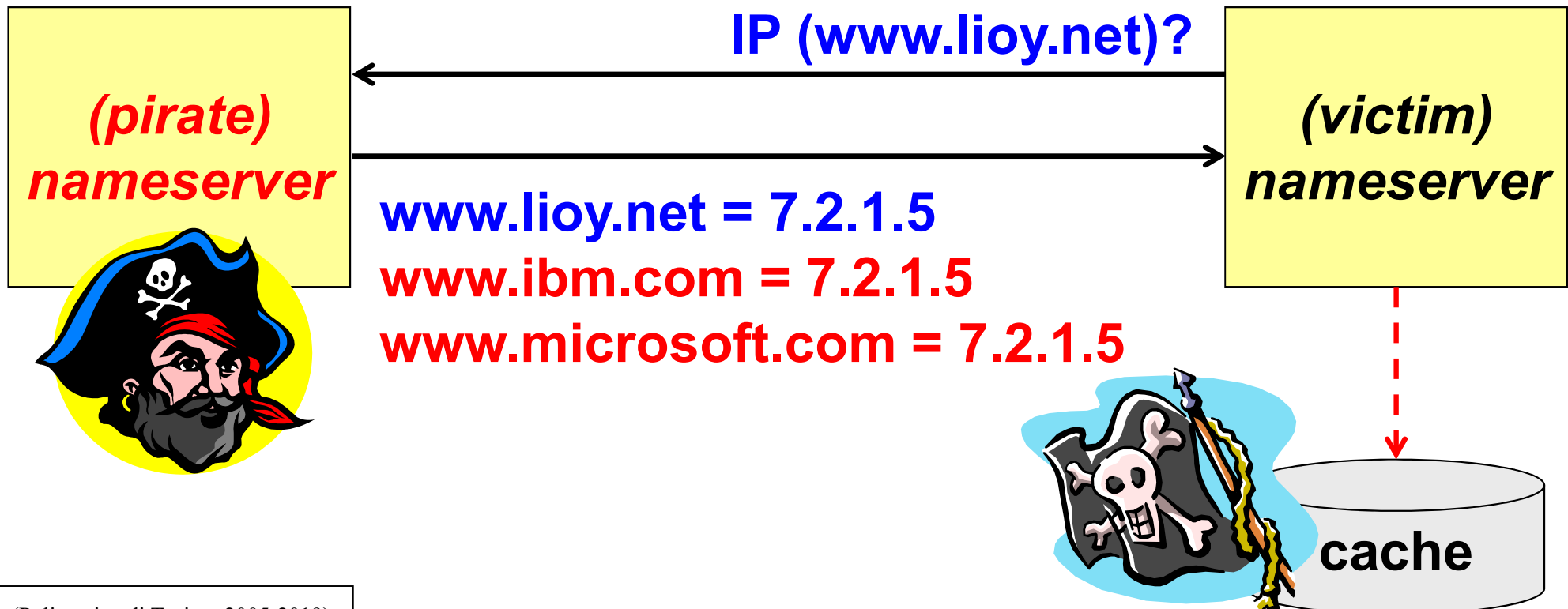
DNS shadow server

- sniffing to intercept the queries
- spoofing to generate fake answers (DoS or traffic redirection to fake sites)



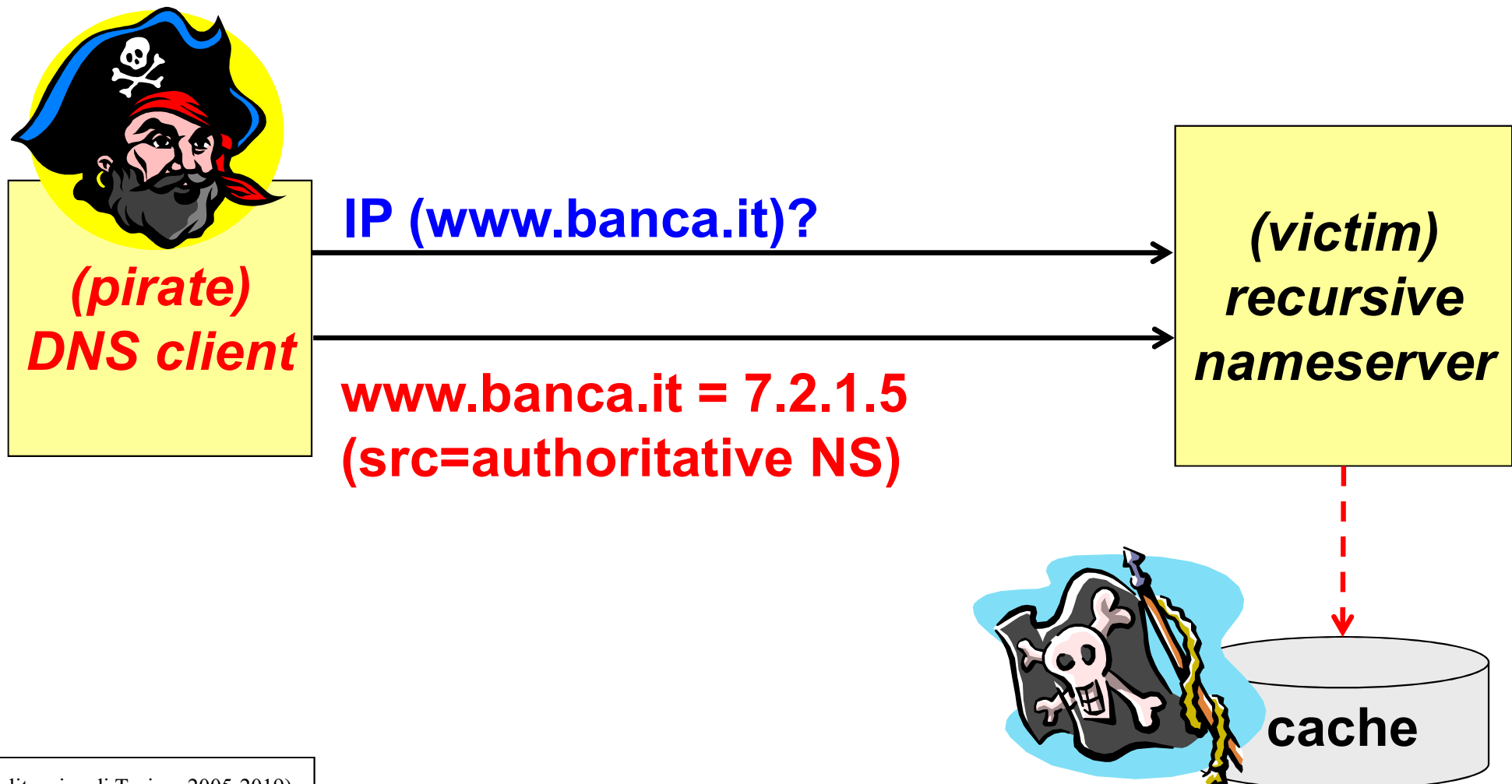
DNS cache poisoning

- attract the victim to make a query on my NS
- provide answers also to queries never done to push / overwrite the victim's cache

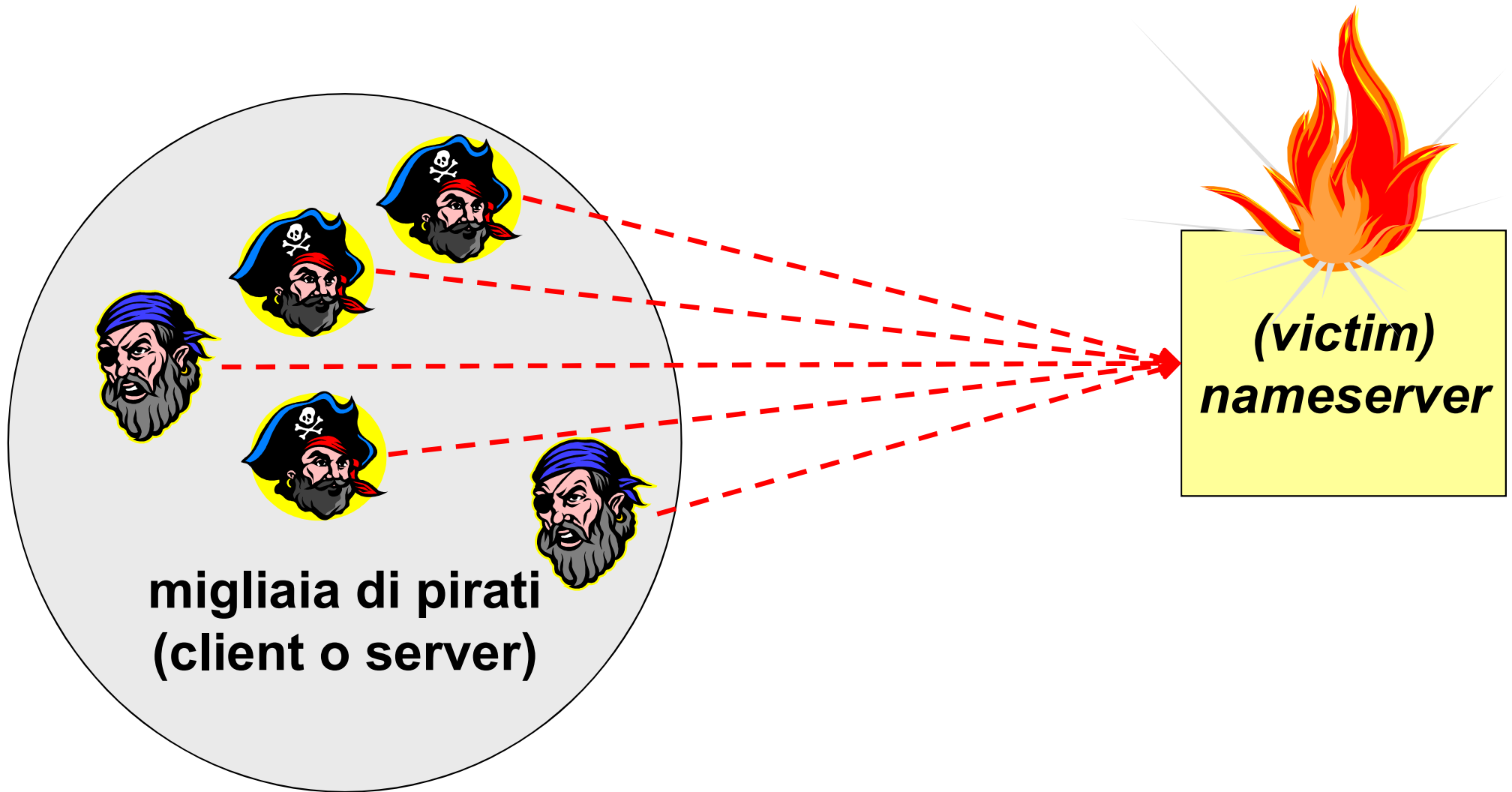


DNS cache poisoning (2nd version)

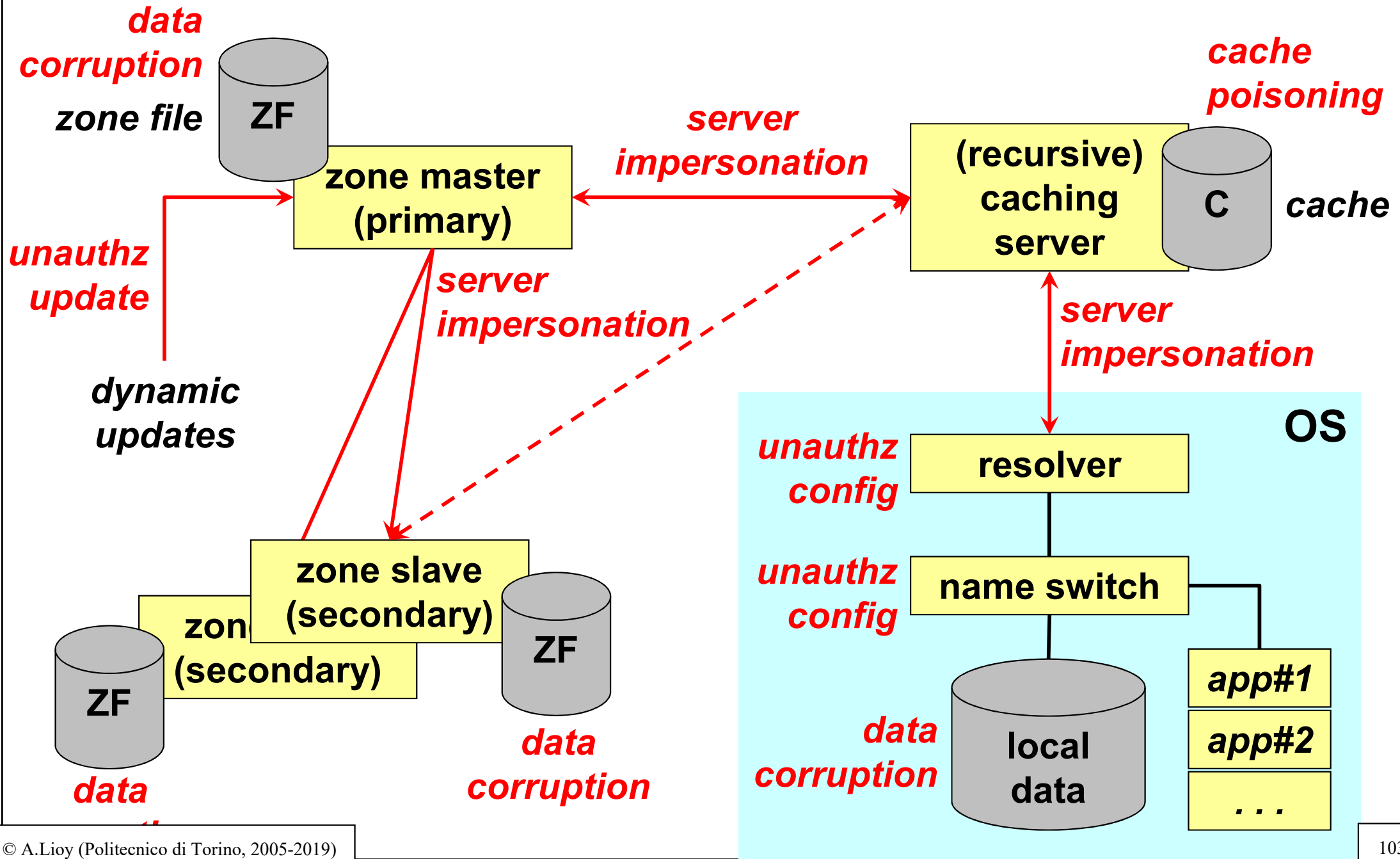
- make a query and self-provide the (wrong) answer too, to insert it into the victim's cache



(DNS) flash crowd



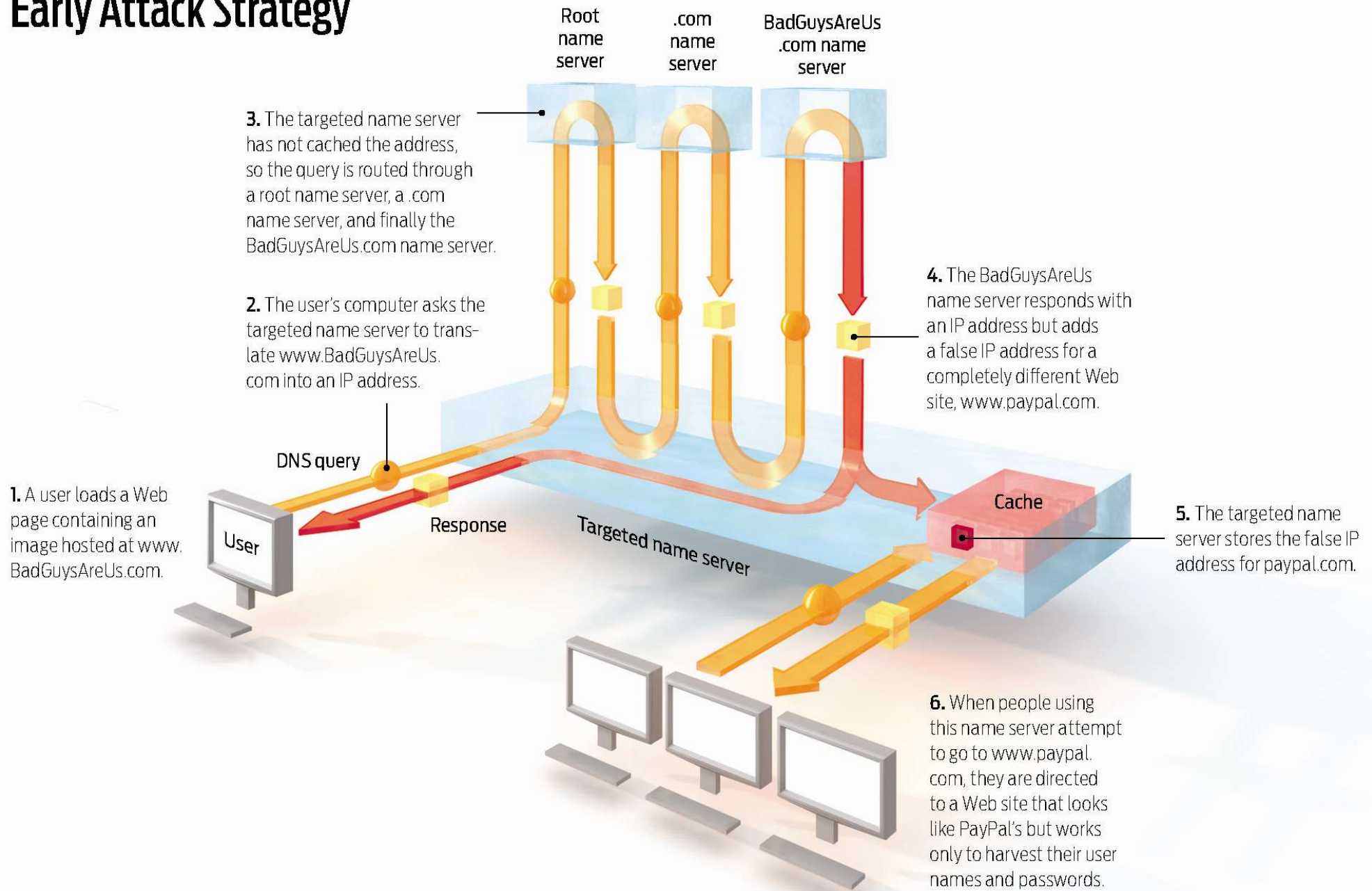
Name-address translation



DNSsec is needed

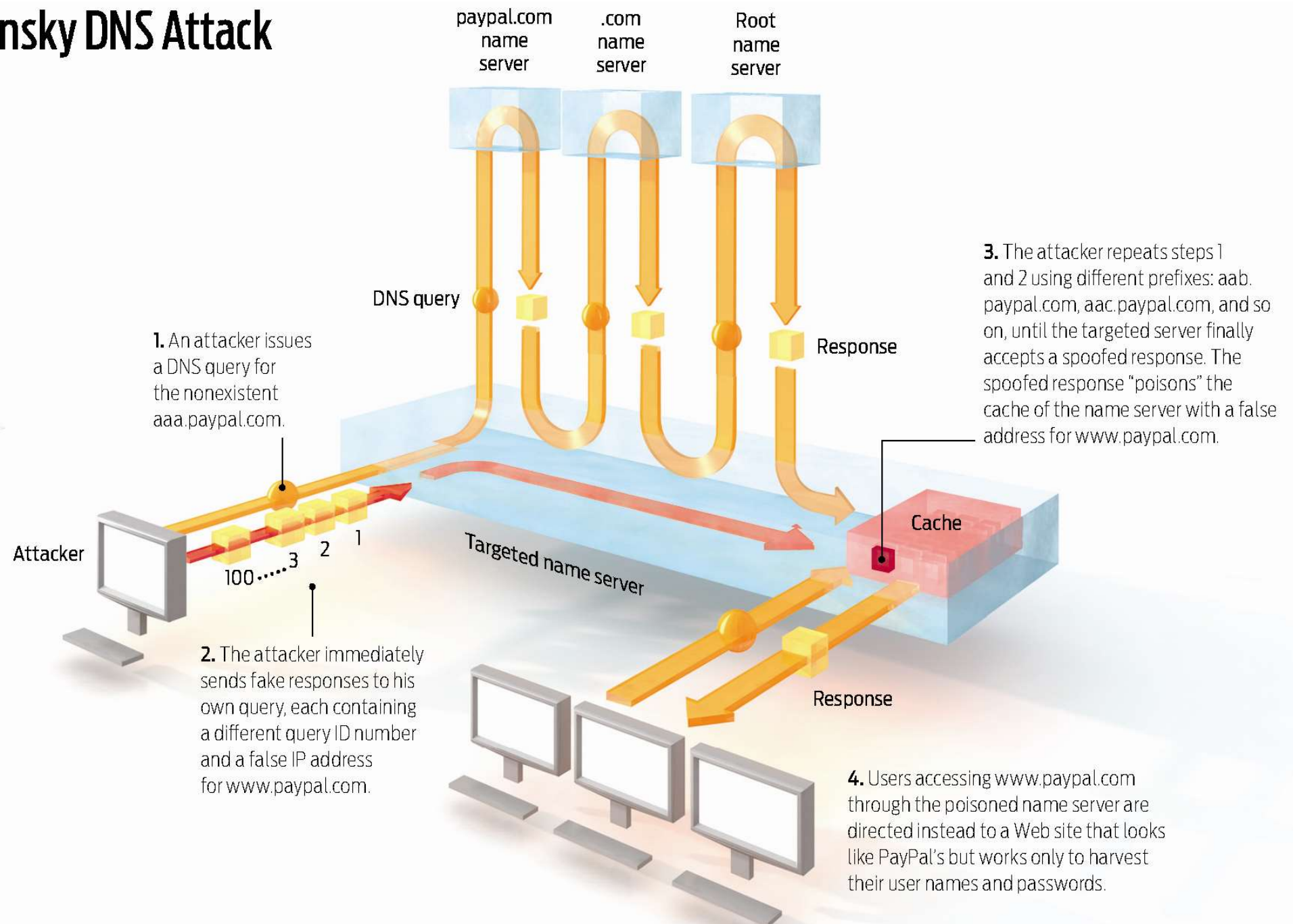
- **(feb'08) Dan Kaminsky finds a new attack that makes cache poisoning**
 - simpler to execute
 - more difficult to avoid
 - applicable also to the 1st level NS records (e.g. com)
- **(jul'08) first advisories and patches**
- **(ago'08) talk by Kaminsky at Black Hat '08**
- **(sep'08) USA makes compulsory use of DNSsec for the .gov domain starting january 2009**

Early Attack Strategy



"Fresh Phish", IEEE Spectrum, oct'08, doi: 10.1109/MSPEC.2008.4635052

Kaminsky DNS Attack



"Fresh Phish", IEEE Spectrum, oct'08, doi: 10.1109/MSPEC.2008.4635052

DNSsec

- **digital signature of DNS records**
 - who is "authoritative" for a certain domain?
 - which is the PKI? (certificates, trusted root CA)
- **complex management of the DNS infrastructure**
 - hierarchical and delegated signatures
 - distributed signatures
- **handling of non-existent names?**
 - the ABSENCE of a record must be signed too
 - this requires sorting of the records

Some issues with DNSsec

- **no signature of the DNS query**
- **no root CA, level 1 keys distributed OOB**
- **no security in the dialogue between the DNS client and DNS (local) server**
 - use IPsec, TSIG or SIG(0)
- **signature to be performed by the DNS server**
 - computational overhead
 - management overhead (on-line secure crypto host)
- **bigger record size**
- **scarce experimental results**
 - configuration? performance?

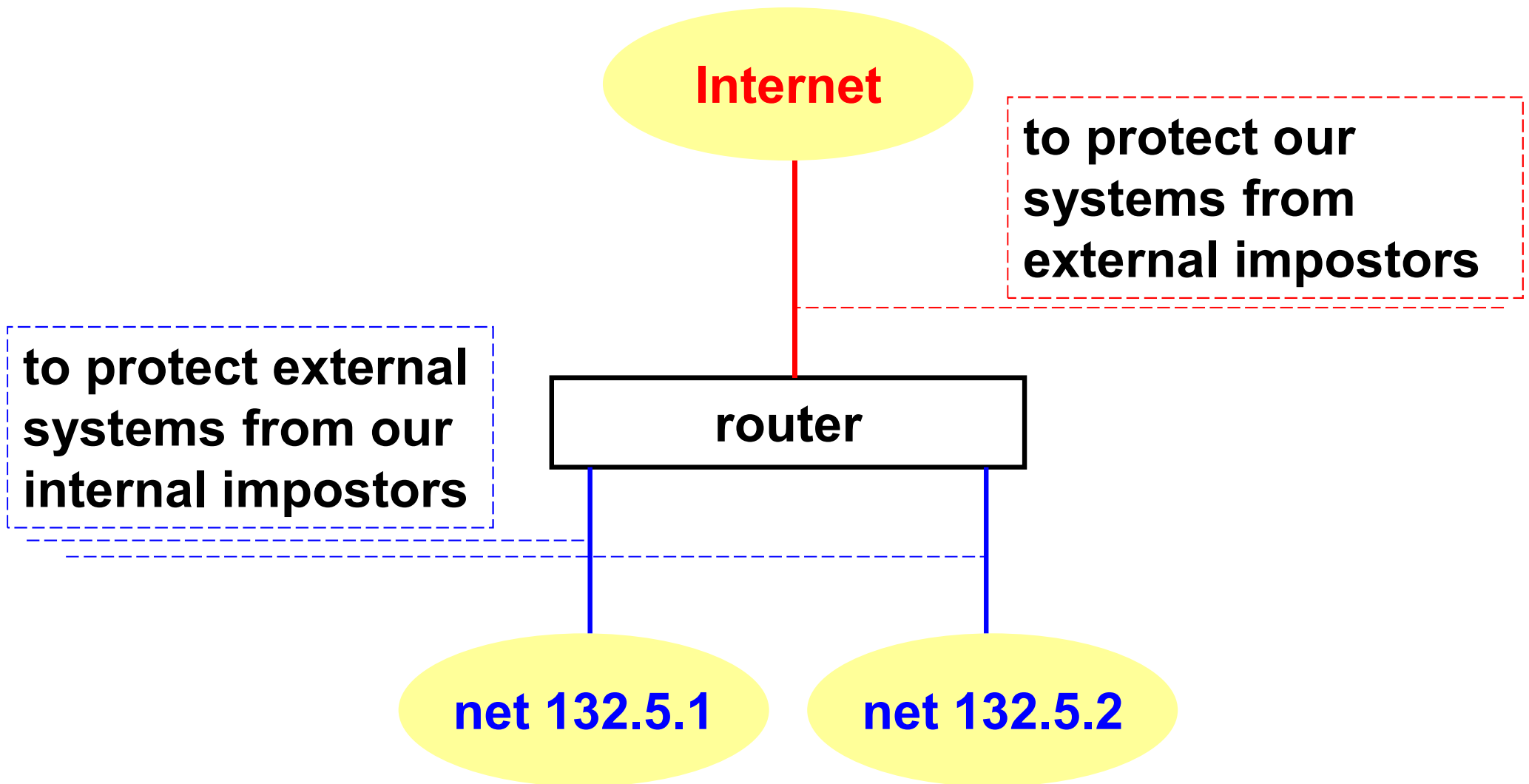
Routing security

- **low security in the system access to routers for management (telnet, SNMP)**
- **low security in the exchange of routing tables:**
 - authentication based on IP addresses
 - optional protection with a keyed-digest
 - a shared-key is required!
 - key-management is required!
- **dynamical routing variations also on end-nodes possible via ICMP**

Protection from IP spoofing

- to protect ourselves from external impostors
- also to protect the external world from our internal impostors (=net-etiquette)
- RFC-2827 “Network ingress filtering: defeating Denial of Service attacks which employ IP source address spoofing”
- RFC-3704 “Ingress filtering for multihomed networks”
- RFC-3013 “Recommended Internet Service Provider security services and procedures”

Filters for IP spoofing protection



Example of IP spoofing protection

```
access-list 101 deny ip
    132.5.0.0 0.0.255.255 0.0.0.0 255.255.255.255
interface serial 0
ip access-group 101 in

access-list 102 permit ip
    132.5.1.0 0.0.0.255 0.0.0.0 255.255.255.255
interface ethernet 0
ip access-group 102 in

access-list 103 permit ip
    132.5.2.0 0.0.0.255 0.0.0.0 255.255.255.255
interface ethernet 1
ip access-group 103 in
```