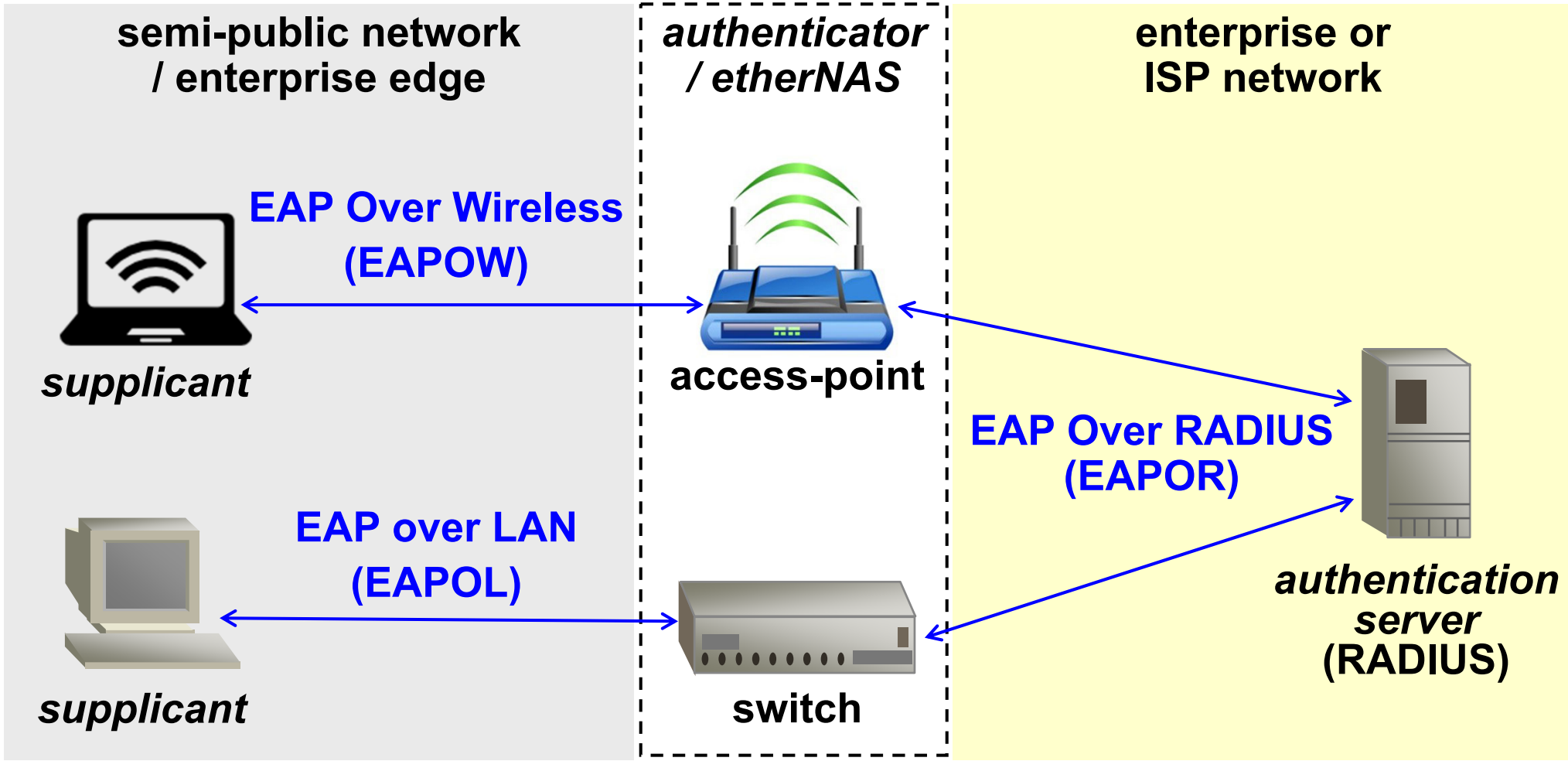
IEEE 802.1x

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

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

© A.Lioy (Politecnico di Torino, 2005-2023)

802.1x - architecture



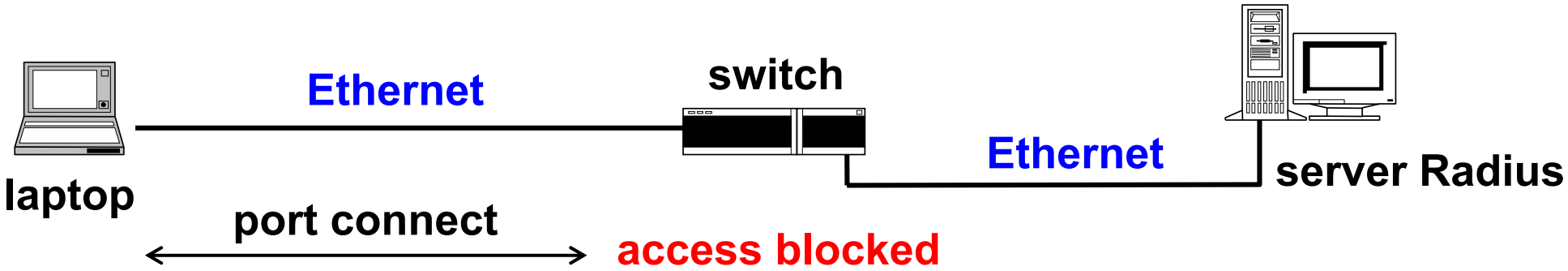
© A.Lioy (Politecnico di Torino, 2005-2023)

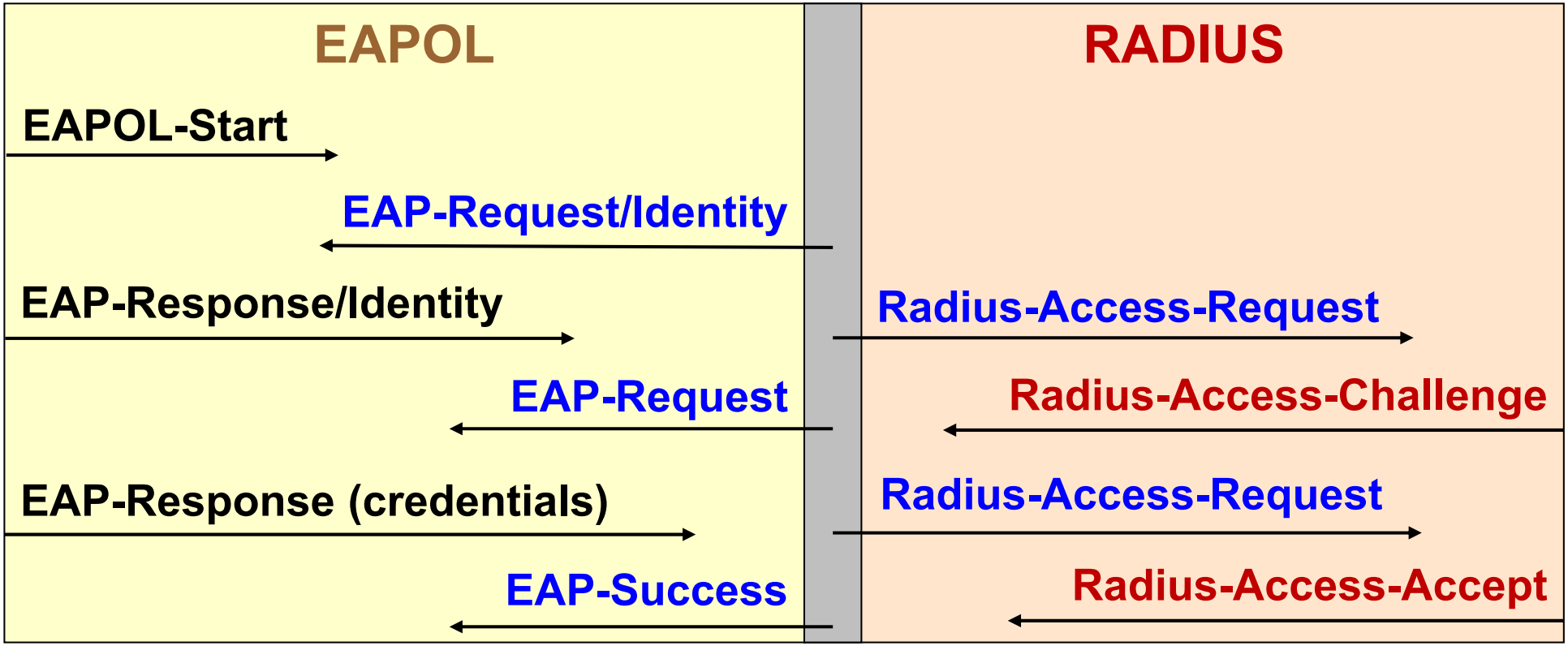
IEEE 802.1x

- **authentication and key-management framework**
- **supplicant authentication:**
  - with EAP methods, end-to-end supplicant-AS
- **key management (for supplicant and etherNAS):**
  - may derive session keys for use in packet authentication, integrity, and confidentiality
  - standard algorithms for key derivation (e.g. TLS, SRP, …)
- **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 for new mechanisms

© A.Lioy (Politecnico di Torino, 2005-2023)

802.1x - messages



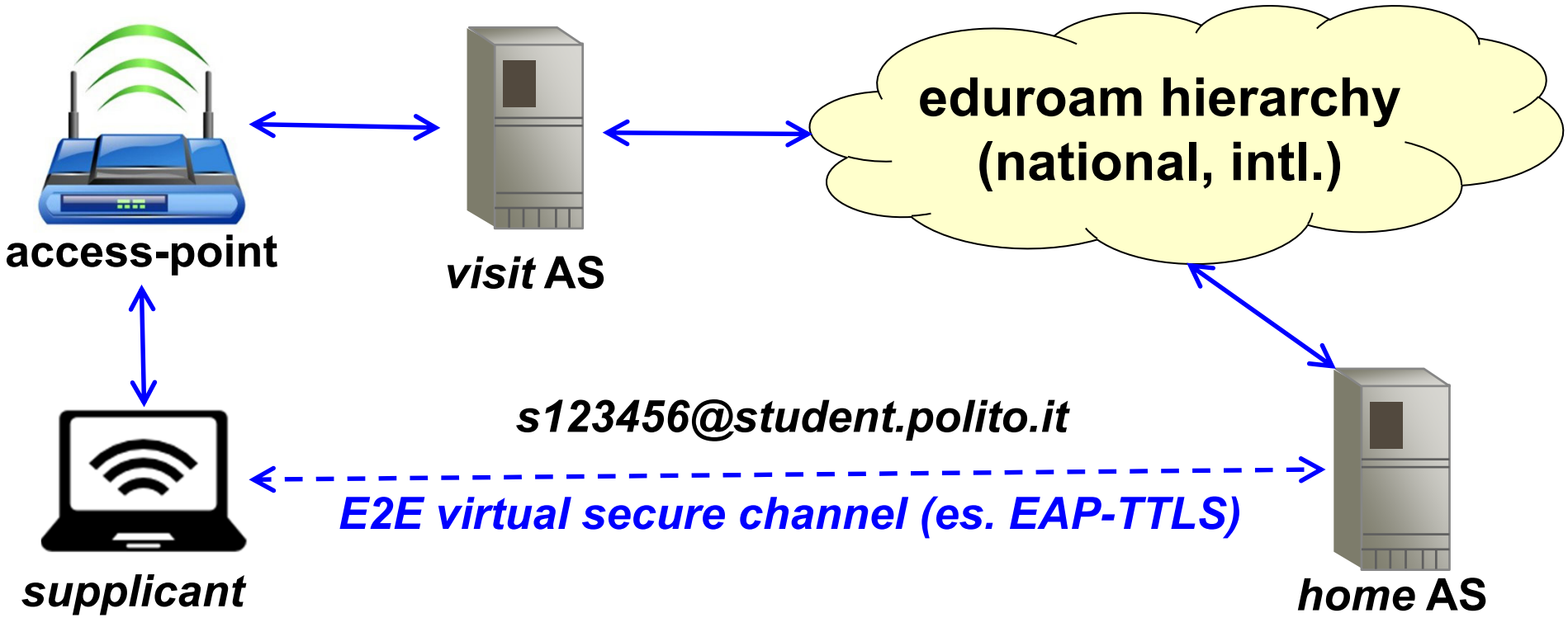


access allowed

© A.Lioy (Politecnico di Torino, 2005-2023)

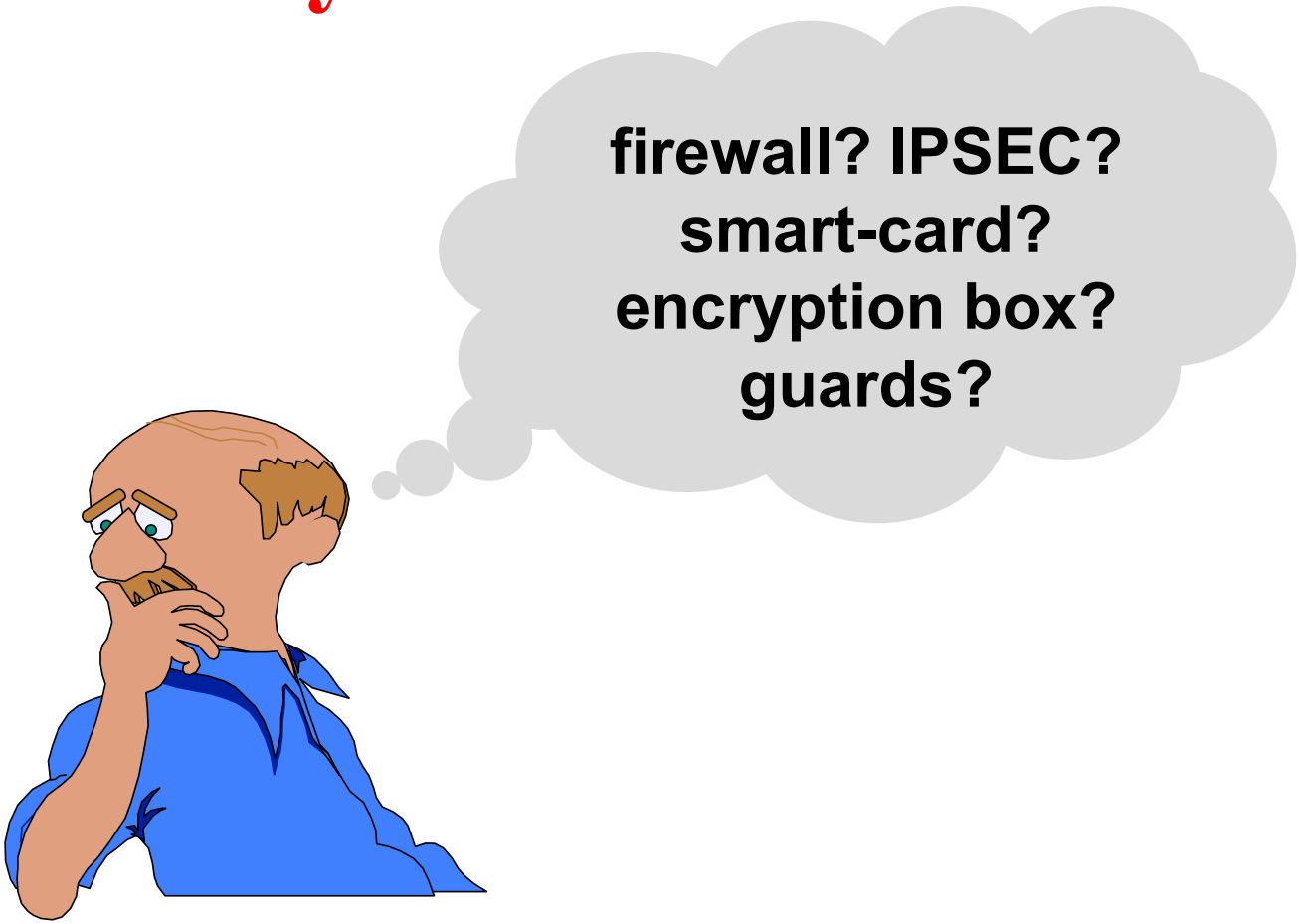
eduroam

- **WiFi access at research institutes (Italy, Europe, …) and other places (e.g some airports)**
  - (3/12/2023) 106 countries
  - uses 802.1x + RADIUS federation



© A.Lioy (Politecnico di Torino, 2005-2023)

Which is the best OSI level to implement security?



Application
Presentation
Session
Transport
Network
Data Link
Physical

© A.Lioy (Politecnico di Torino, 2005-2023)

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)

© A.Lioy (Politecnico di Torino, 2005-2023)

DHCP (in)security

- **non-authenticated (!!) broadcast (!) protocol providing:**
  - IP address, netmask, default gateway
  - local nameserver, local DNS suffix
- **activation of a fake DHCP server is trivial**
  - because the DHCP request is L2 broadcast
- **possible attacks from the fake DHCP server:**
  - denial-of-service
    - provides a wrong network configuration
  - MITM
    - provide configuration with 2-bit subnet + gw equal attacker
    - if we activate NAT, we can intercept the replies too
  - malicious name-address translation (e.g. for phishing, pharming)

© A.Lioy (Politecnico di Torino, 2005-2023)

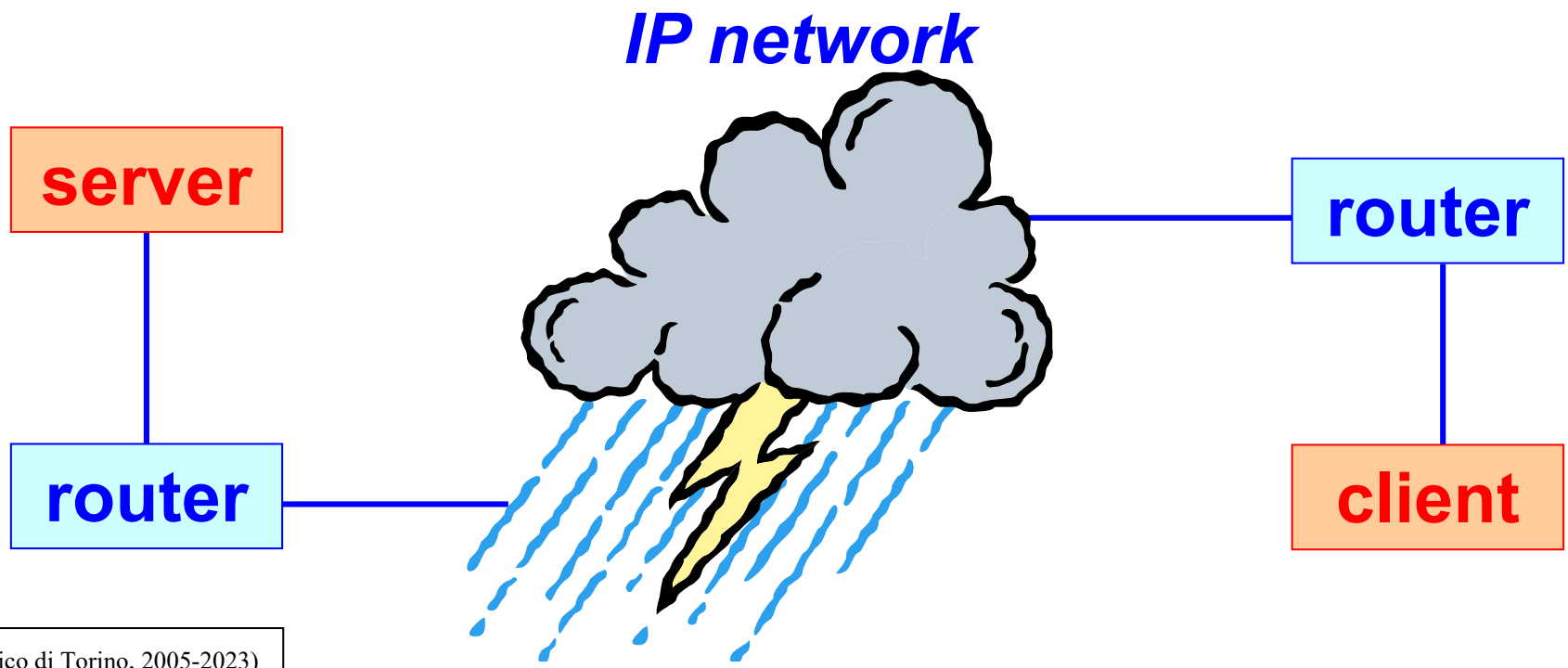
DHCP protection

- **some switches (e.g. Cisco) offer:**
  - DHCPsnooping = only transmit replies from “trusted ports”
  - IP guard = switching only IPs got from a DHCP server (but there is a limit to the number of recognized addresses)
- **RFC-3118 “Authentication for DHCP messages”**
  - use of HMAC-MD5 to authenticate the messages
  - problem = key distribution and management (shared key!)
  - rarely adopted

© A.Lioy (Politecnico di Torino, 2005-2023)

Security at network level (L3)

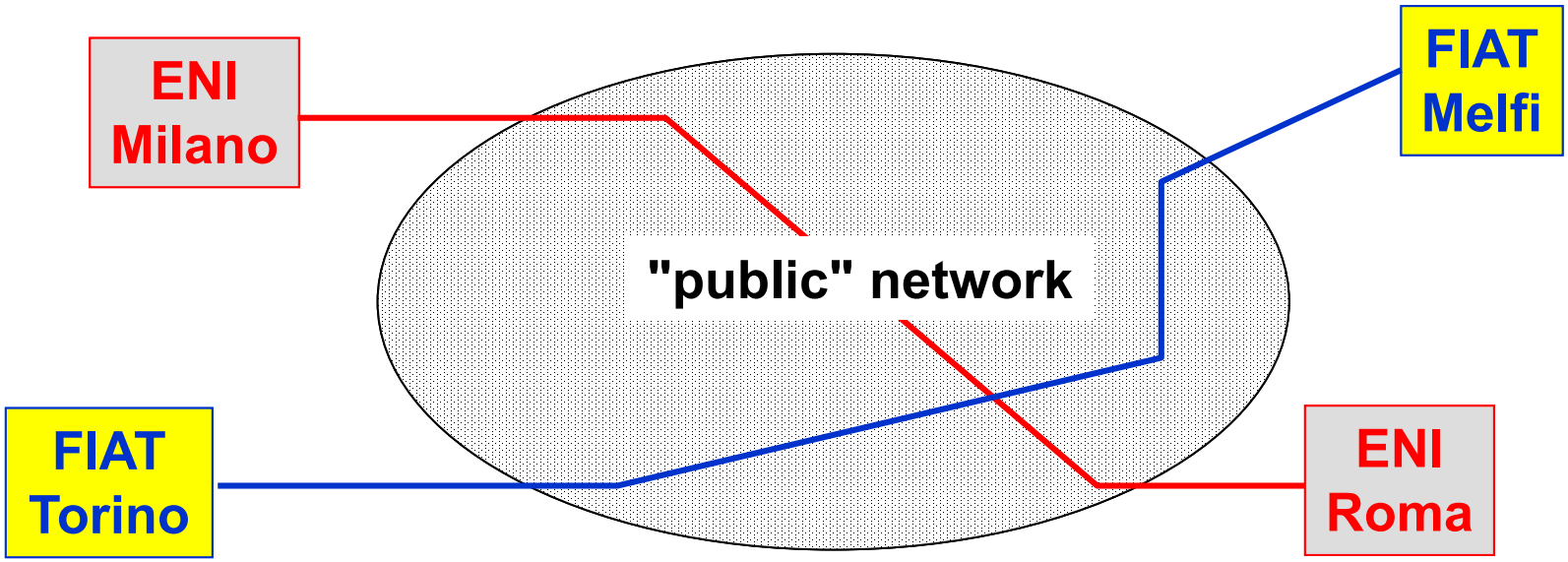
- end-to-end protection for L3-homogeneous networks (e.g. IP networks)
- creation of VPN (Virtual Private Network)



© A.Lioy (Politecnico di Torino, 2005-2023)

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



© A.Lioy (Politecnico di Torino, 2005-2023)

Techniques to create a VPN

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

© A.Lioy (Politecnico di Torino, 2005-2023)

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

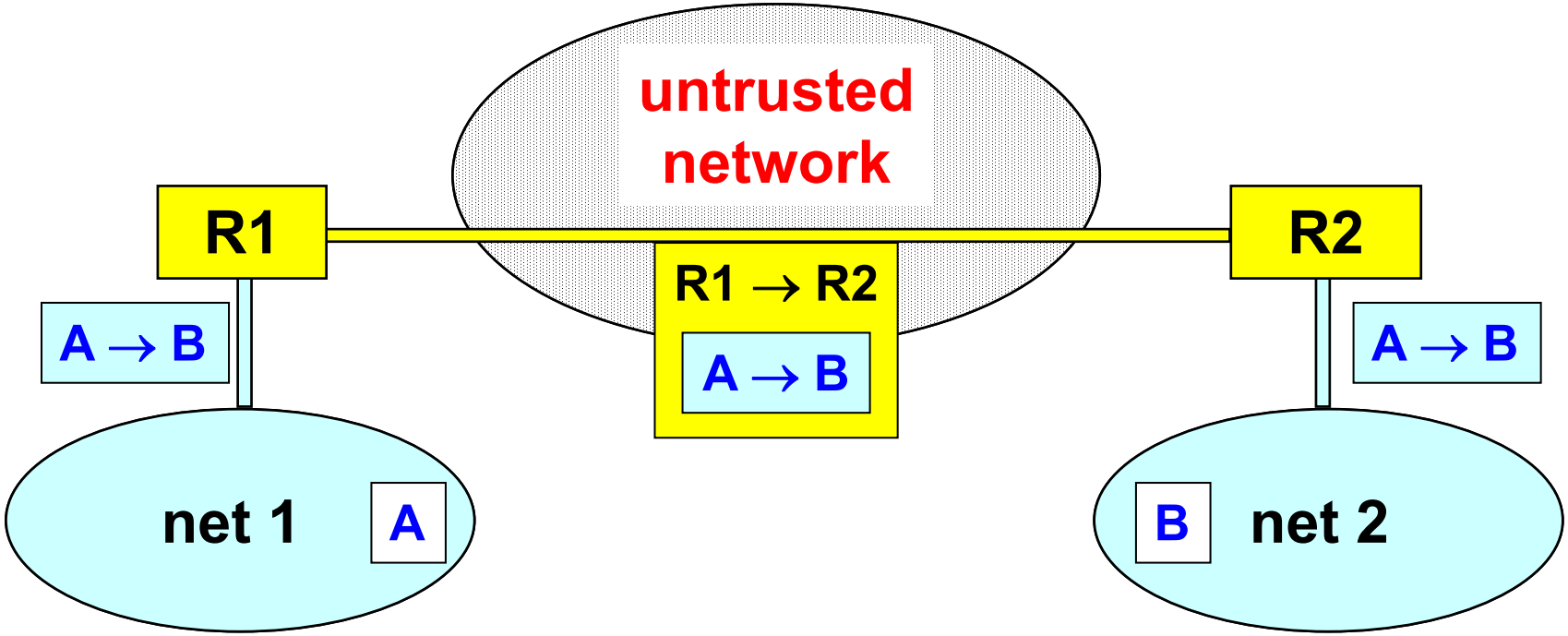
© A.Lioy (Politecnico di Torino, 2005-2023)

2. VPN via tunnel

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

© A.Lioy (Politecnico di Torino, 2005-2023)

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

© A.Lioy (Politecnico di Torino, 2005-2023)

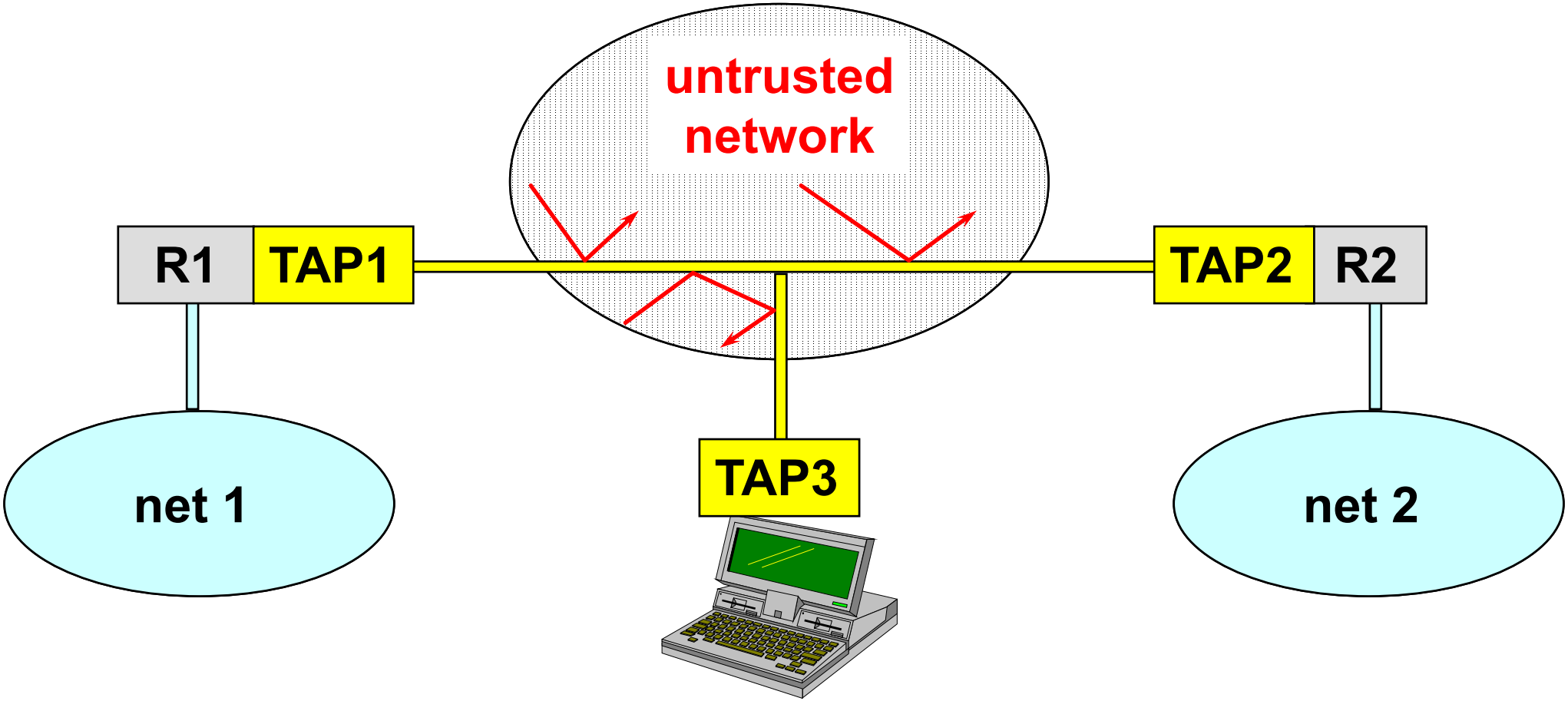
3. VPN via secure IP tunnel

- **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)**

© A.Lioy (Politecnico di Torino, 2005-2023)

VPN via secure IP tunnel



© A.Lioy (Politecnico di Torino, 2005-2023)

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)

© A.Lioy (Politecnico di Torino, 2005-2023)

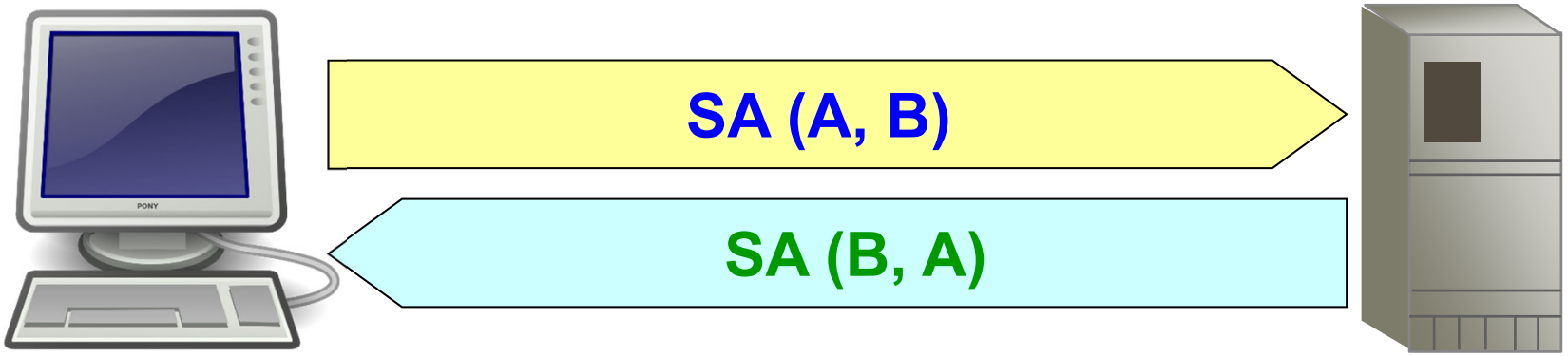
IPsec security services

- **authentication of IP packets:**
  - computation of a keyed-digest with a shared key
  - provides:
    - data integrity and sender authentication
    - (partial) protection against “replay” attacks as the packet contains a sequence number
- **confidentiality of IP packets:**
  - payload encryption with a symmetric algo and a shared key
  - provides data privacy
- **peer authentication when creating the SA:**
  - key agreement after authN (shared key or digital signature)

© A.Lioy (Politecnico di Torino, 2005-2023)

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



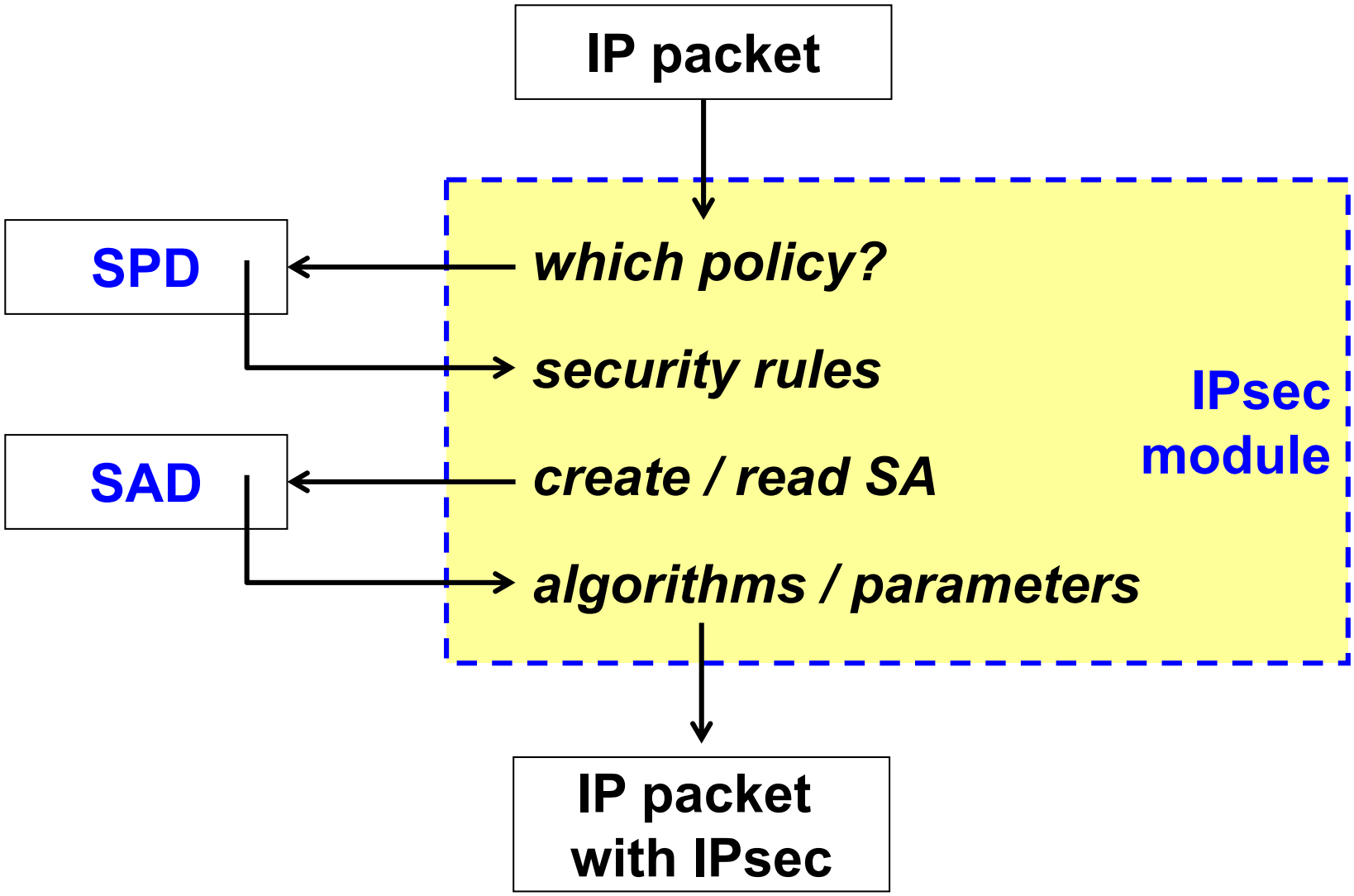
© A.Lioy (Politecnico di Torino, 2005-2023)

IPsec local database

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

© A.Lioy (Politecnico di Torino, 2005-2023)

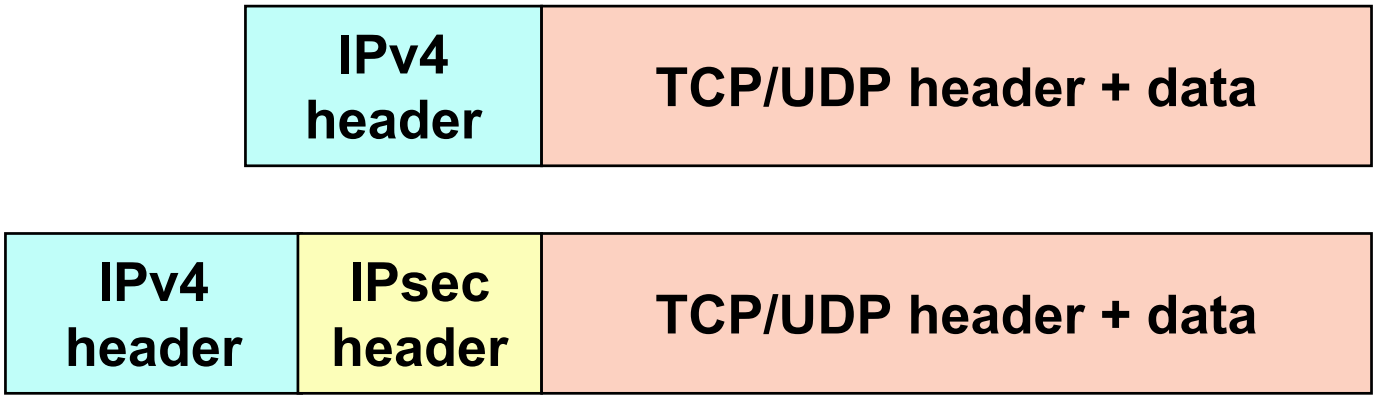
How IPsec works (sending)



© A.Lioy (Politecnico di Torino, 2005-2023)

Transport mode IPsec

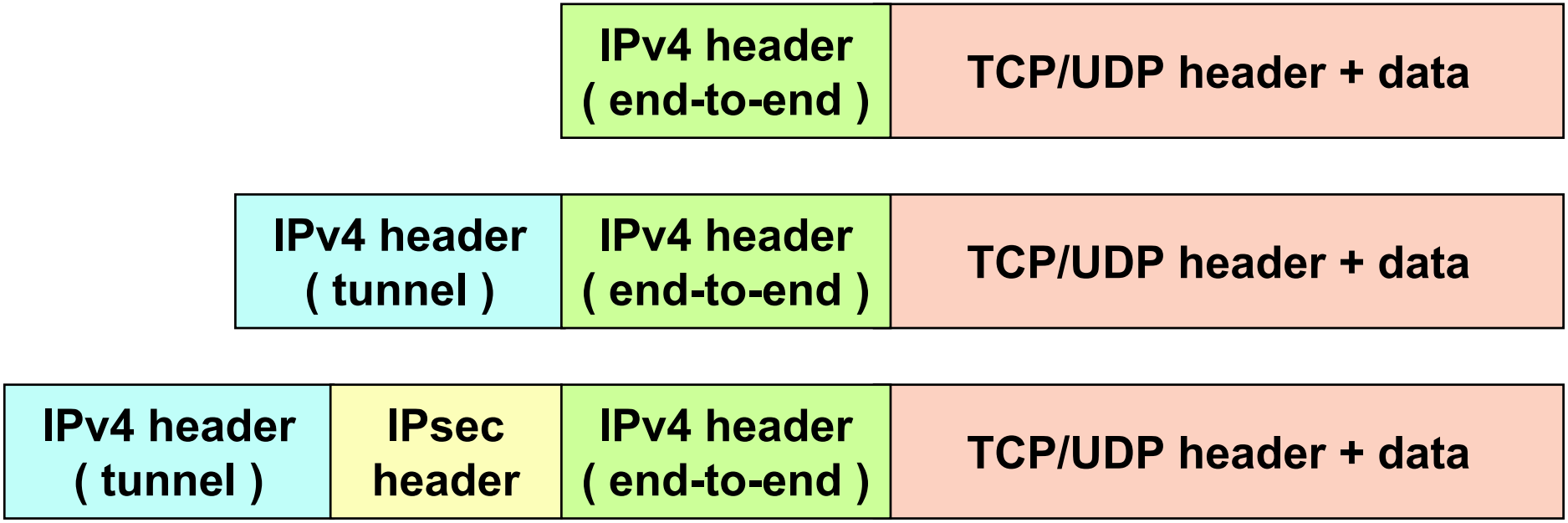
- used for end-to-end security, that is used by hosts, not gateways (exception: traffic for the gateway itself, e.g. SNMP, ICMP)
- pro: computationally light
- con: no protection of header variable fields



© A.Lioy (Politecnico di Torino, 2005-2023)

Tunnel mode IPsec

- used to create a VPN, usually by gateways
- pro: protection of E2E header variable fields
- con: computationally heavy



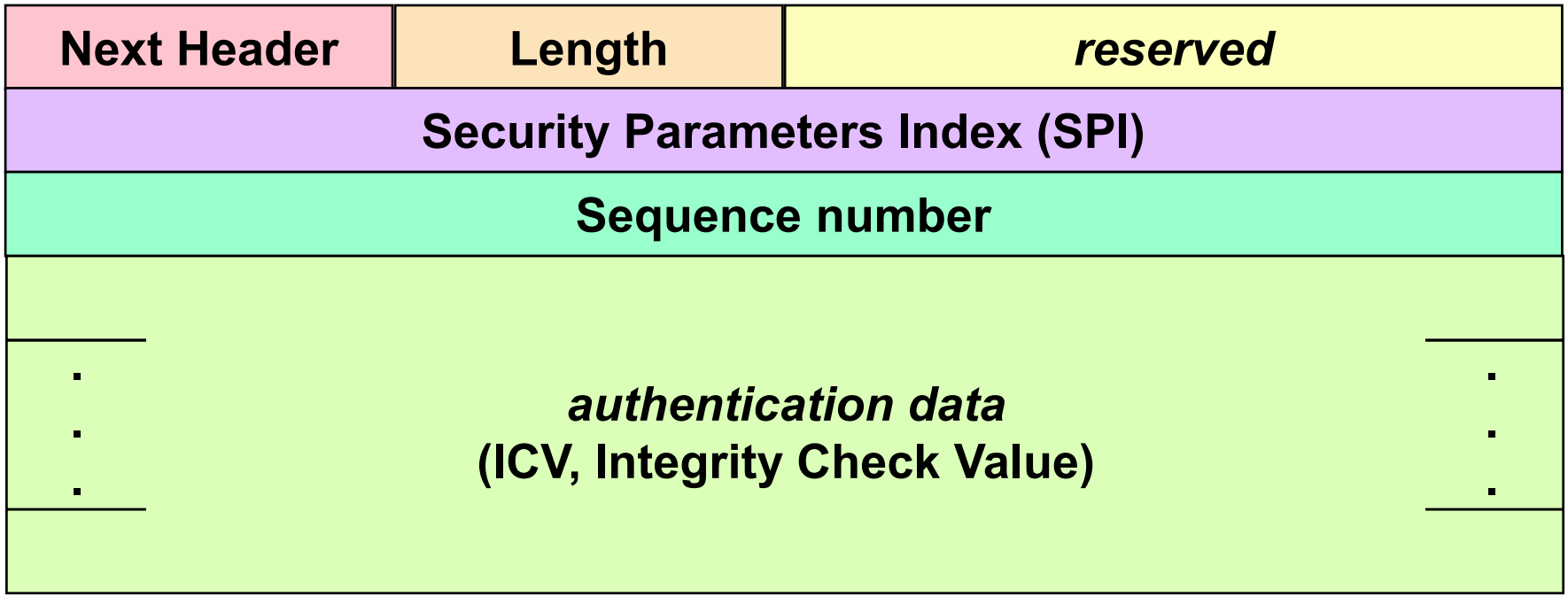
© A.Lioy (Politecnico di Torino, 2005-2023)

AH

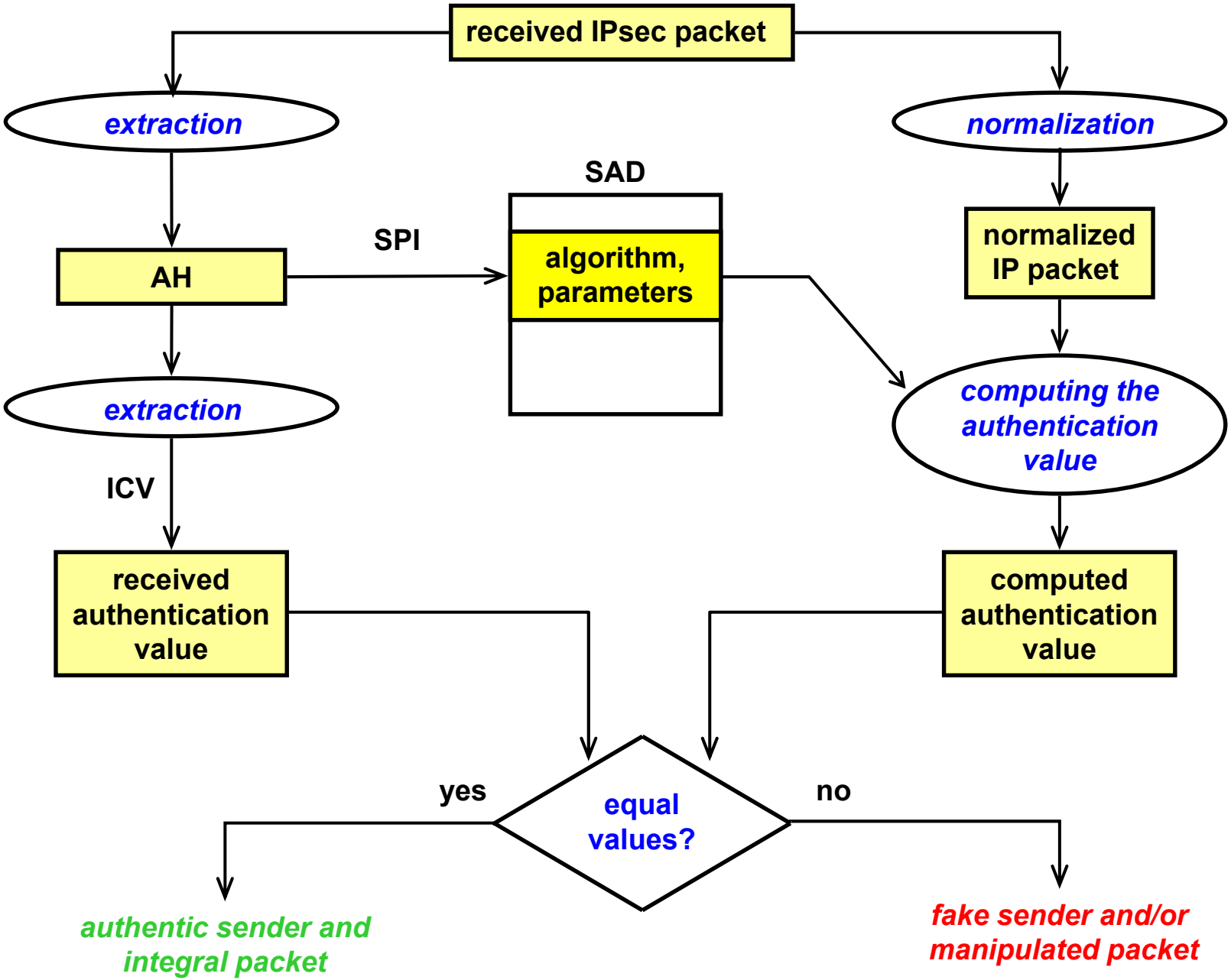
- **Authentication Header**
- **mechanism (first version, RFC-1826):**
  - data integrity and sender authentication
  - compulsory support of keyed-MD5 (RFC-1828)
  - optional support of keyed-SHA-1 (RFC-1852)
- **mechanism (second version, RFC-2402):**
  - data integrity, sender authentication and (partial)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

© A.Lioy (Politecnico di Torino, 2005-2023)

AH - format (RFC-4302)



© A.Lioy (Politecnico di Torino, 2005-2023)



© A.Lioy (Politecnico di Torino, 2005-2023)

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

© A.Lioy (Politecnico di Torino, 2005-2023)

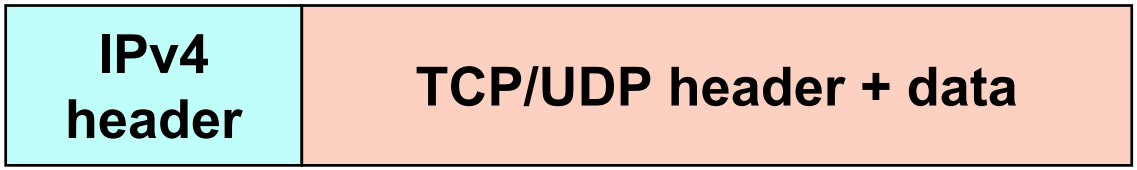
ESP

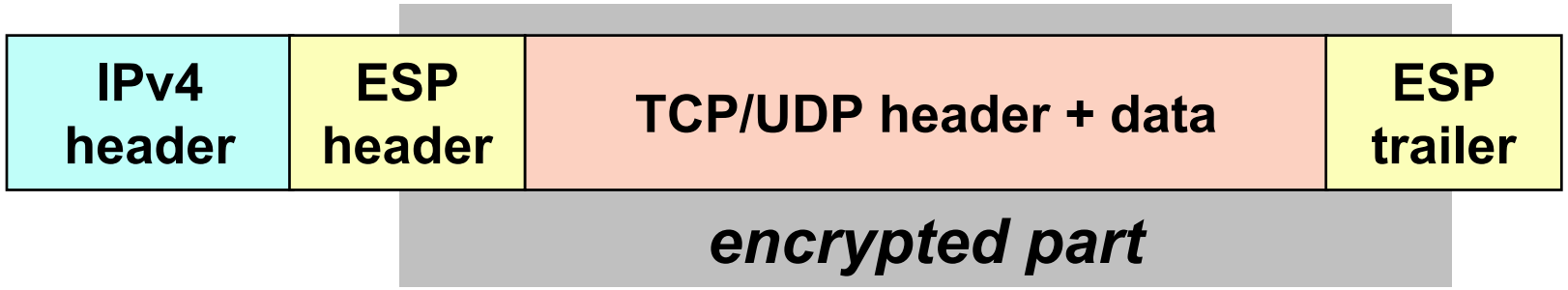
- **Encapsulating Security Payload**
- **first version (RFC-1827) gave only confidentiality**
- **base mechanism: DES-CBC (RFC-1829)**
- **other mechanisms possible**
- **second version (RFC-2406):**
  - provides also authentication (but the IP header, so the coverage is not equivalent to that of AH)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

© A.Lioy (Politecnico di Torino, 2005-2023)

ESP in transport mode

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

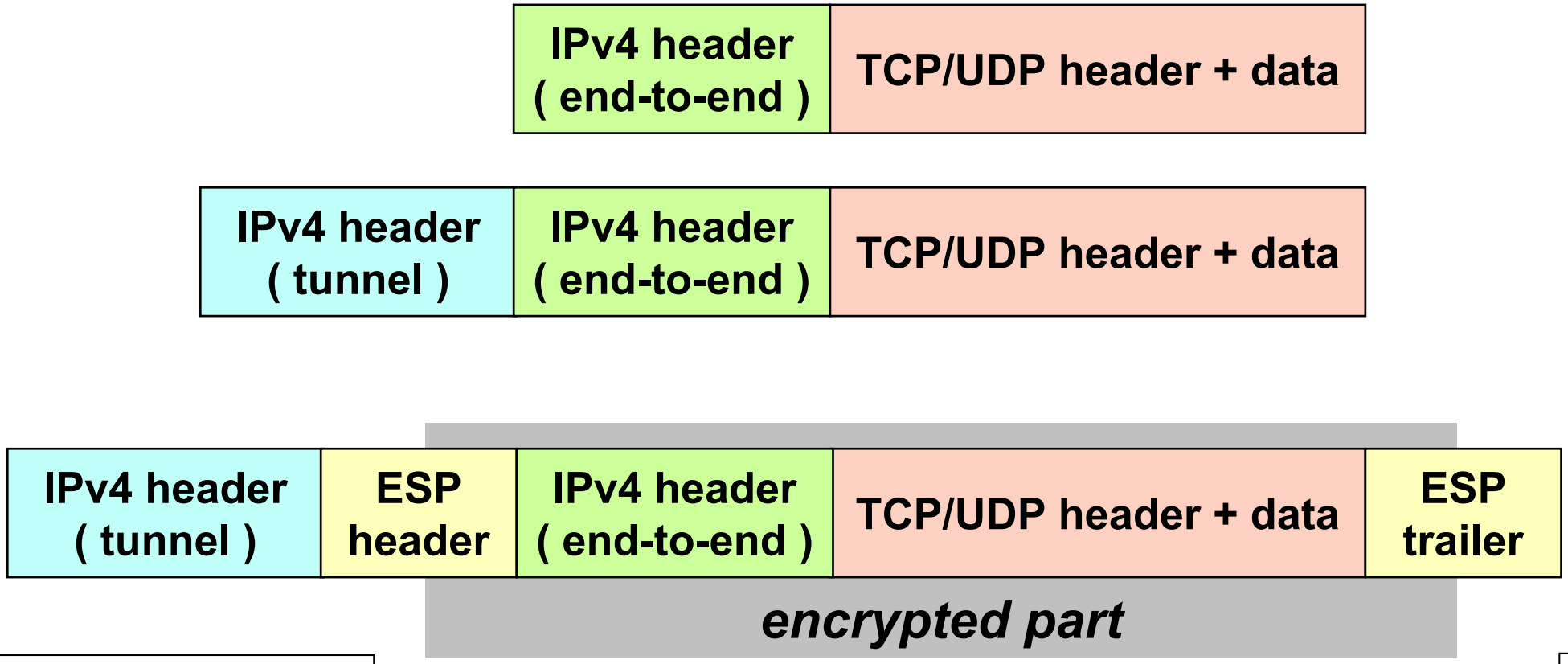




© A.Lioy (Politecnico di Torino, 2005-2023)

ESP in tunnel mode

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



© A.Lioy (Politecnico di Torino, 2005-2023)

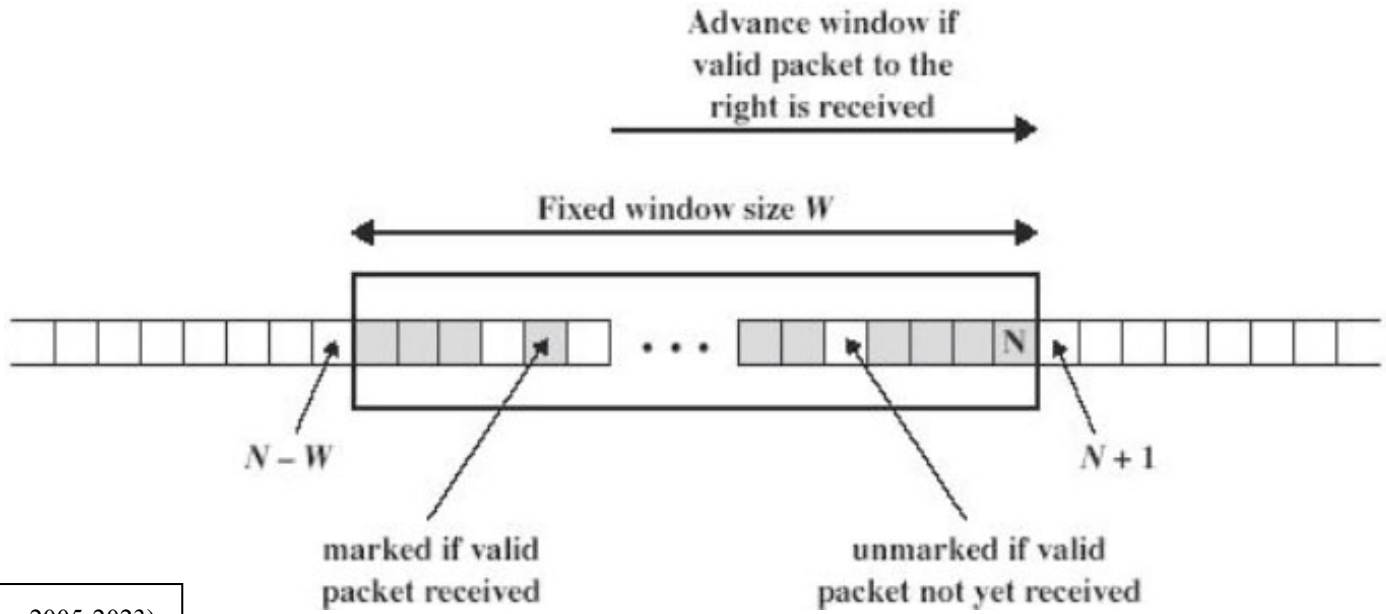
IPsec implementation details

- **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)

© A.Lioy (Politecnico di Torino, 2005-2023)

IPsec replay protection

- at SA creation, 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
- moving window: outside it, no replay protection



© A.Lioy (Politecnico di Torino, 2005-2023)

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

© A.Lioy (Politecnico di Torino, 2005-2023)

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

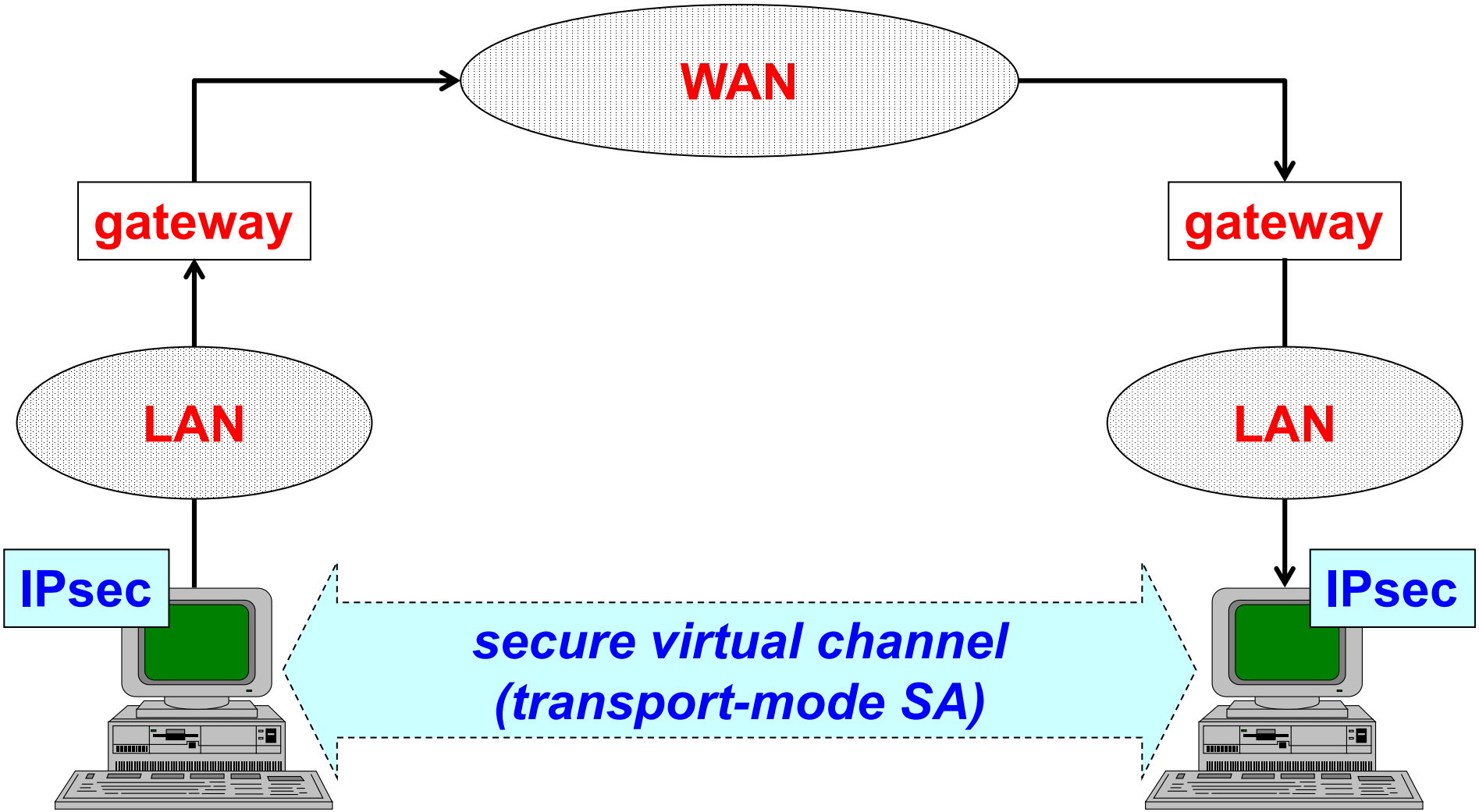
© A.Lioy (Politecnico di Torino, 2005-2023)

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 …

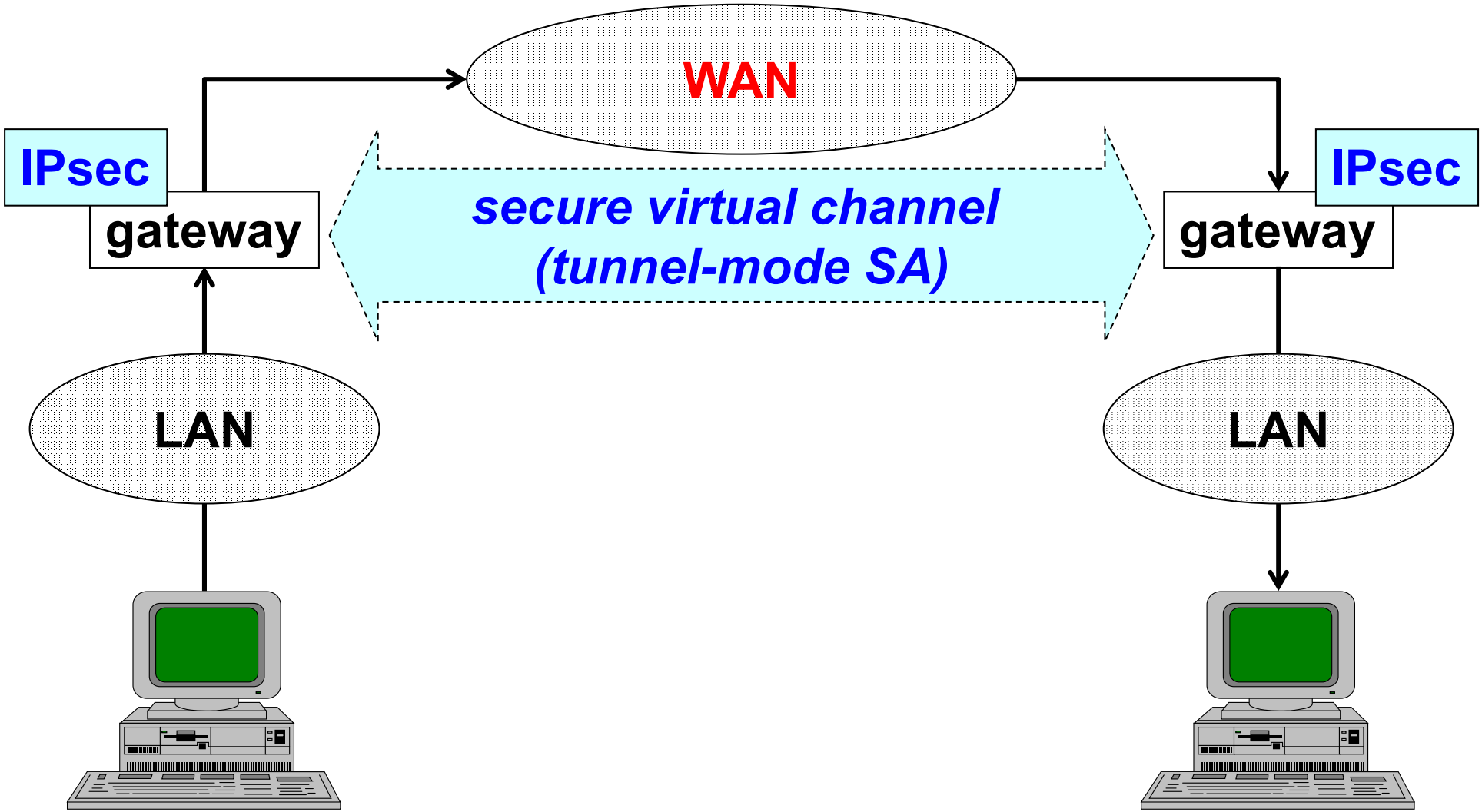
© A.Lioy (Politecnico di Torino, 2005-2023)

End-to-end security



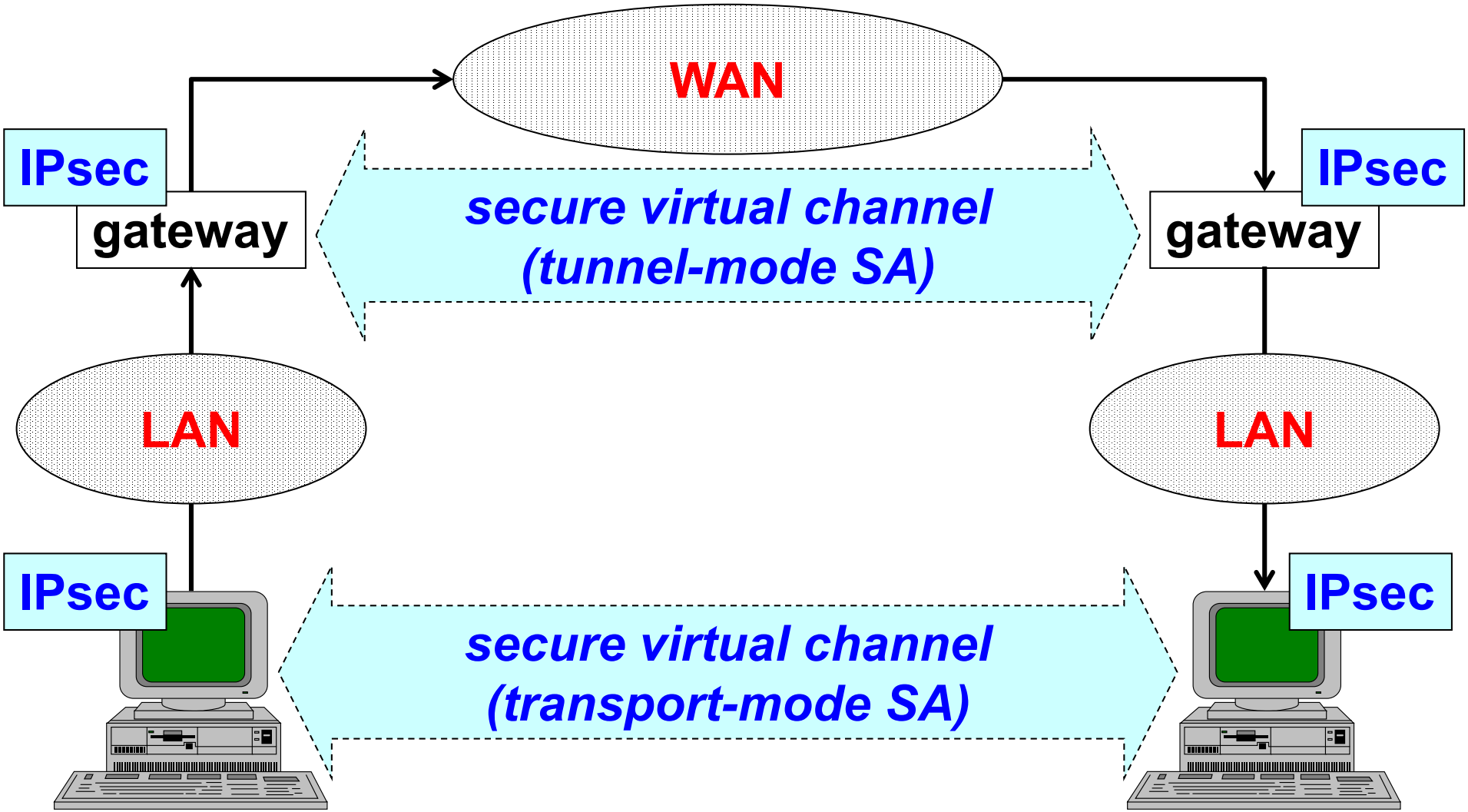
© A.Lioy (Politecnico di Torino, 2005-2023)

Basic VPN



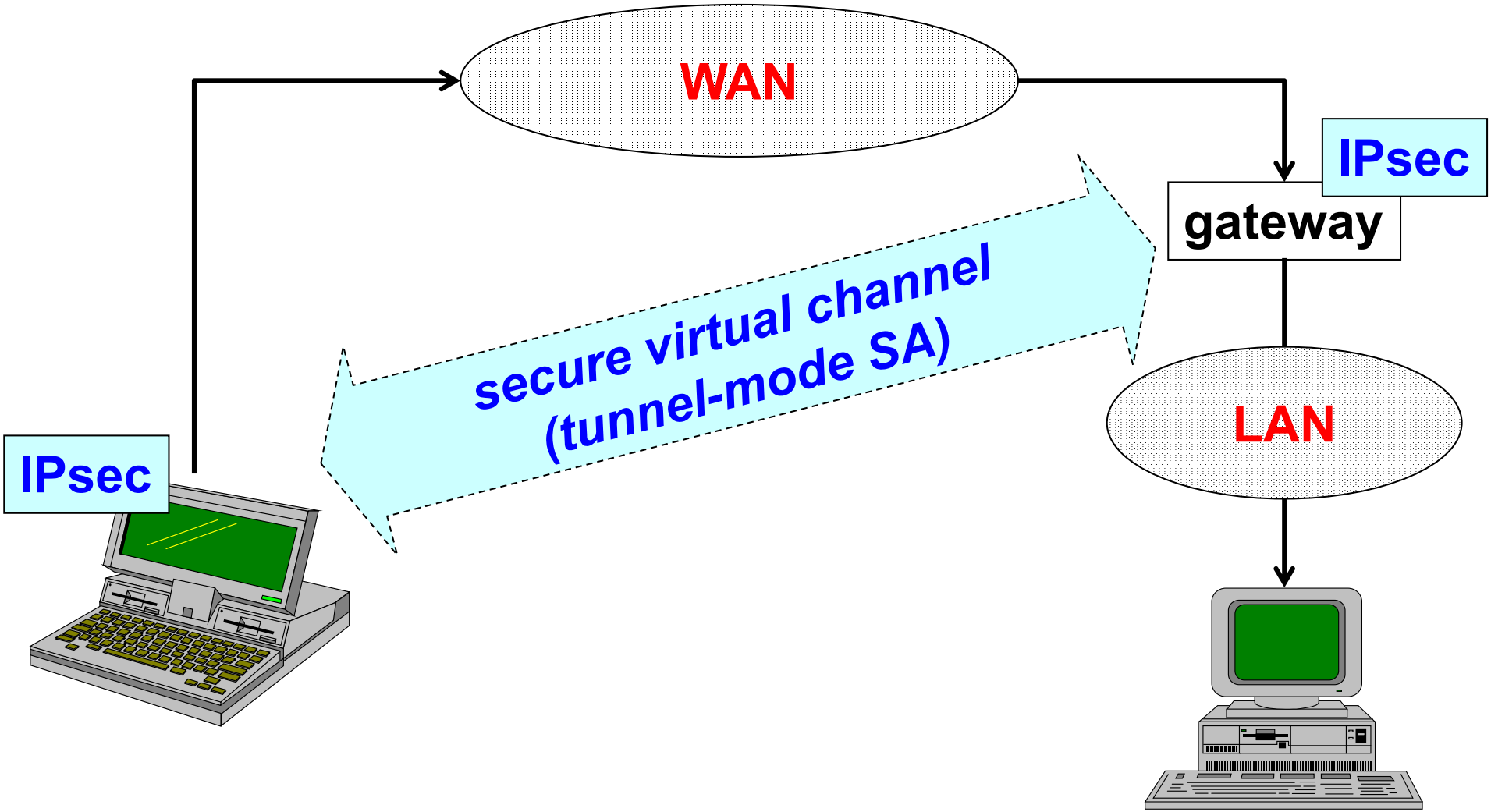
© A.Lioy (Politecnico di Torino, 2005-2023)

End-to-end security with basic VPN



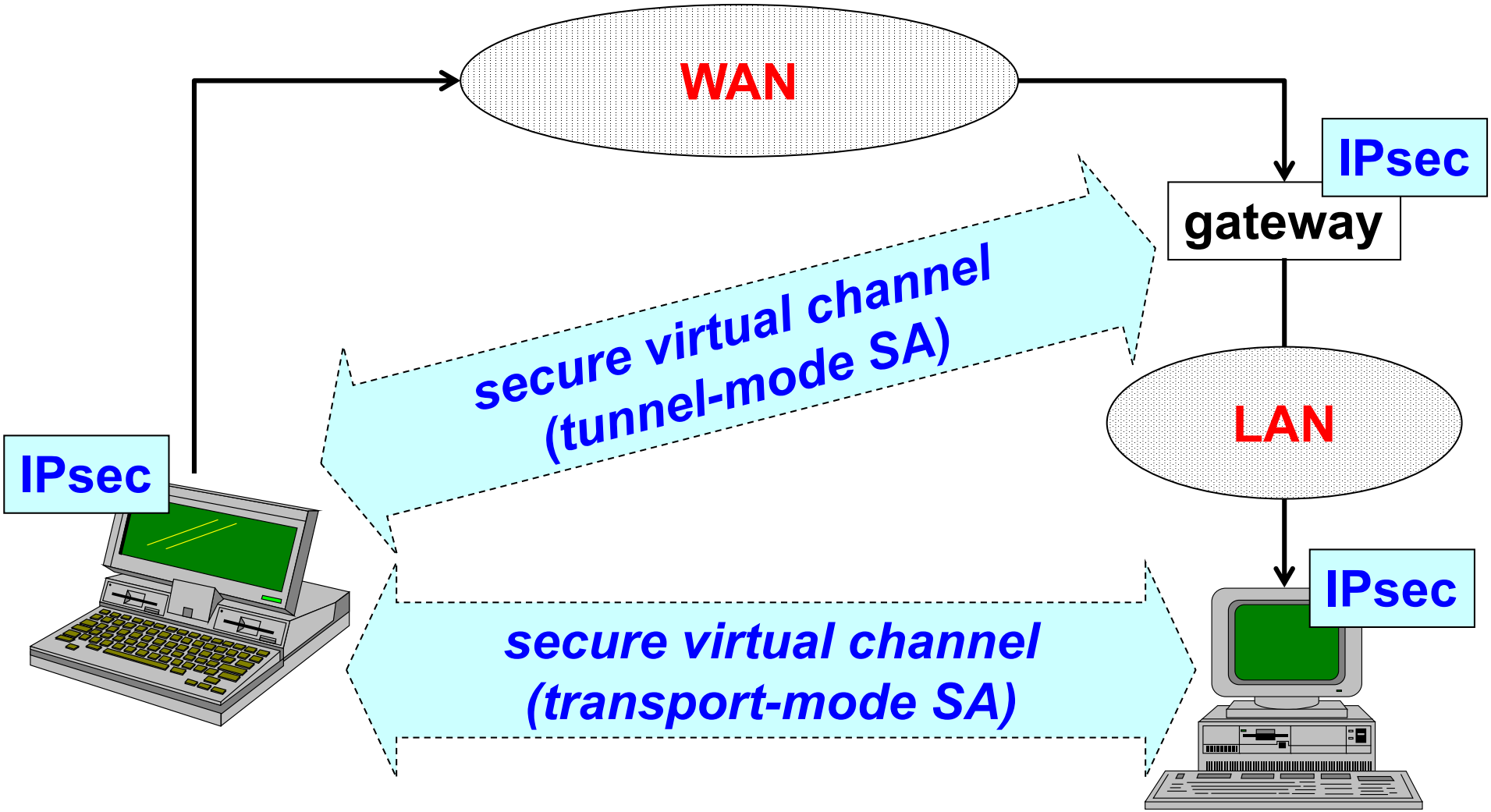
© A.Lioy (Politecnico di Torino, 2005-2023)

Secure gateway



© A.Lioy (Politecnico di Torino, 2005-2023)

Secure remote access



© A.Lioy (Politecnico di Torino, 2005-2023)

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?)

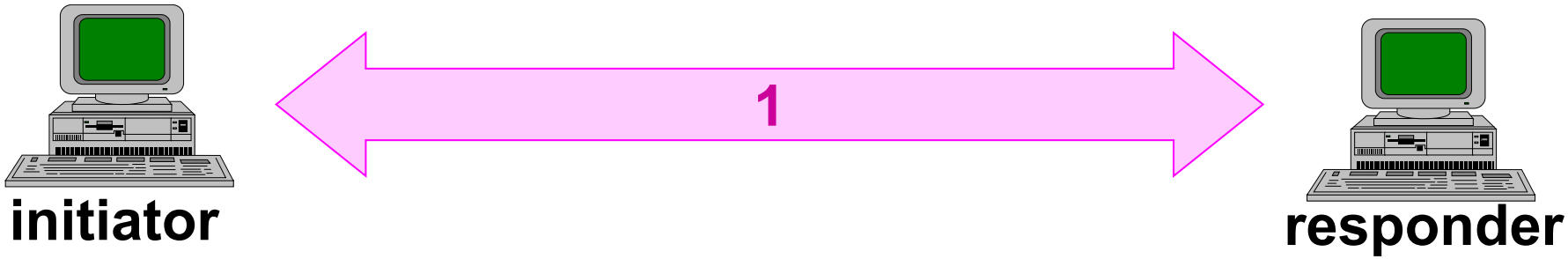
© A.Lioy (Politecnico di Torino, 2005-2023)

ISAKMP, OAKLEY, and IKE

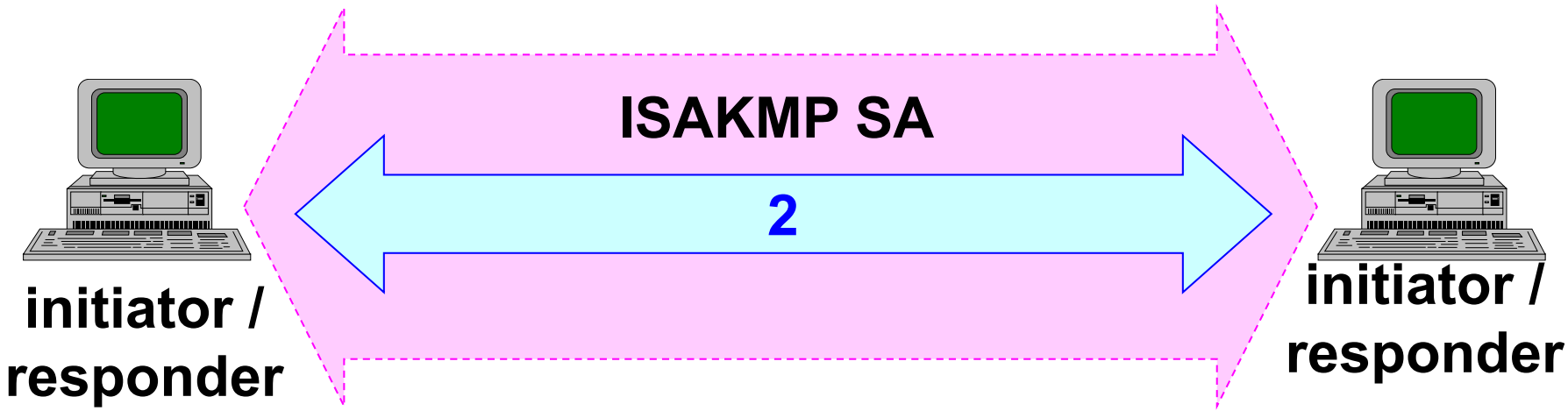
- **ISAKMP, Internet Security Association and Key Management Protocol (RFC-2408)**
  - procedures to negotiate, set-up, modify and delete a SA
  - key exchange method not fixed:
    - OAKLEY (RFC-2412): protocol for authenticated exchange of symmetric keys
- **IKE, Internet Key Exchange (RFC-2409) = ISAKMP + OAKLEY**
  - creation of one 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**

© A.Lioy (Politecnico di Torino, 2005-2023)

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"

© A.Lioy (Politecnico di Torino, 2005-2023)

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

© A.Lioy (Politecnico di Torino, 2005-2023)

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)

© A.Lioy (Politecnico di Torino, 2005-2023)

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)**

© A.Lioy (Politecnico di Torino, 2005-2023)

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**

© A.Lioy (Politecnico di Torino, 2005-2023)

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, …)**

© A.Lioy (Politecnico di Torino, 2005-2023)

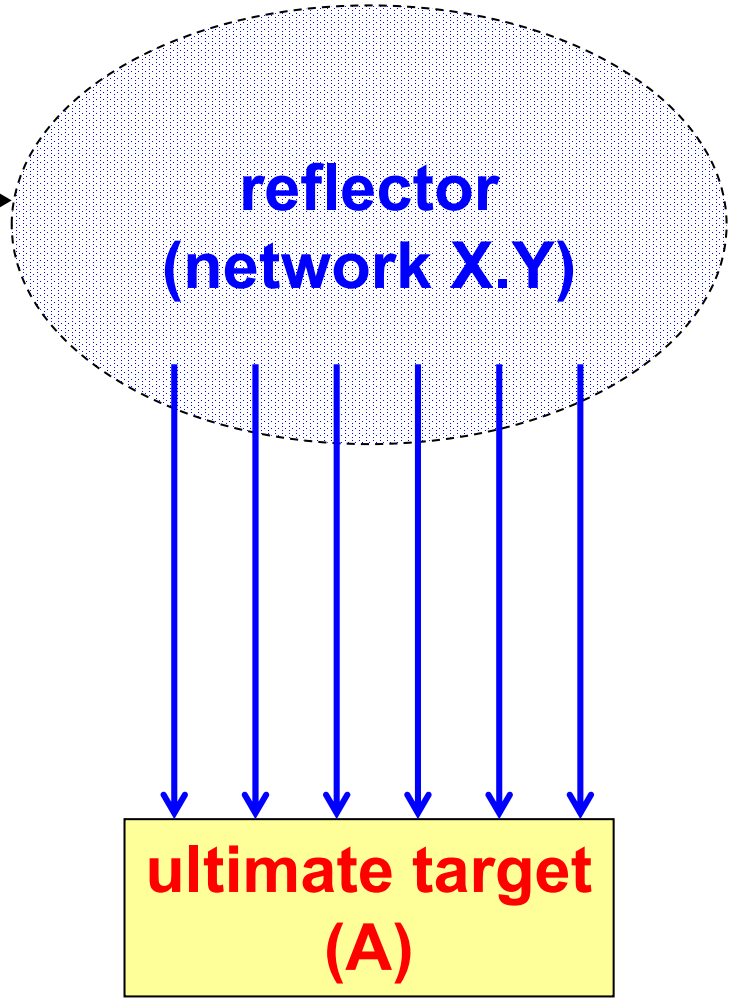
ICMP security

- **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 quench
  - redirect
  - time exceeded for a datagram

© A.Lioy (Politecnico di Torino, 2005-2023)

Smurfing attack

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



© A.Lioy (Politecnico di Torino, 2005-2023)

Anti-smurfing countermeasures

- **for external attacks:**
  - reject IP broadcast packets at your border
  - example (CISCO syntax)

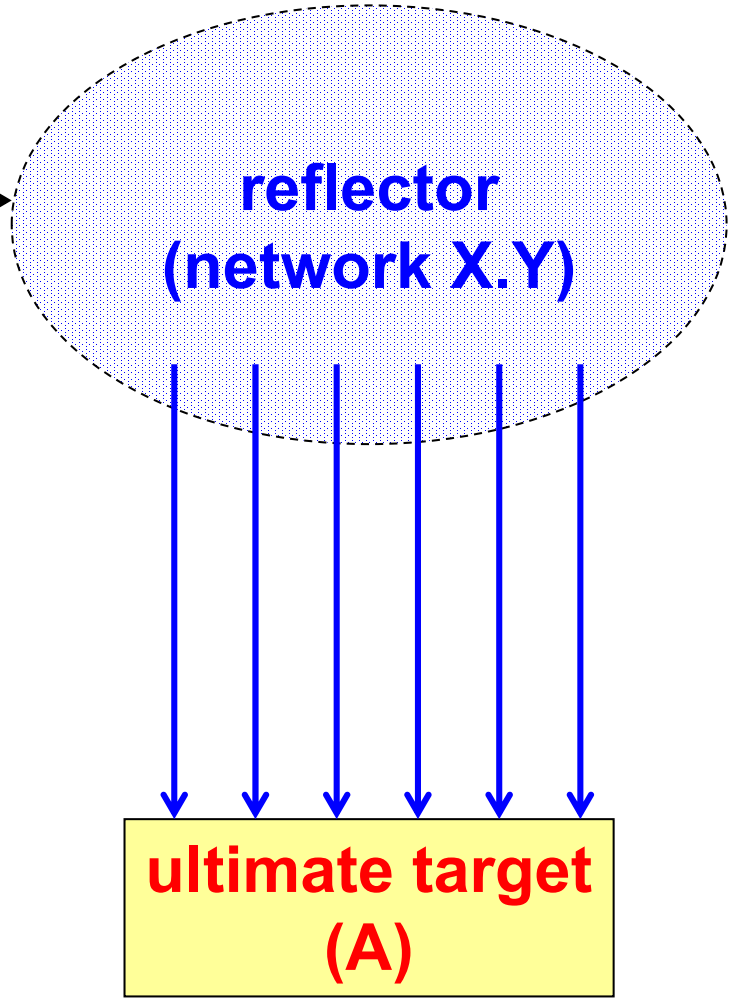
```
interface serial0
no ip directed-broadcast
```

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

© A.Lioy (Politecnico di Torino, 2005-2023)

Fraggle attack

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



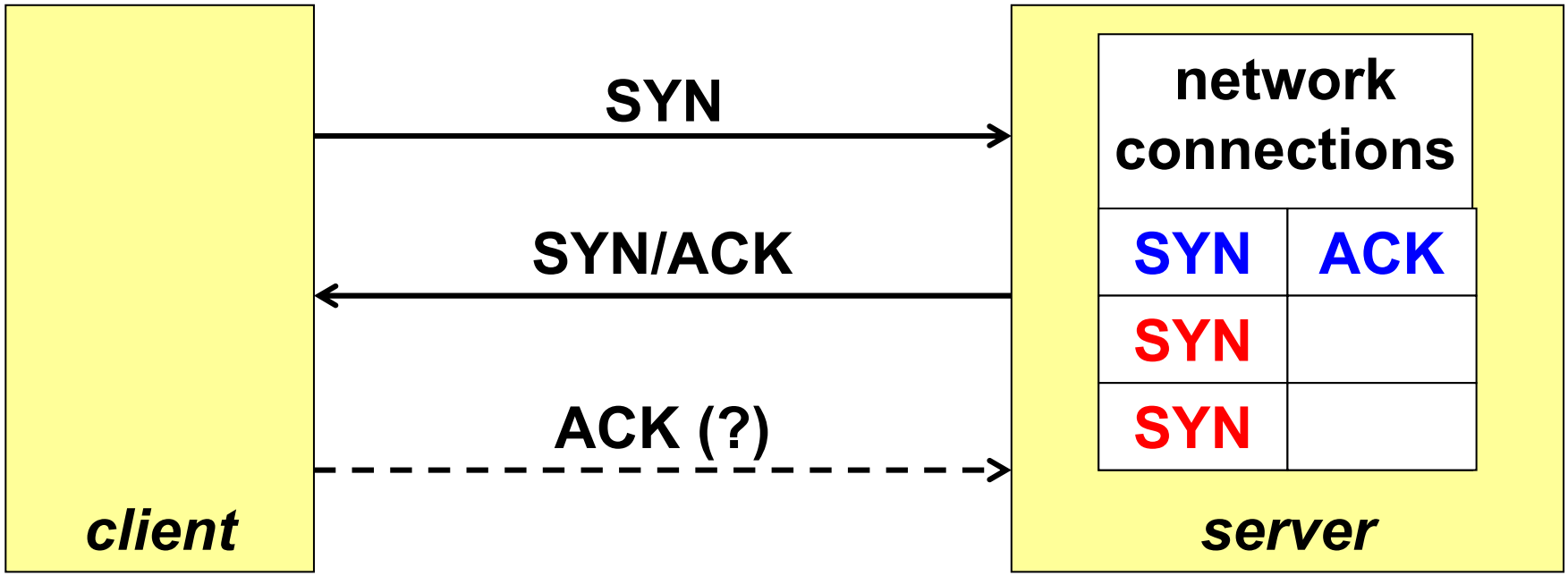
© A.Lioy (Politecnico di Torino, 2005-2023)

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)

© A.Lioy (Politecnico di Torino, 2005-2023)

TCP SYN flooding



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

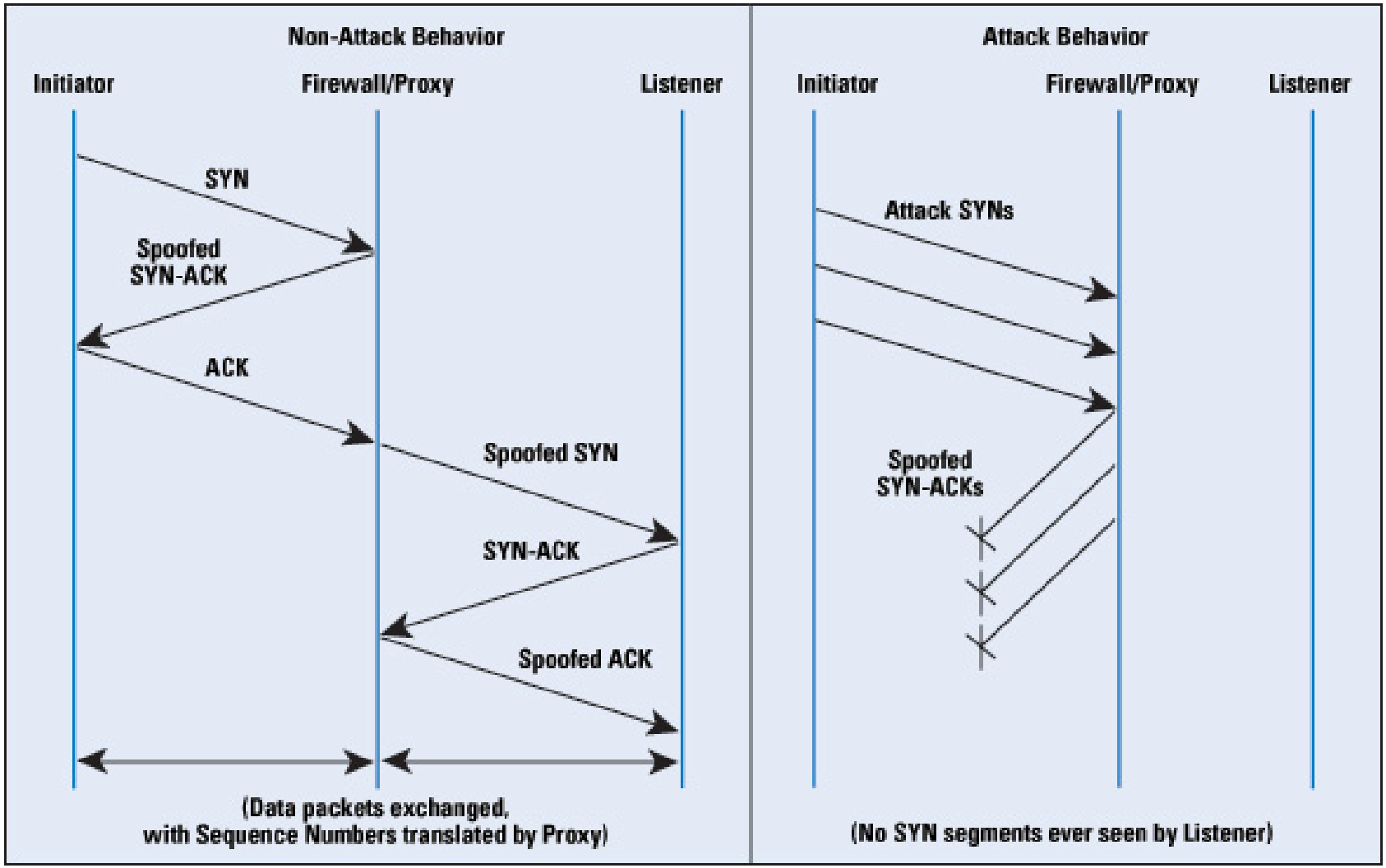
© A.Lioy (Politecnico di Torino, 2005-2023)

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)

© A.Lioy (Politecnico di Torino, 2005-2023)

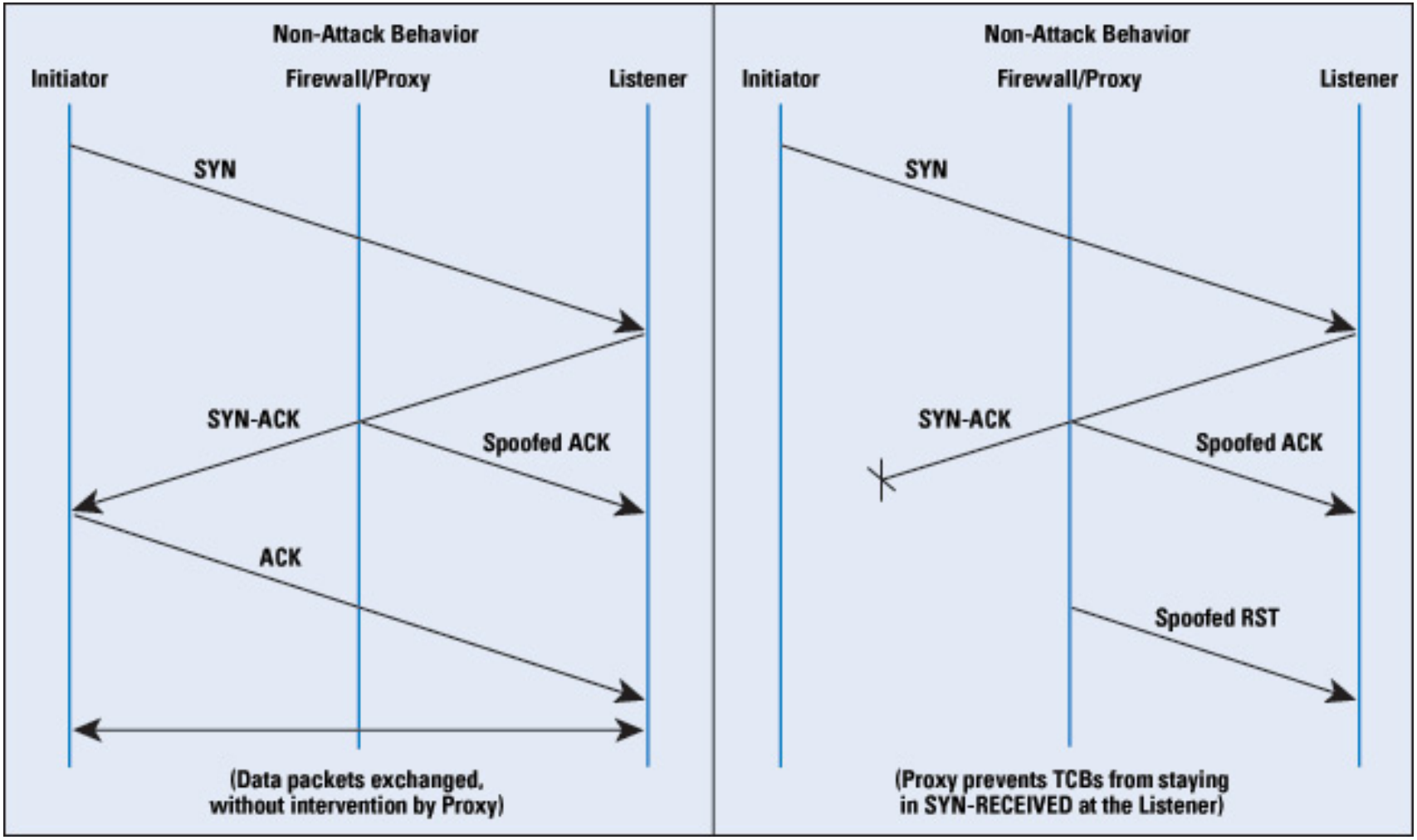
SYN interceptor (or firewall relay)



http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-4/syn_flooding_attacks.html

© A.Lioy (Politecnico di Torino, 2005-2023)

SYN monitor (or firewall gateway)



http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-4/syn_flooding_attacks.html

© A.Lioy (Politecnico di Torino, 2005-2023)

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

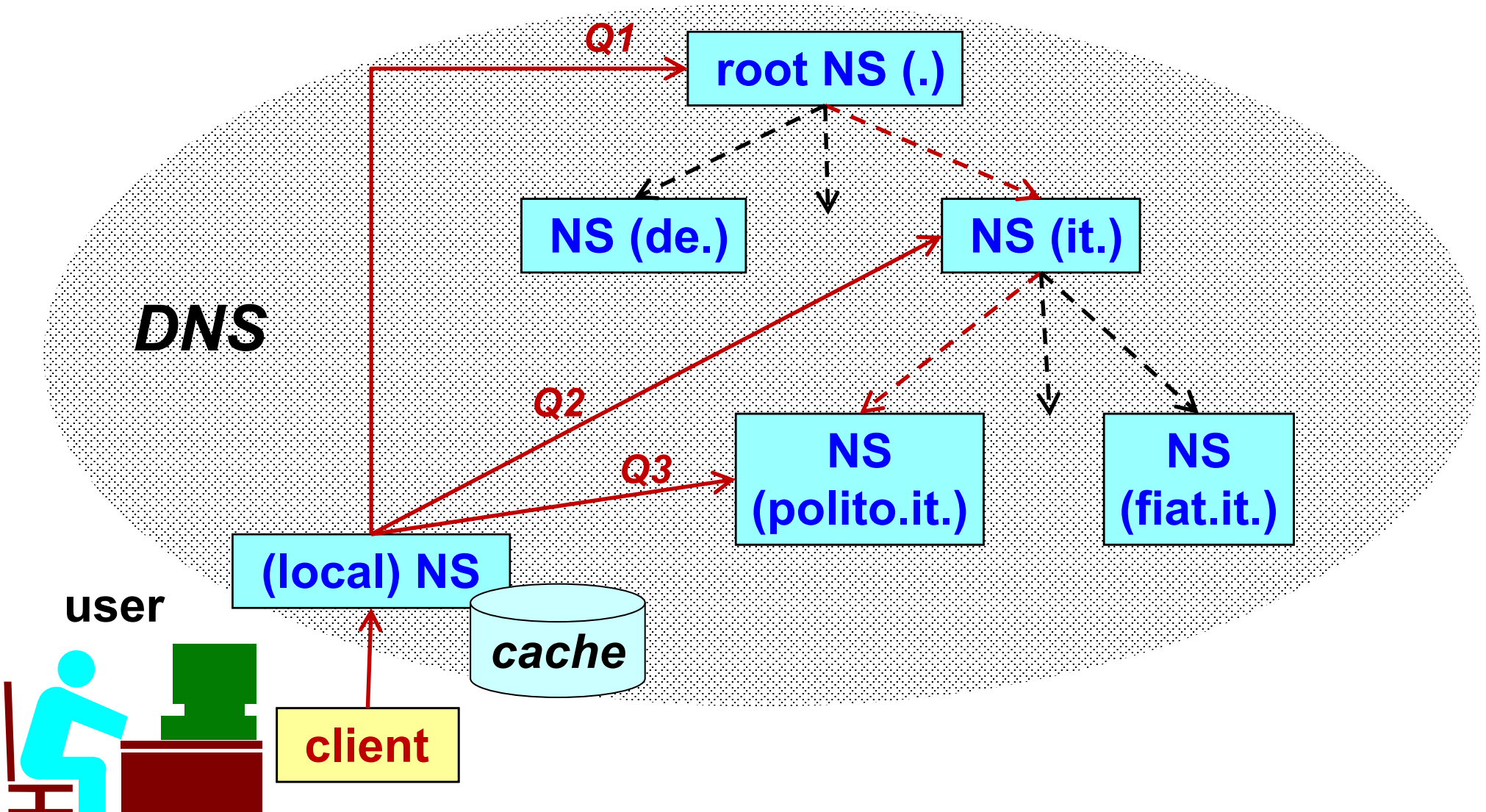
© A.Lioy (Politecnico di Torino, 2005-2023)

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 / deployment**

© A.Lioy (Politecnico di Torino, 2005-2023)

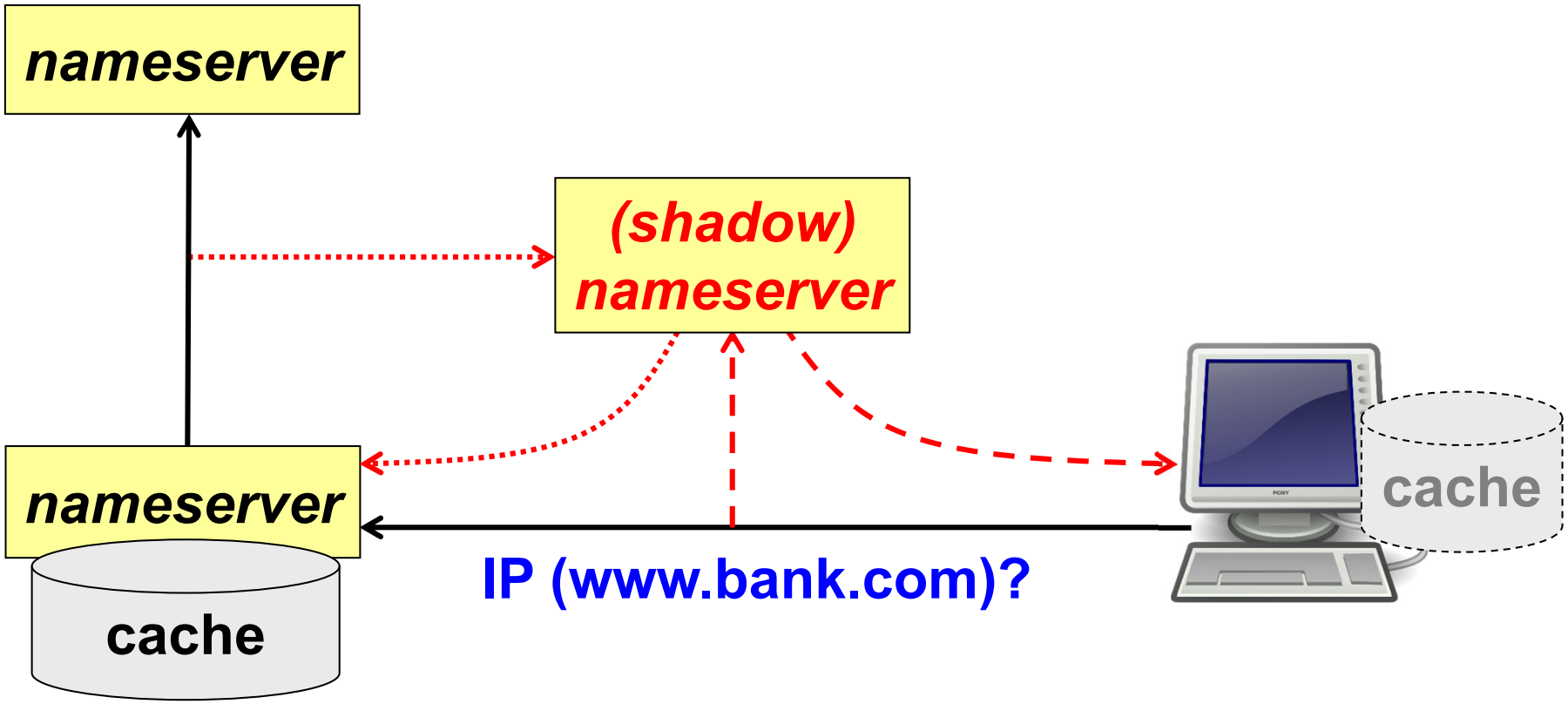
DNS architecture (iterative query)



© A.Lioy (Politecnico di Torino, 2005-2023)

DNS shadow server

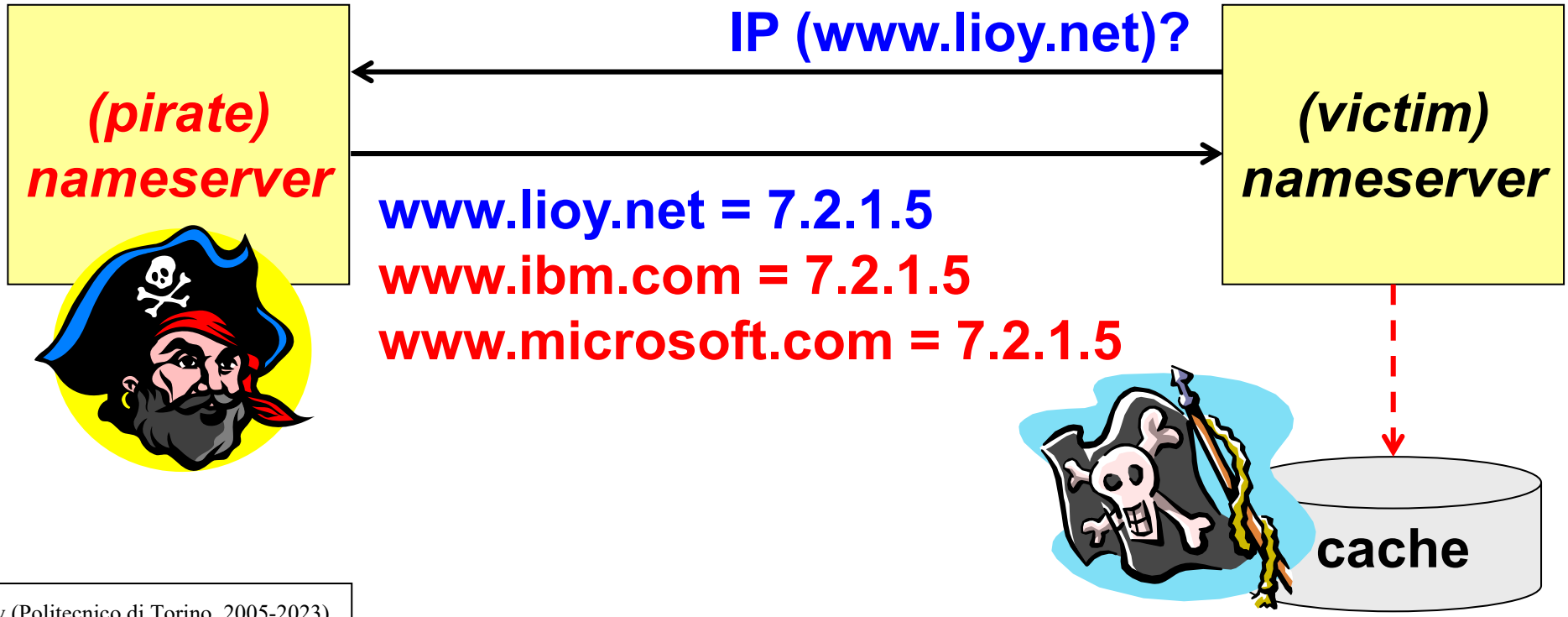
- sniffing to intercept the queries
- spoofing to generate fake answers (DoS or traffic redirection to fake sites)



© A.Lioy (Politecnico di Torino, 2005-2023)

DNS cache poisoning

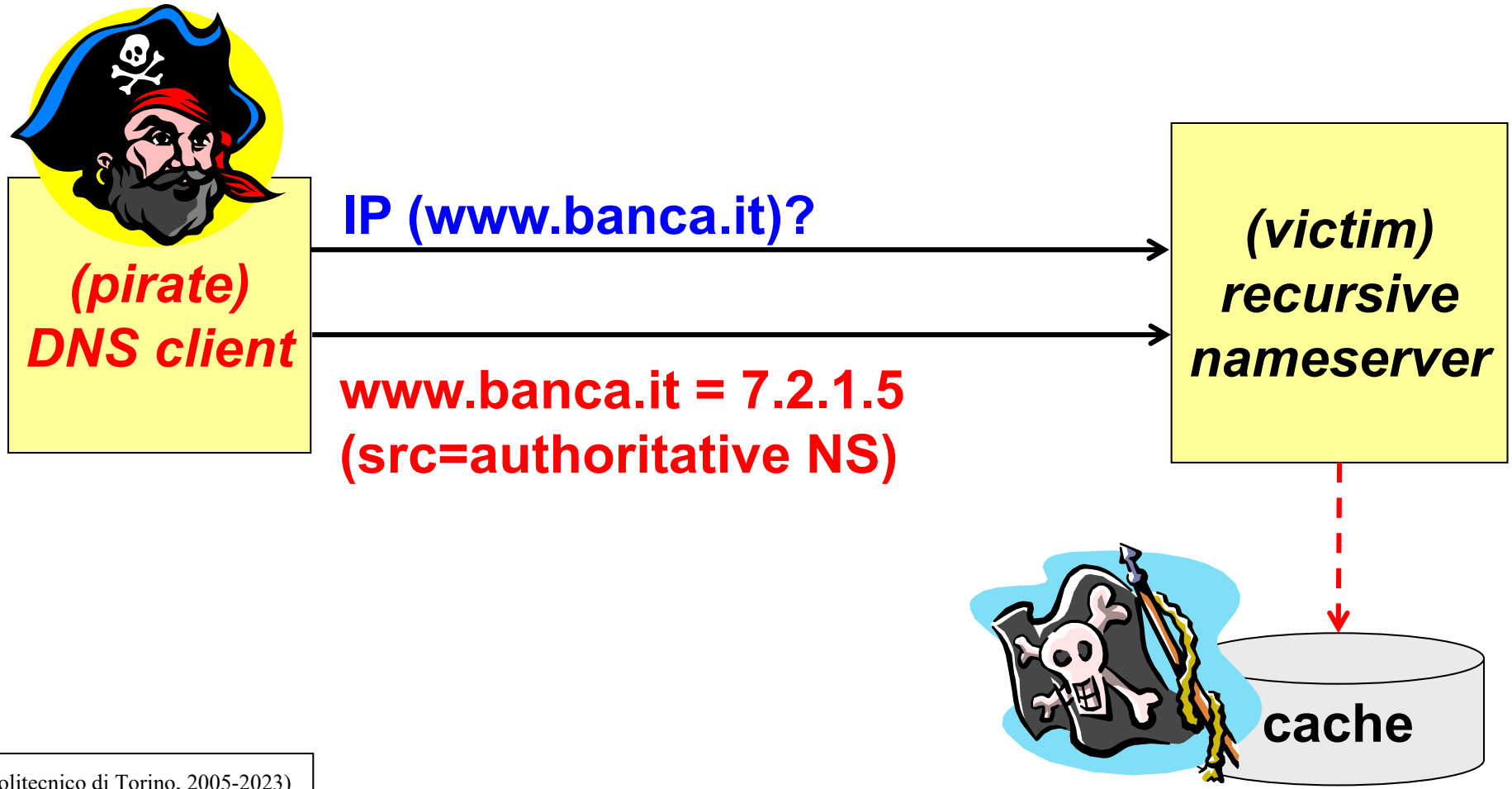
- attract the victim to make a query on my NS
- provide answers also to queries never done to push / overwrite the victim's cache



© A.Lioy (Politecnico di Torino, 2005-2023)

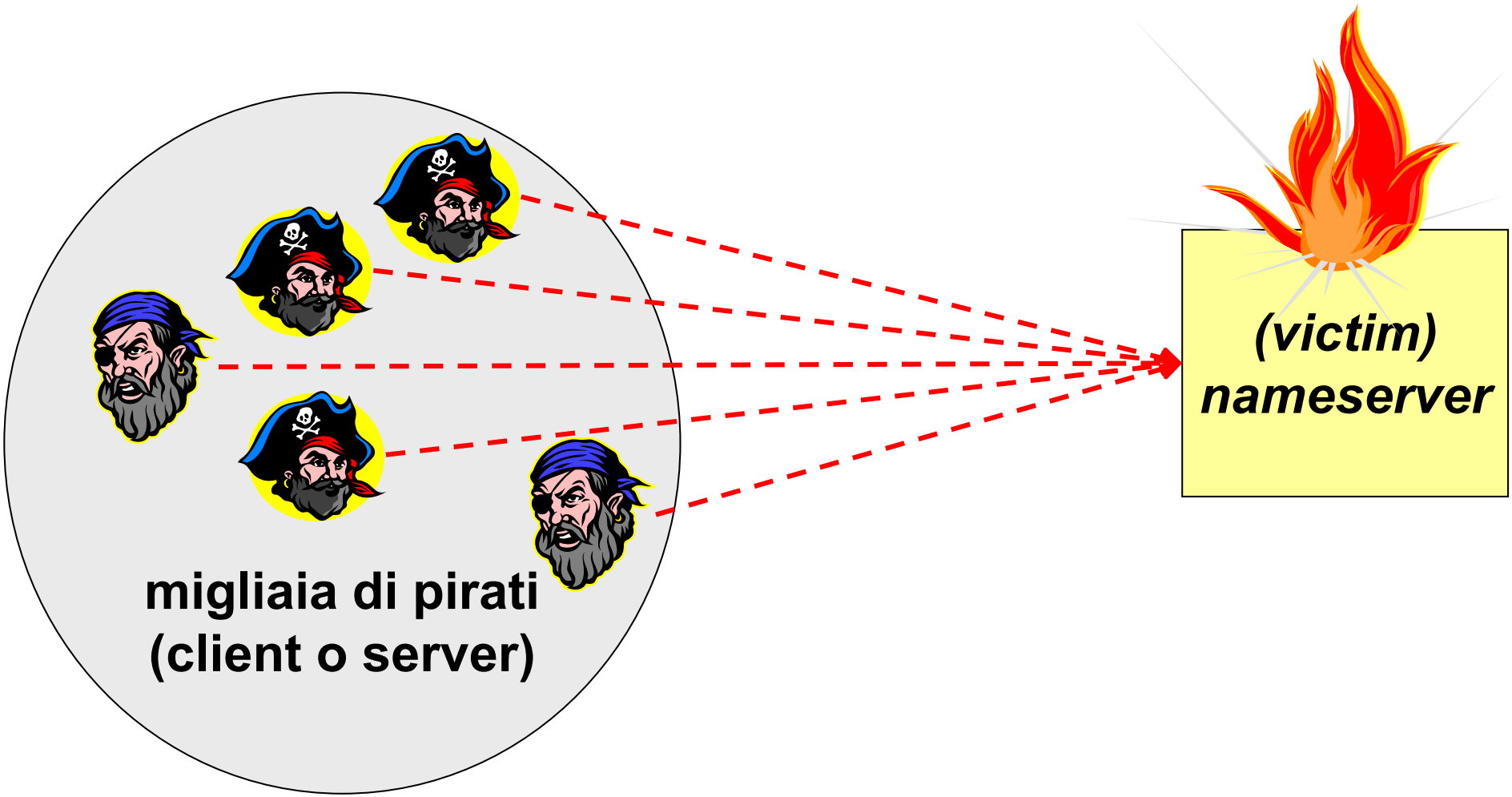
DNS cache poisoning (2nd version)

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



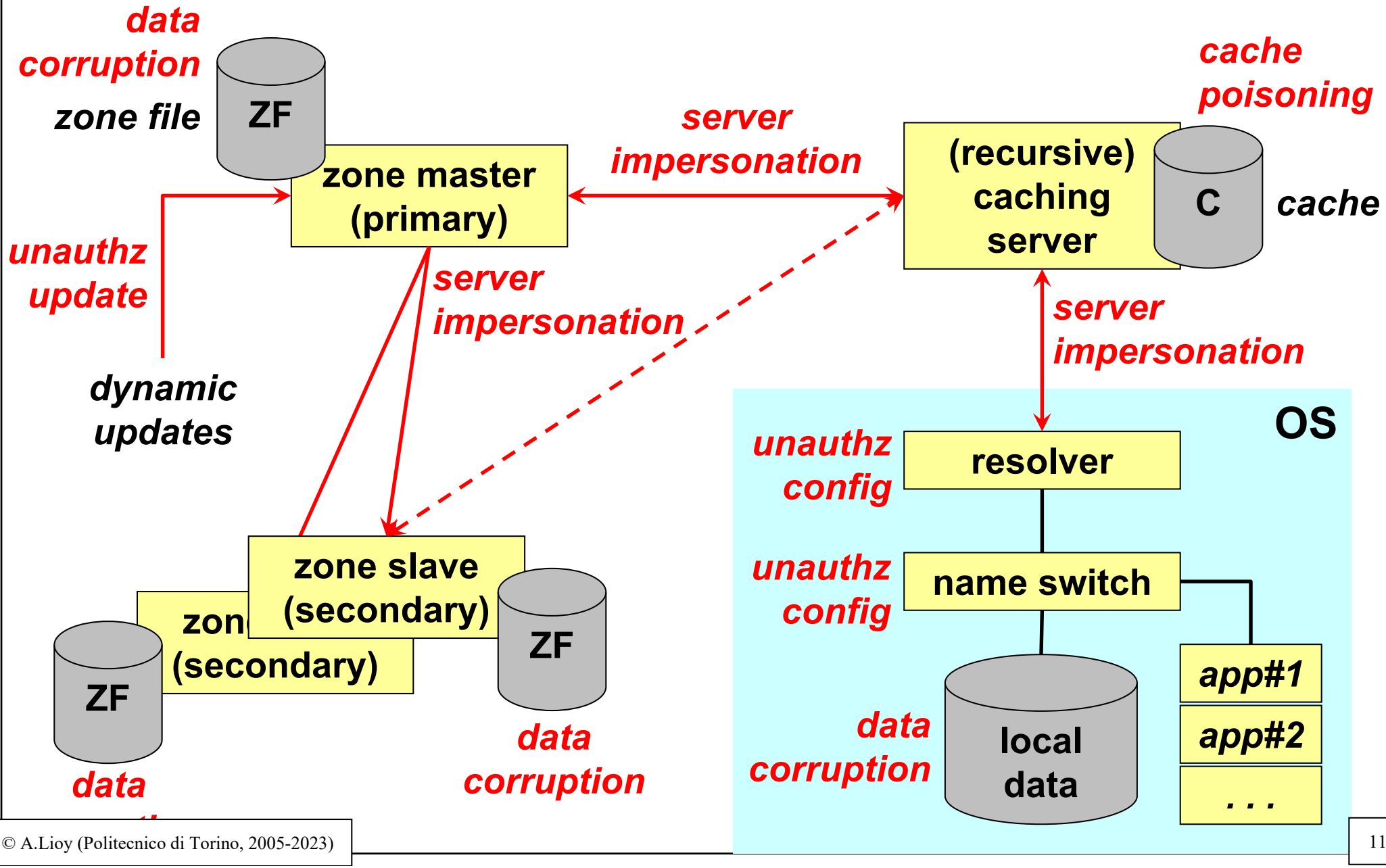
© A.Lioy (Politecnico di Torino, 2005-2023)

(DNS) flash crowd



© A.Lioy (Politecnico di Torino, 2005-2023)

Name-address translation



© A.Lioy (Politecnico di Torino, 2005-2023)

DoT and DoH

- **apart from attacks against the nameservers, DNS has got a user privacy problem for the queries:**
  - can be read while in transit
  - can be read and logged by the nameserver
- **DNS-over-TLS (DoT)**
  - query and response encapsulated in a secure TLS tunnel
  - but it is still evident that it's a DNS exchange
- **DNS-over-HTTPS (RFC-8484)**
  - query and response are part of a normal HTTPS exchange
  - externally it looks like visiting a secure web page
- **well-known service providers of DoH/DoT:**
  - Cloudflare (1.1.1.1) and Google (8.8.8.8 and 8.8.4.4)

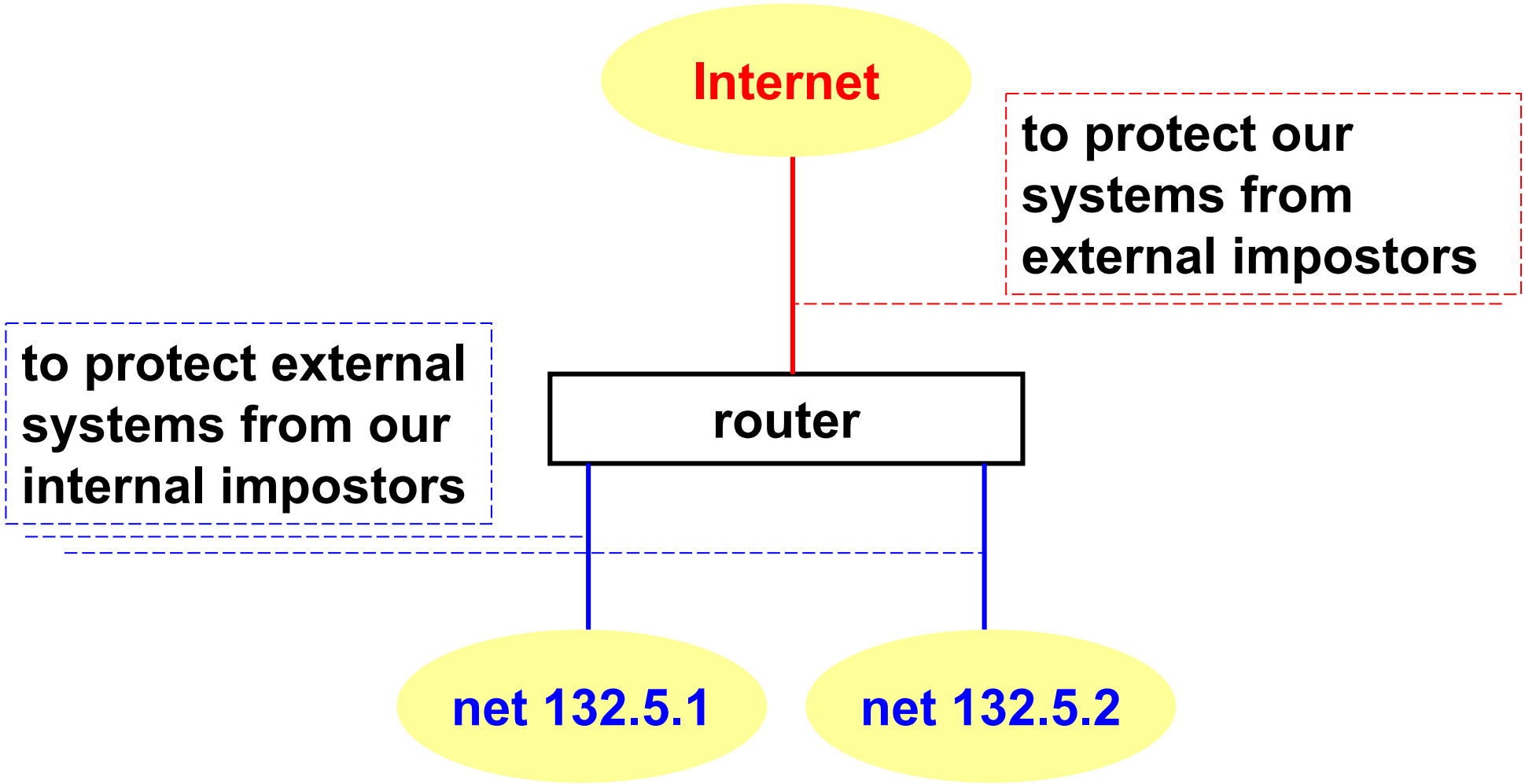
© A.Lioy (Politecnico di Torino, 2005-2023)

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”

© A.Lioy (Politecnico di Torino, 2005-2023)

Filters for IP spoofing protection



© A.Lioy (Politecnico di Torino, 2005-2023)

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
```

© A.Lioy (Politecnico di Torino, 2005-2023)