Politecnico di Torino

Security of IP networks

Diana Gratiela Berbecaru
diana.berbecaru@polito.it

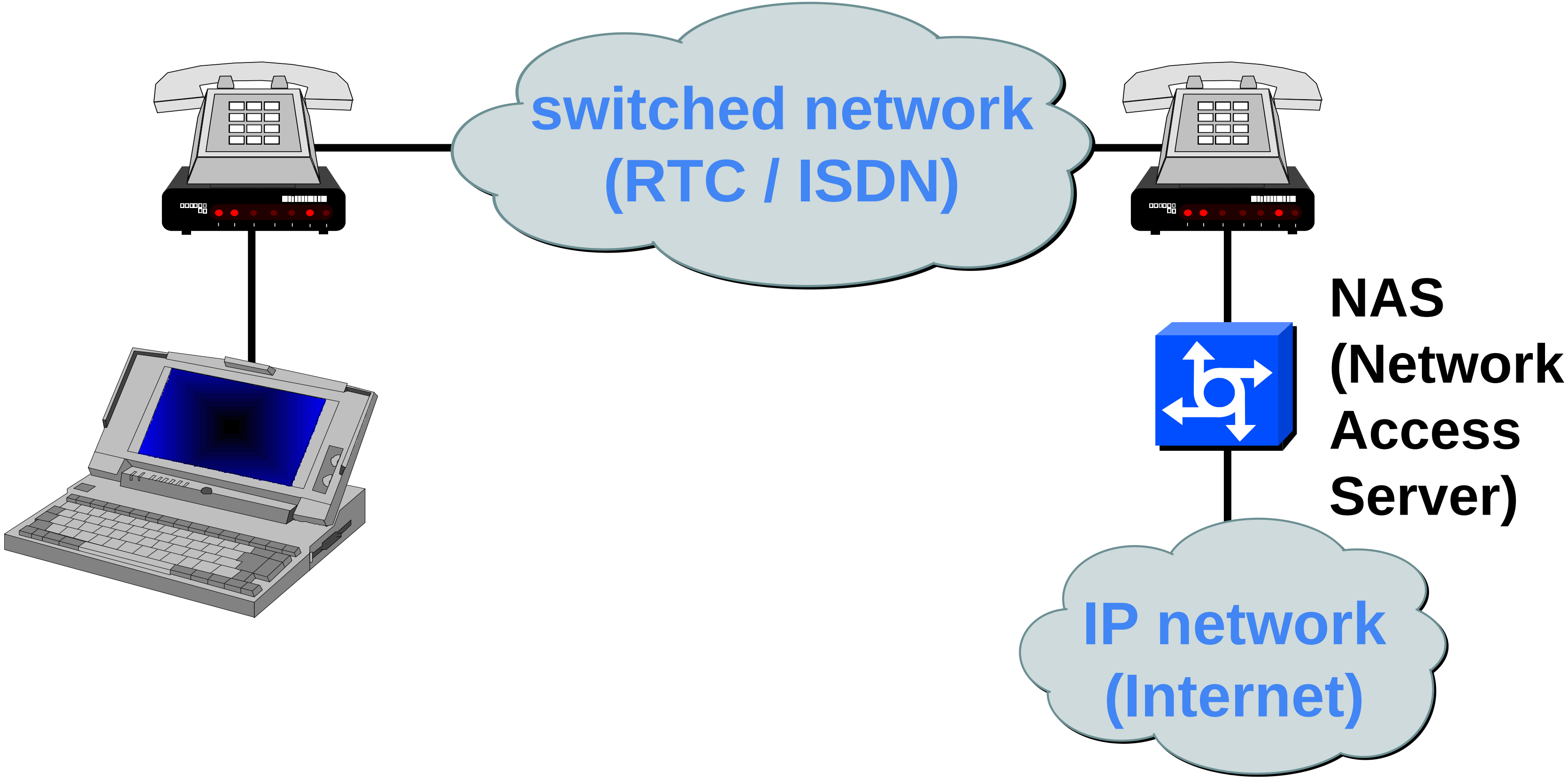
Politecnico di Torino
Dip. Automatica e Informatica

AY. 2023 - 2024

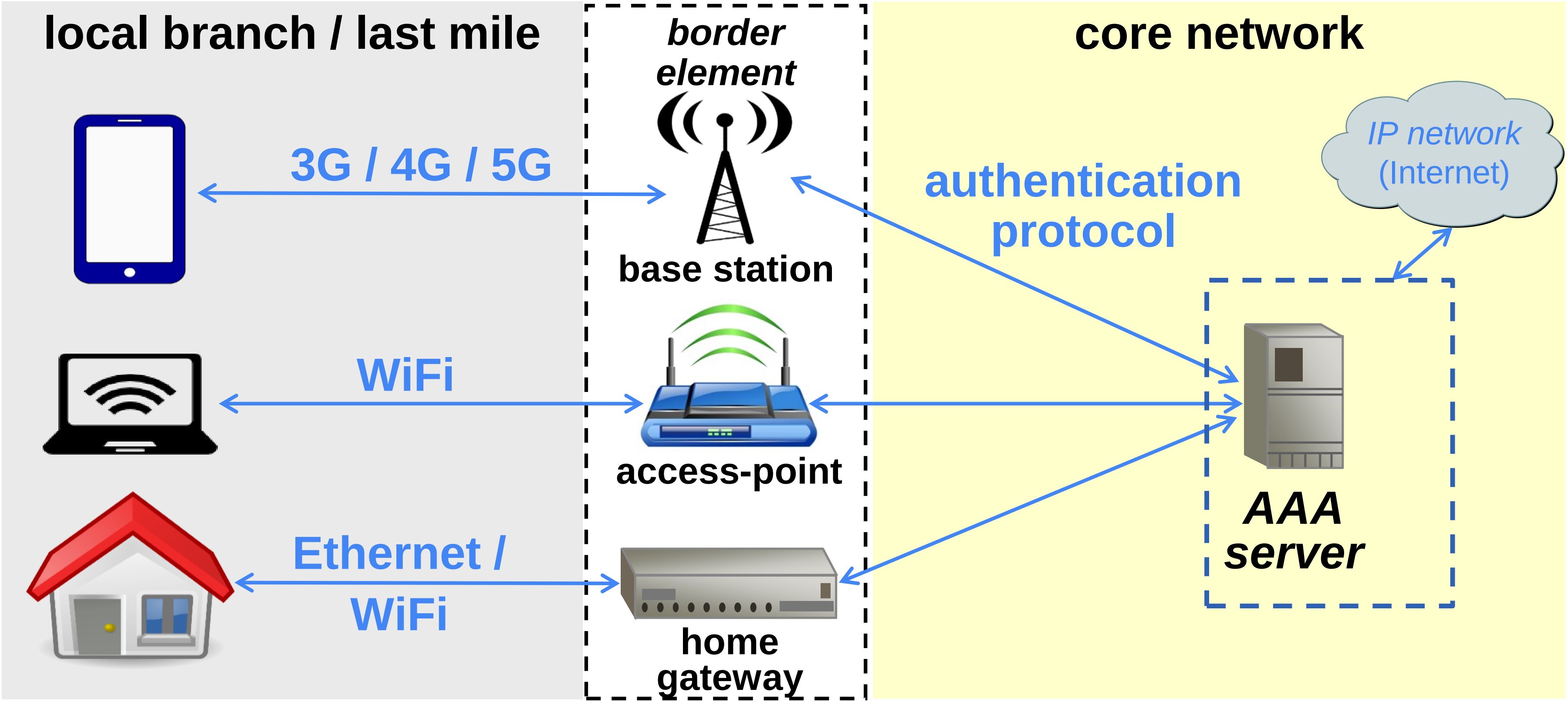
Acknowledgment

- **Slides content has been prepared by Prof. Antonio Lioy for the course Information Systems Security (2005 - 2022)**
  - modifications applied
- **… so this set of slides is entirely compatible with the course of the previous year(s)**
- **some figures have been imported from Chapter 12.3 «Security architecture: access control, EAP, RADIUS» in the book of Paul C. van Oorschot «Computer Security and the Internet: Tools and Jewels from Malware to Bitcoin» (https://people.scs.carleton.ca/~paulv/toolsjewels.html)**

Remote access via dial-up lines (old way)



Network access control (NAC) systems (modern way)



NAC systems deal with three types of components

- **Access requester (AR)**
  - node that is attempting to access the network and may be any device, including workstations, servers, printers, cameras, phones, and any other IP-enabled devices
  - also referred to as *supplicants*, or clients
- **Network access server (NAS)**
  - functions as an access control point for users in remote locations
  - may include its own authentication services or rely on a separate authentication service from the AAA server
- **AAA server**
  - determines what access should be granted to the AR
  - often relies on backend systems to help determine the AR's condition/health

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 (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 (RFC-1334) - obsolete**
  - user password sent in clear
  - authentication only once when the channel is created
- **CHAP - Challenge Handshake Authentication Protocol (RFC-1994) – not flexible, outdated too**
  - symmetric challenge (based on user password)
  - authentication request optionally repeated (with a challenge) during transmission – decision taken by the NAS
- **EAP – Extensible Authentication Protocol - modern**
  - external techniques (challenge, OTP, TLS)

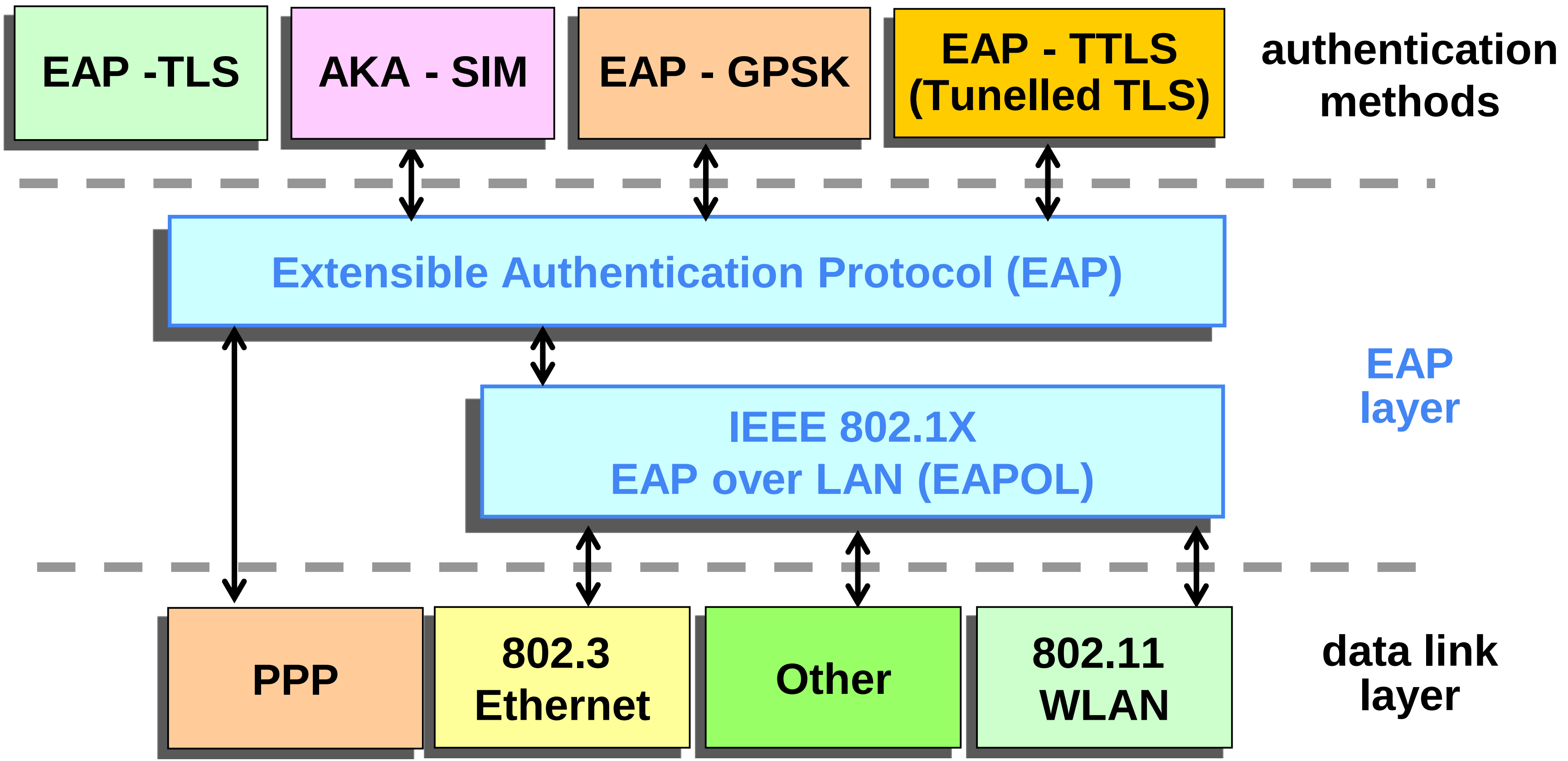
EAP

- **RFC-3748 (extended by RFC-5247)**
  “PPP Extensible Authentication Protocol (EAP)”
- **designed as a flexible L2 framework for network access and authentication protocols**
- **provides a generic transport service for the exchange of authentication information**
  - supports some originally predefined authentications mechanisms (e.g. MD5-challenge, OTP)
  - … but later on other mechanisms have been added, e.g. TLS
- **can operate over a variety of network and link level facilities, including point-to-point, LANs, and wireless**

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 - architecture



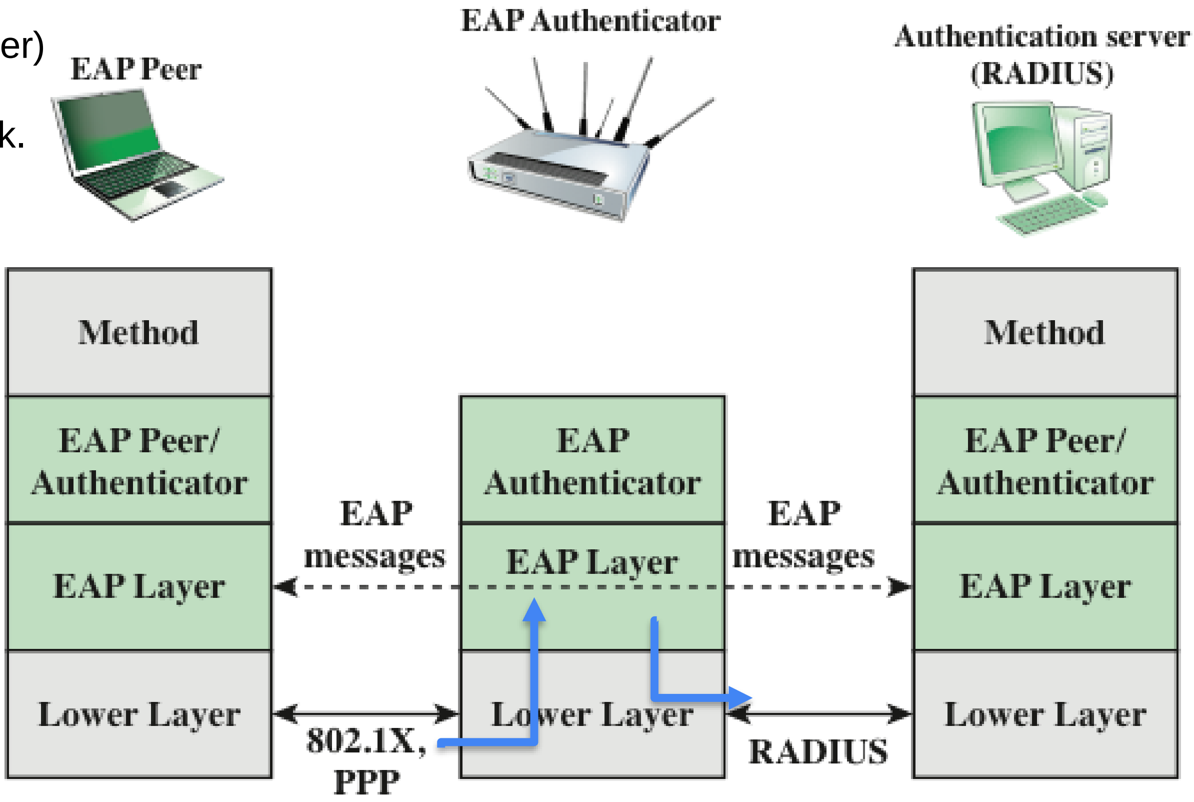
EAP

- **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-TTLS = tunnelled TLS (to operate any EAP method protected by TLS)
  - EAP - GPSK (Generic Pre-Shared Key)
  - AKA-SIM (RFC-4186, RFC-4187)

EAP Protocol Exchanges

EAP peer: Client (e.g. computer) that is attempting to access a network.

EAP authenticator: An access point or NAS that requires EAP authentication prior to granting access to a network.

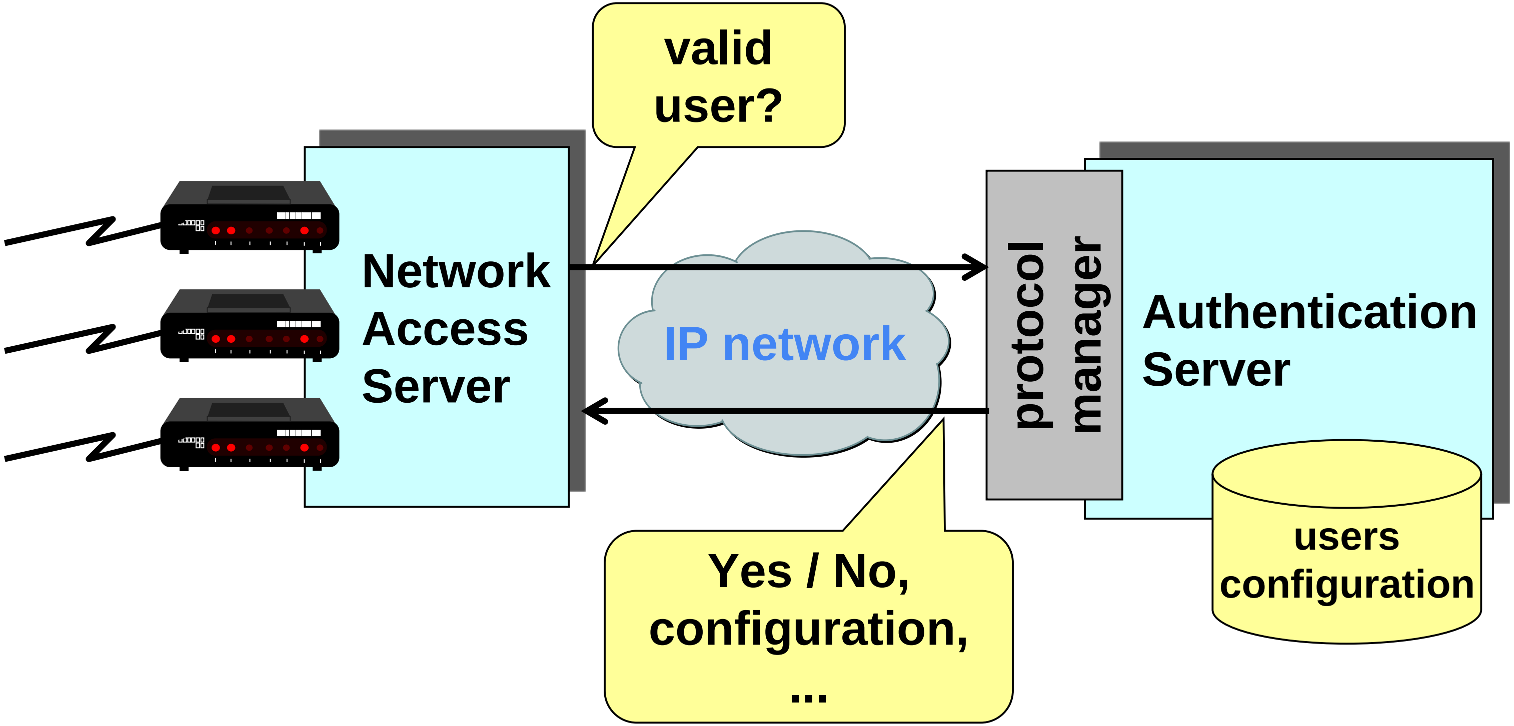


Authentication server: A server computer that negotiates the use of a specific EAP method with an EAP peer, validates the EAP peer’s credentials, and authorizes access to the network. Typically, the authentication server is a Remote Authentication Dial-In User Service (RADIUS) server.

Example: EAP-TLS

EAP Peer	Authenticator
	<< EAP-Request/Identity <>
EAP-Response/ Identity (MyID) ->	
	<< EAP-Request (type=EAP-TLS): <TLS Start>
EAP-Response/ EAP-Type=EAP-TLS (TLS client_hello)->	
	<- EAP-Request/ EAP-Type=EAP-TLS (TLS server_hello, TLS certificate, [TLS server_key_exchange,] TLS certificate_request, TLS server_hello_done)
EAP-Response/ EAP-Type=EAP-TLS (TLS certificate, TLS client_key_exchange, TLS certificate_verify, TLS change_cipher_spec, TLS finished) ->	
	<- EAP-Request/ EAP-Type=EAP-TLS (TLS change_cipher_spec, TLS finished)
EAP-Response/ EAP-Type=EAP-TLS ->	
	<- EAP-Success

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

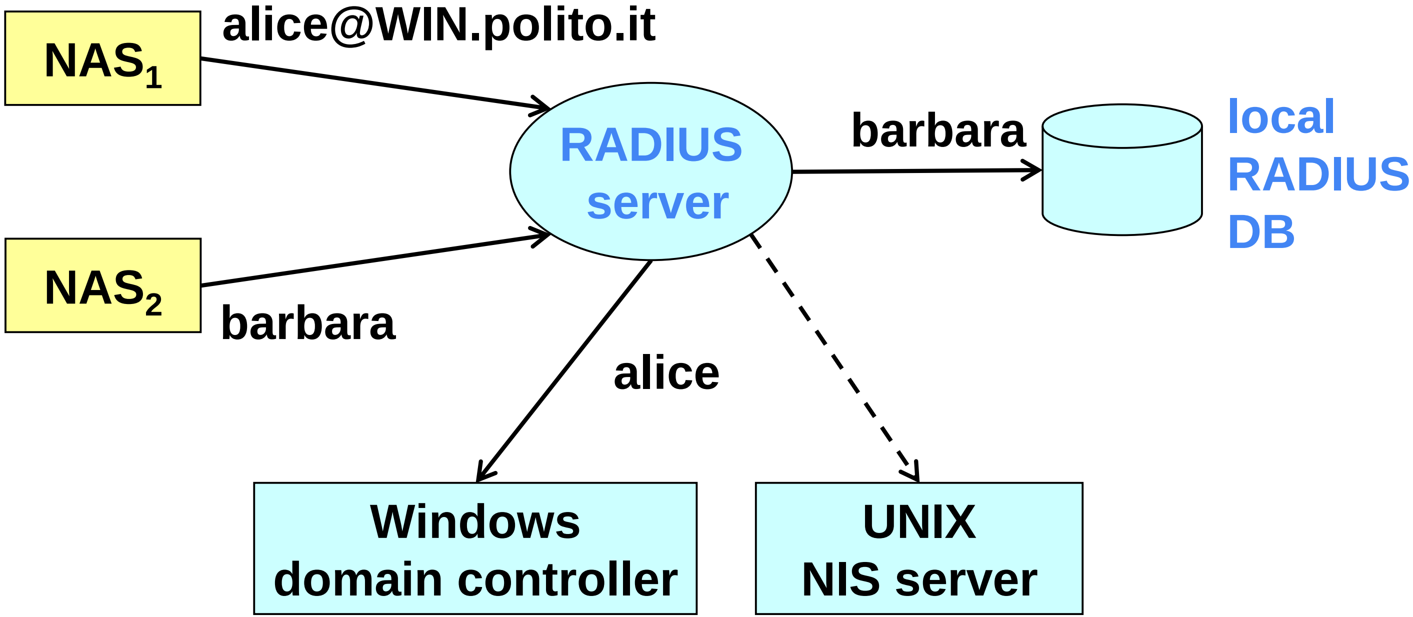
- **Remote Authentication Dial-In User Service**
- **Livingston Technologies (1991) then IETF**
- **supports authentication, authorization and accounting to control network access:**
  - physical ports (analogical, ISDN, IEEE 802)
  - virtual ports (tunnel, wireless access)
- **centralized administration 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?

- **what if an attacker is sniffing NAS req (if contains pwd)?**
  - we need protection for NAS req for confidentiality and privacy
- **what if an attacker issues fake AS resp (to block valid or allow invalid user) or changes AS resp (Y > N or N > Y)**
  - we need protection for AS resp for data authN & integrity
- **what if an attacker replays an AS resp?**
  - we need protection protection against replays of AS resp (by properly tying them to NAS req)
- **what if fake NAS tries to perform pwd enumeration?**
  - we need authN of NAS req
- **what if (too) many NAS reqs arrive from a fake NAS? DoS!**
  - 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):**

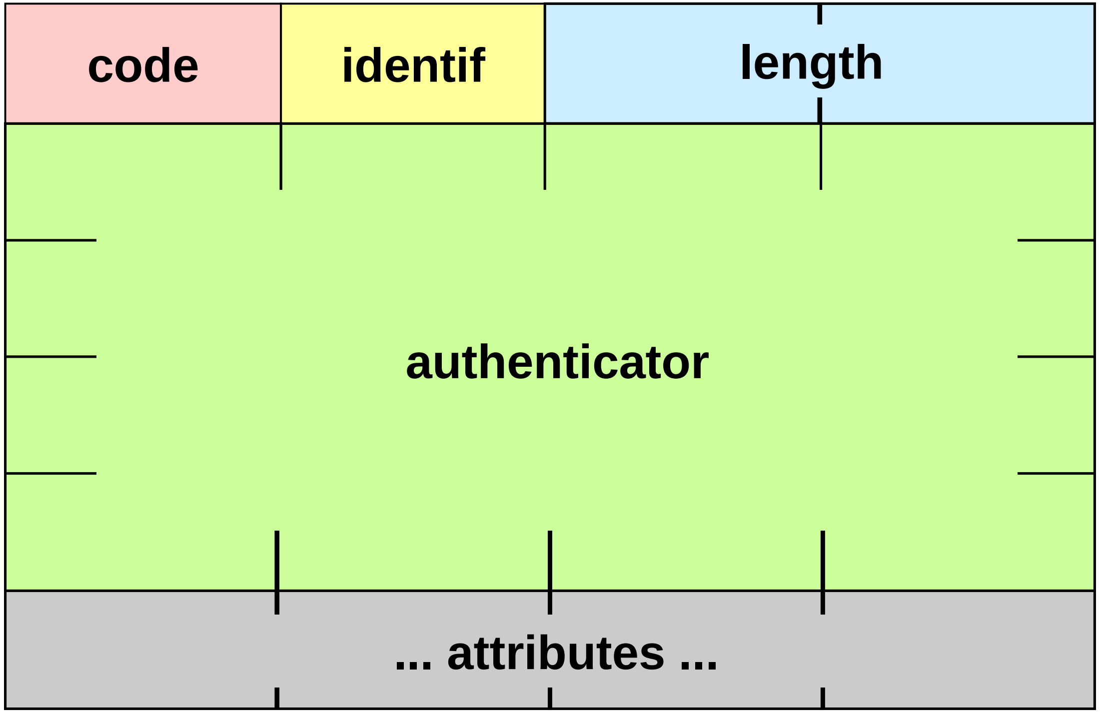
password ⊕ md5(key+authenticator)

RADIUS

- **user authentication via PAP, CHAP, token-card and any other EAP (methods)**
- **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: Framed-Protocol, Framed-IP-Address, Framed-IP-Netmask, MS-Primary-DNS-server, MS-Primary-DNS-server,..
    - for terminal: host, port

RADIUS - authenticator

- **double purpose:**
  - in the request sent to RADIUS server: masks the password
  - in the response created by RADIUS server: provides authentication of the responses and protection from replay attacks
- **in Access-Request:**
  - it is named *Request Authenticator (RequestAuth)*
  - 16 bytes 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

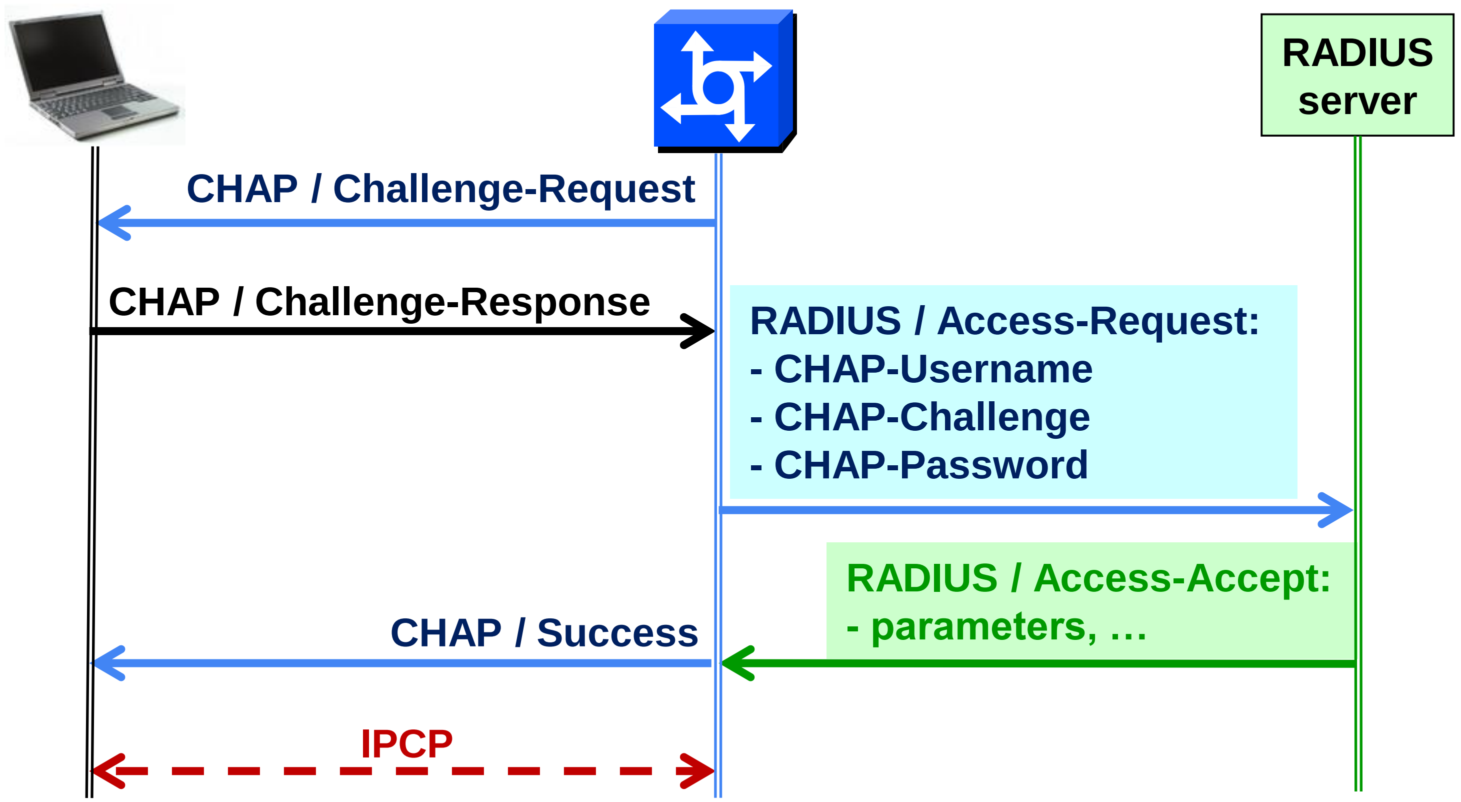
type	length	value

- **type = 1 (User-Name)**
  - value = text, network access identifier (NAI), DN
- **type = 2 (User-Password)**
  - value = password ⊕ md5 (key || RequestAuth)
- **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



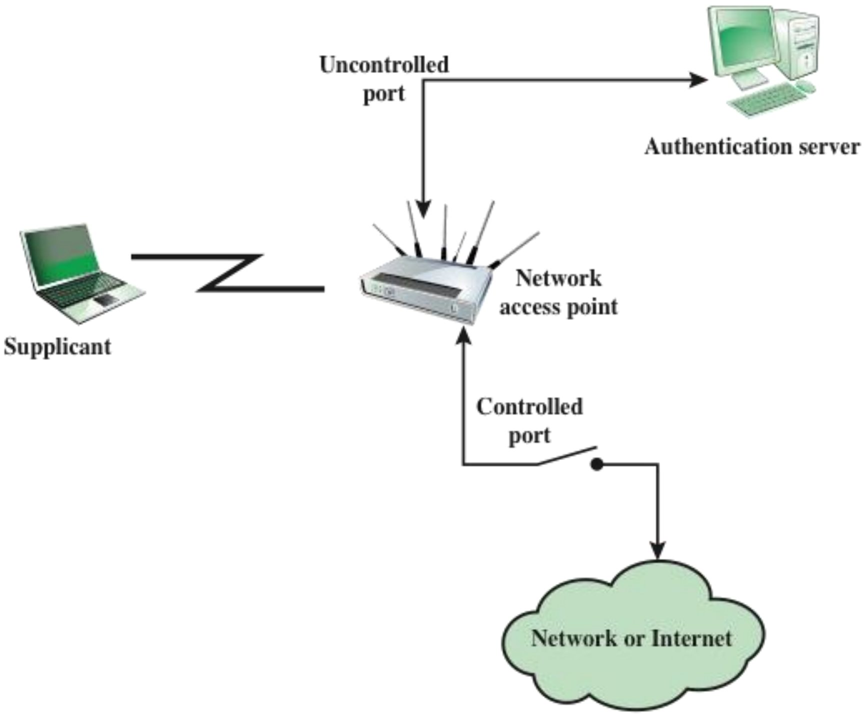
IEEE 802.1X

- **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

802.1X access control: ports

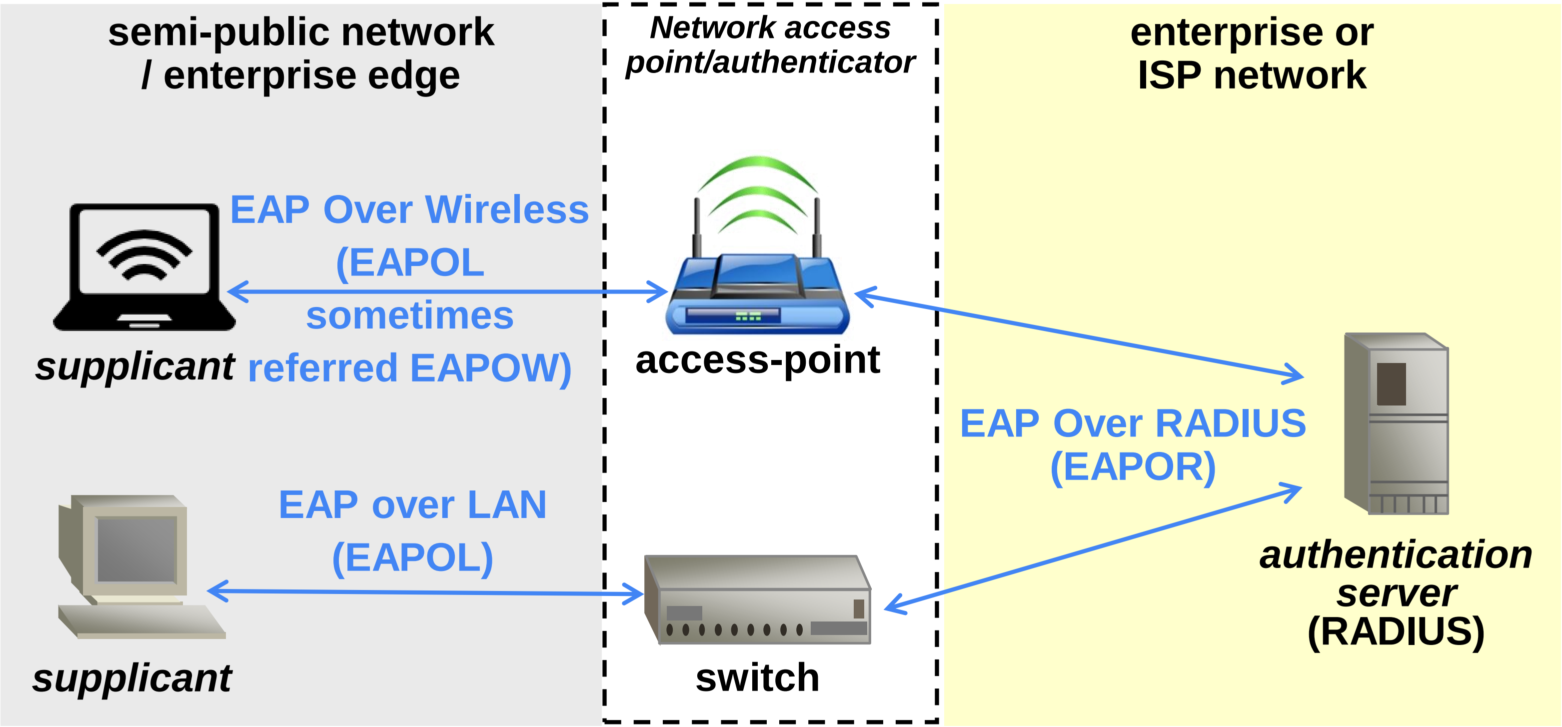
- **an <u>uncontrolled port</u> is always enabled, but limited: it allows only authenticated-related messages between supplicant and AS**
- **a <u>controlled port</u>: begins disabled, preventing exchange of data frames with the rest of the network**
  - after successful authentication, allows the exchange of messages between a supplicant and other systems on the network



IEEE 802.1X

- **provides an 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)

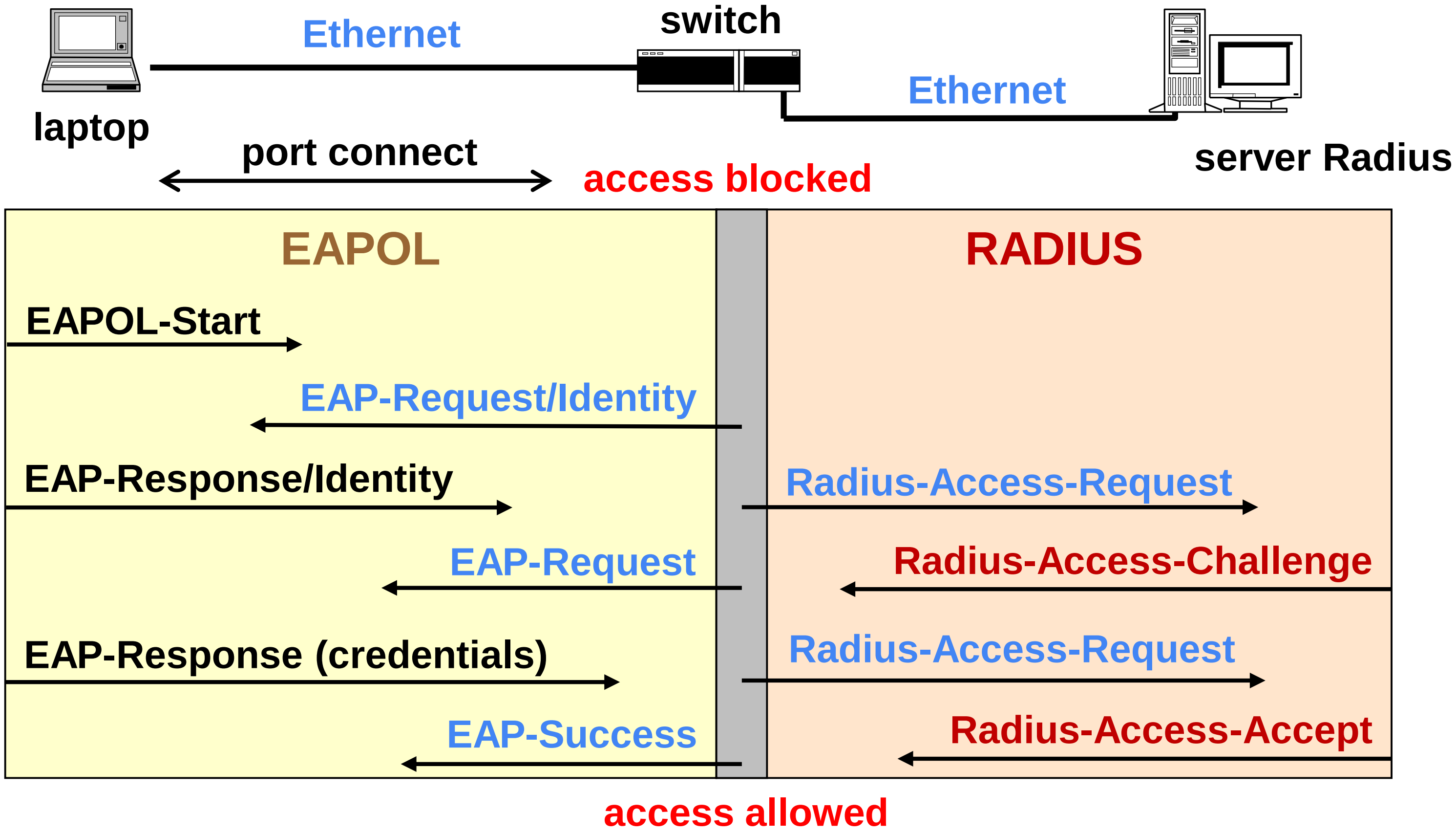
802.1X – typical use



802.1X – NAS “pass-through device”

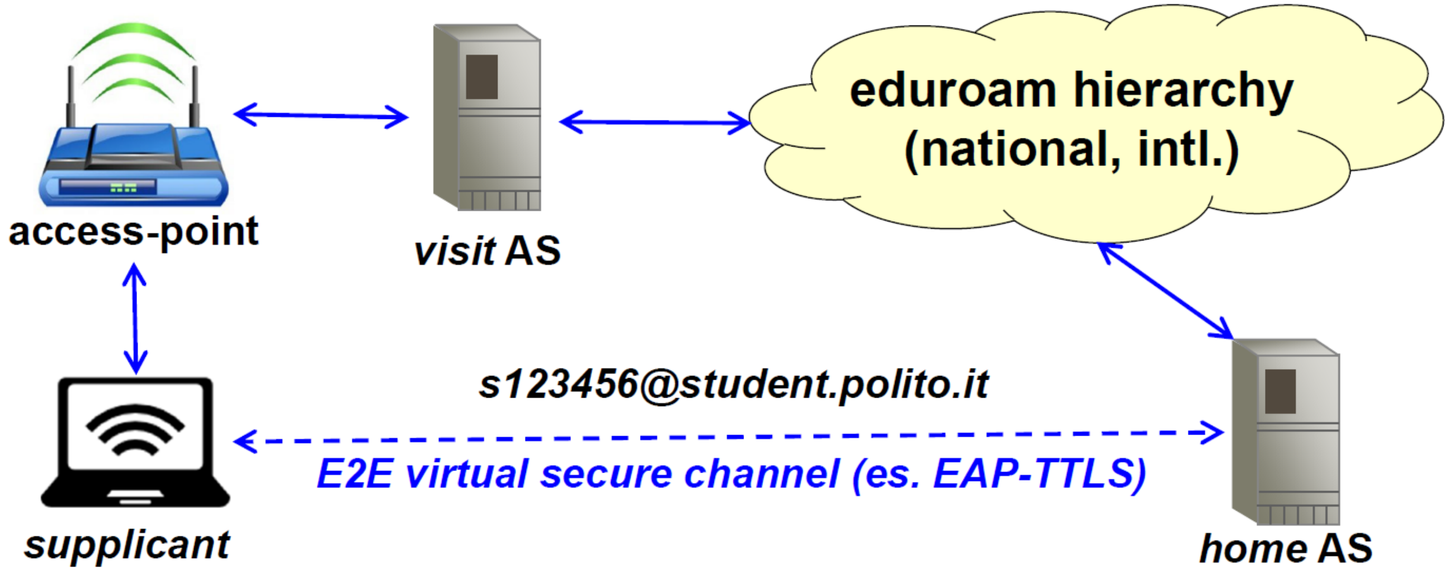
- **exploits the application level for the actual implementation of the security mechanisms**
  - direct dialogue between supplicant and AS
    - NIC and NAS operate as “pass-through device”
  - no change needed on NIC and NAS to implement new mechanisms
  - perfect integration in AAA

802.1X – messages (supplicant via Ethernet)

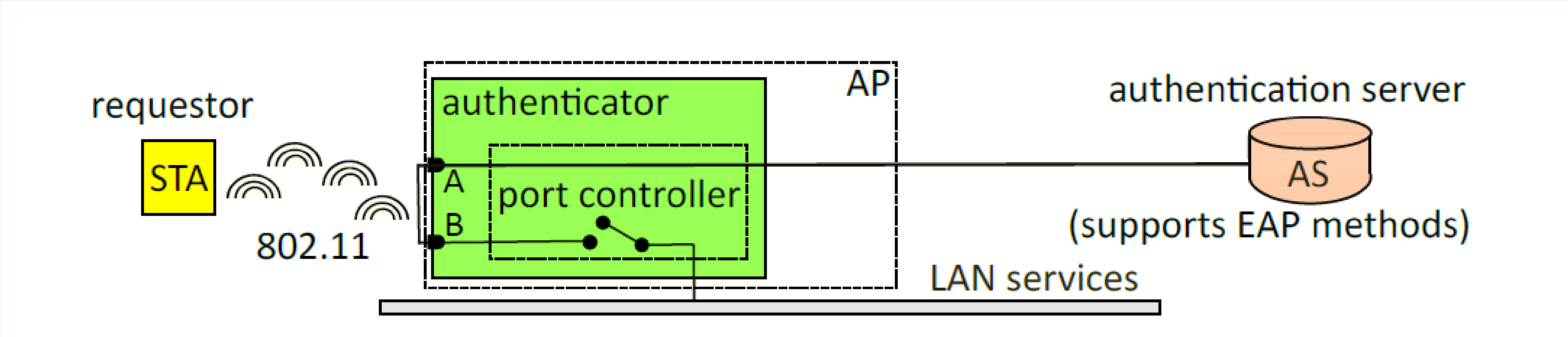


eduroam

- **WiFi access at research institutes (Italy, Europe, …)**
  - (21/11/2021) 106 countries
  - uses 802.1x + RADIUS federation

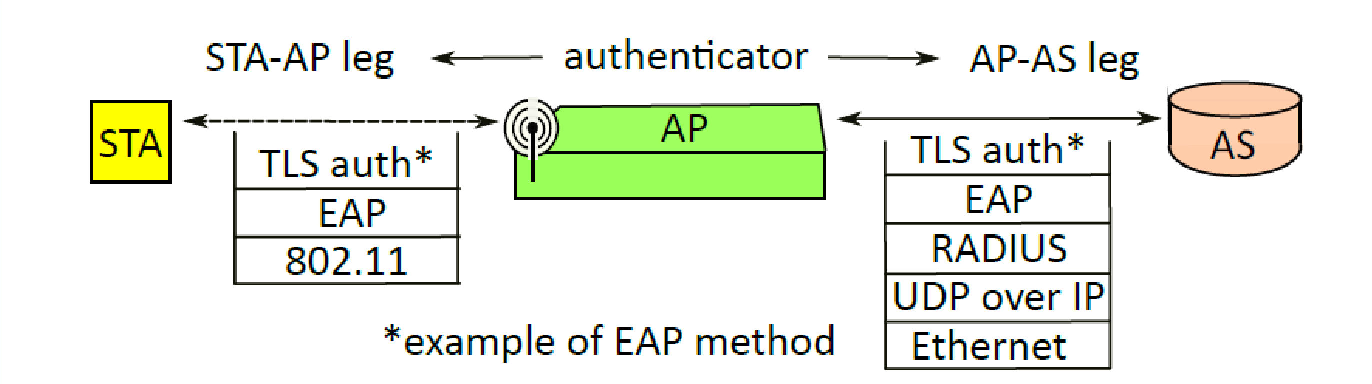


Example: 802.1X use in WLAN



- an optional EAP method between STA (station) and AS generates a pairwise master key (PMK), which is transferred from AS to AP (Access point) (and known by STA)
- upon success of a PMK-based handshake (four-pass handshake) between STA and AP, the authenticator will enable port B.
  - session keys for protecting packets are derived as well
- before then, authentication-related traffic (formatted in EAP messages) is all that is accepted from STA, for relay to AS (as EAP over RADIUS)

Example: 802.1X use in WLAN



- the *authenticator* function in the AP relays messages between the STA and AS in two legs
- network protocol sublayers not shown are EAPOL (between EAP and 802.11), EAP-over-RADIUS and TLS-over-EAP; such sub-layers are often needed when refitting old protocols for new purposes

Security attacks protection

DHCP (in)security

- **non-authenticated (!) broadcast (!) protocol providing:**
  - IP address, netmask, default gateway
  - local DNS nameserver (IP address)
- **activation of a fake DHCP server is trivial**
  - because the DHCP request is L2 broadcast
  - DHCP responses are non-authenticated

DHCP attacks

- **possible attacks from the fake DHCP 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
  - malicious name-address translation (e.g. for phishing)

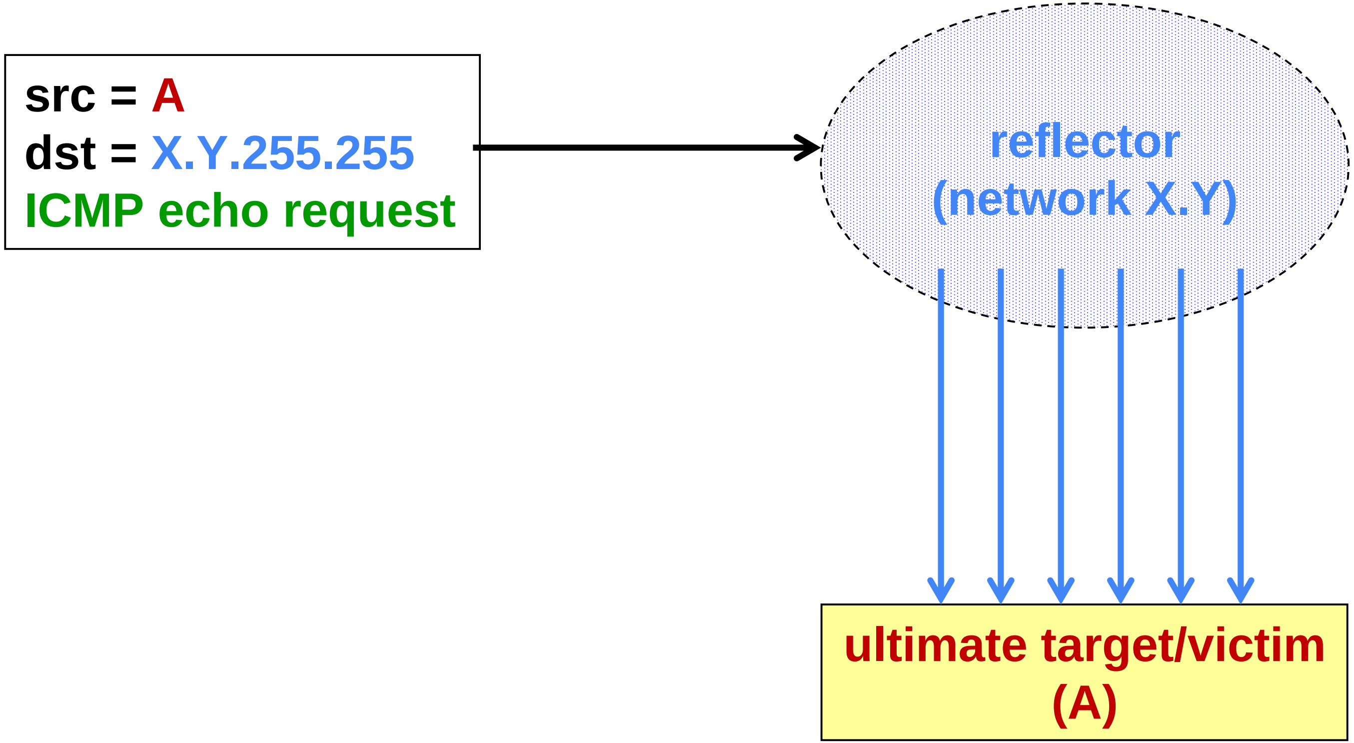
DHCP protection

- **RFC-3118 “Authentication for DHCP messages”**
  - use of HMAC-MD5 to authenticate the messages
  - problem = key distribution and management (shared key!)
  - rarely adopted
- **protection at upper layers (e.g. with TLS)**
  - MITM attack is countered !

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



Anti-smurfing countermeasures

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

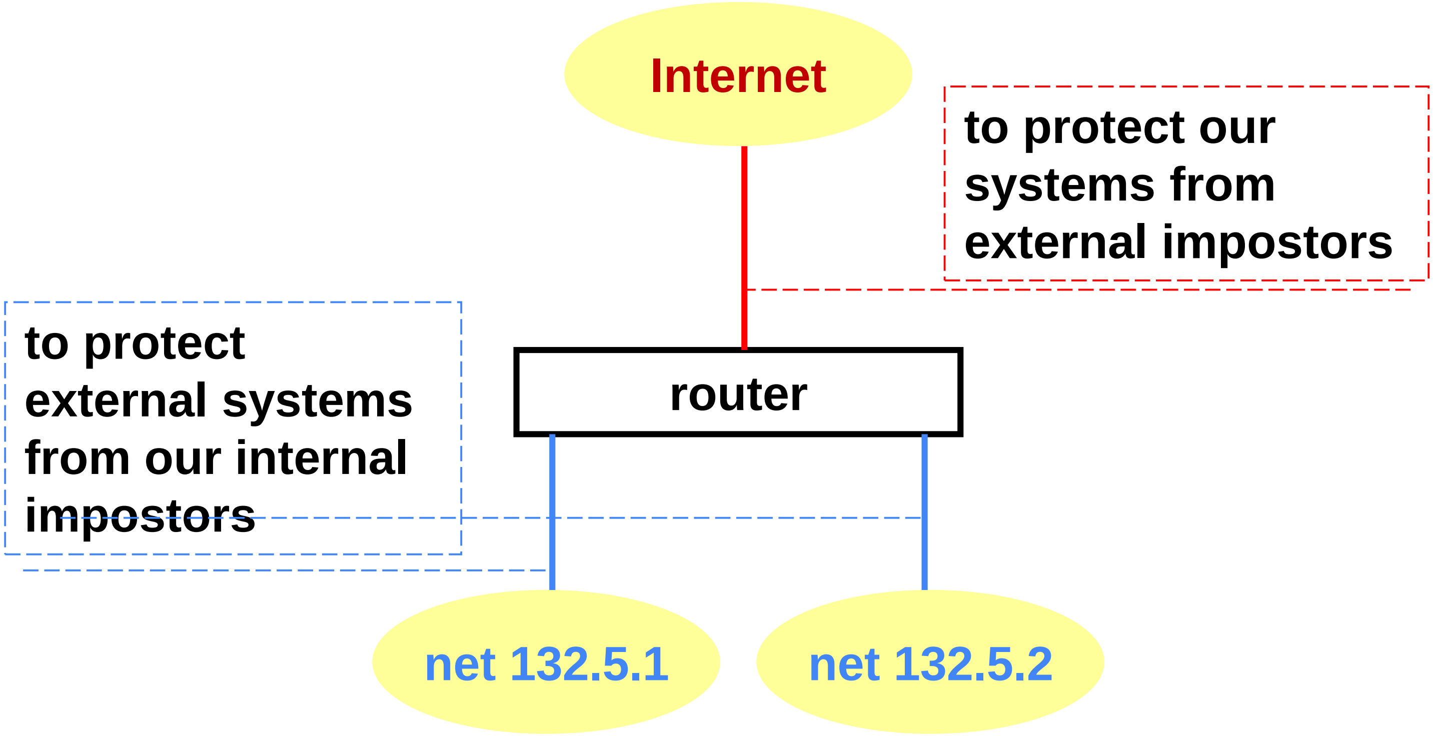
```
interface serial0
no ip directed-broadcast
```

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

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”**

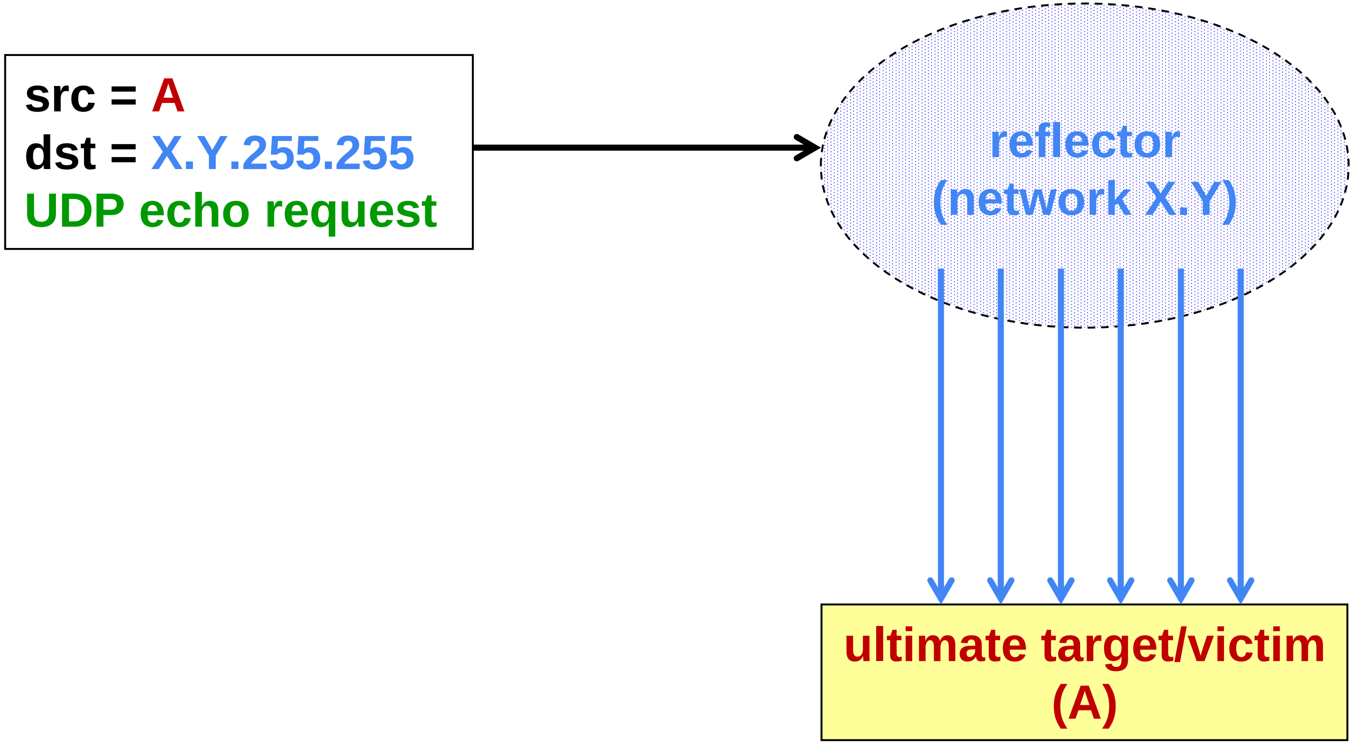
Filters for IP spoofing protection



Example of IP spoofing protection

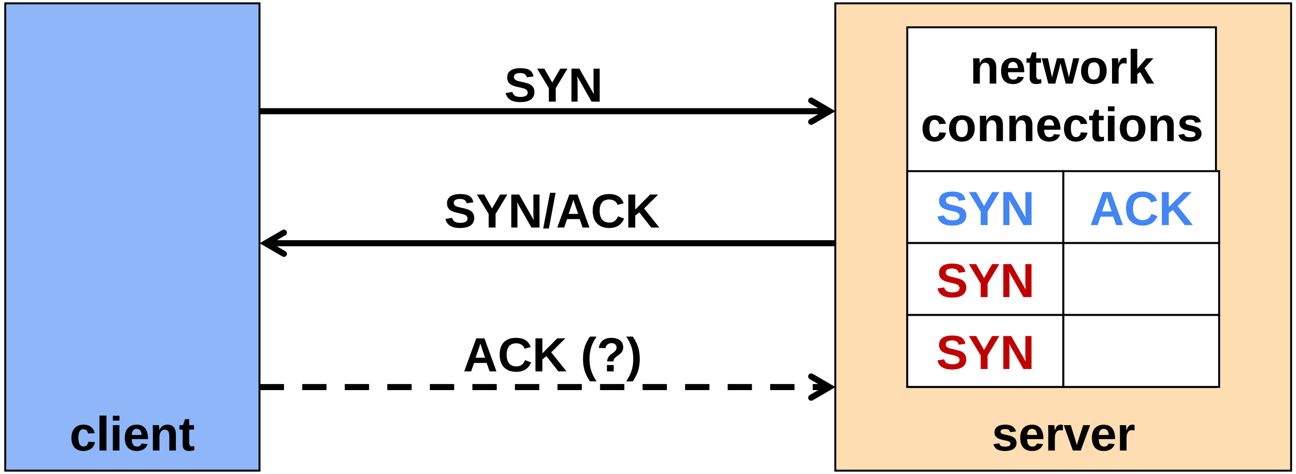
```
access-list 101 deny ip
   132.5.0.0 0.0.255.255 0.0.0.0 255.255.255.255
interface serial 0
ip access-group 101 in

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
```

Fraggle attack



TCP SYN flooding

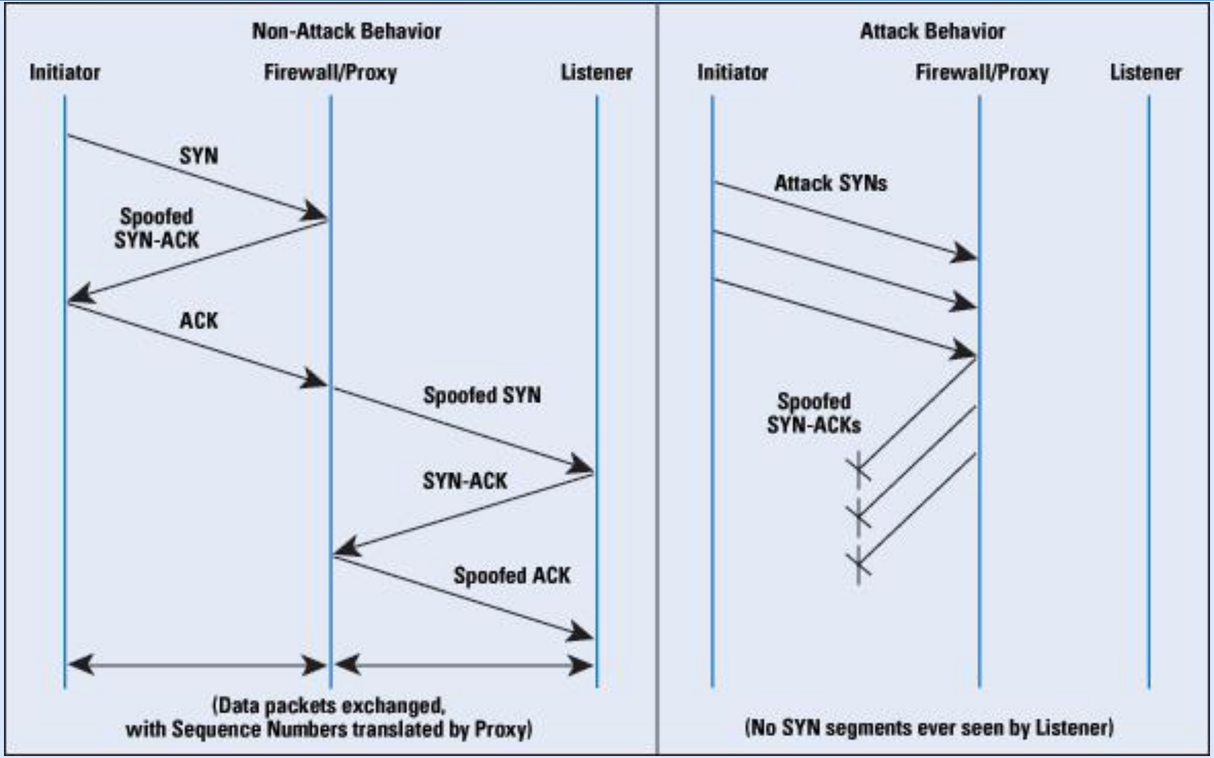
- multiple requests with IP spoofing
- the connection table is saturated until half-open connections timeout (typical value: 75”)



Protection against SYN flooding

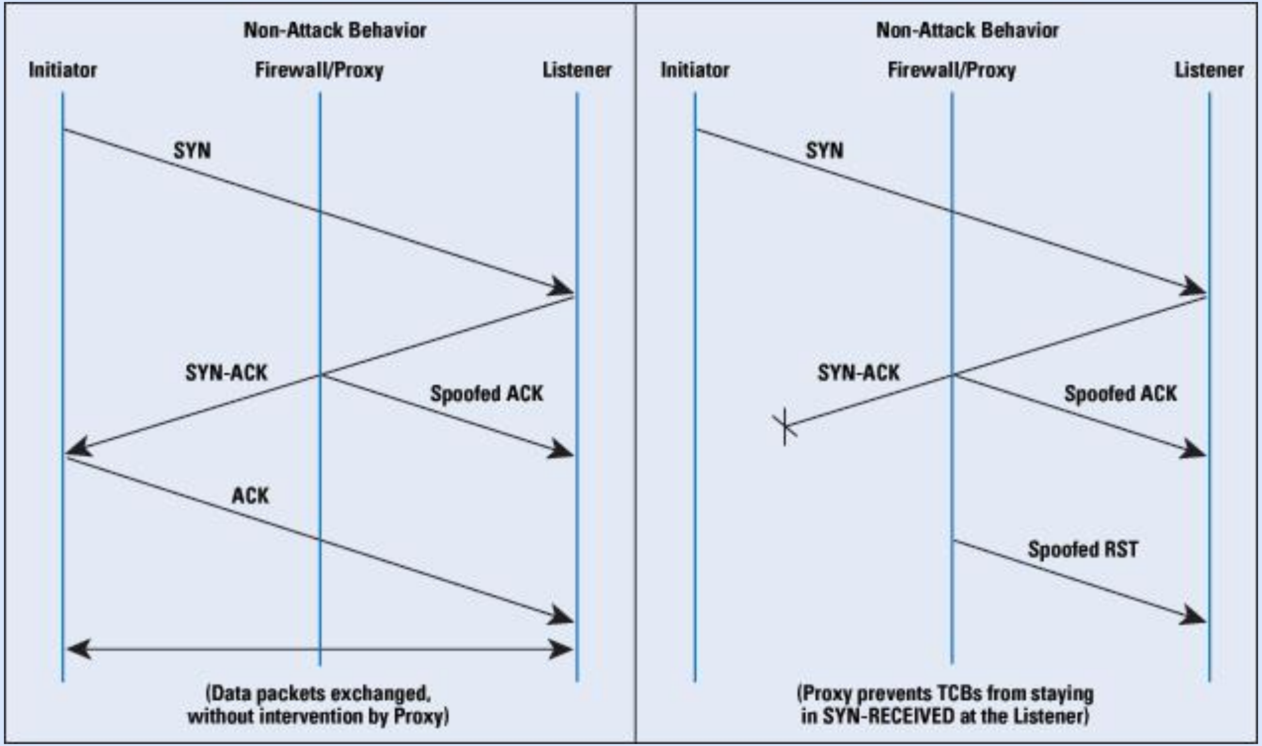
- **decrease the timeout**
  - risk to delete requests from valid but slow clients
- **increase the table size**
  - can be circumvented by sending more requests
- **use a router as "SYN interceptor":**
  - substitutes the server in the first phase
  - if the 3-way handshake completes successfully, then 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/ipj_9-4/syn_flooding_attacks.html

SYN monitor (or firewall gateway)



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

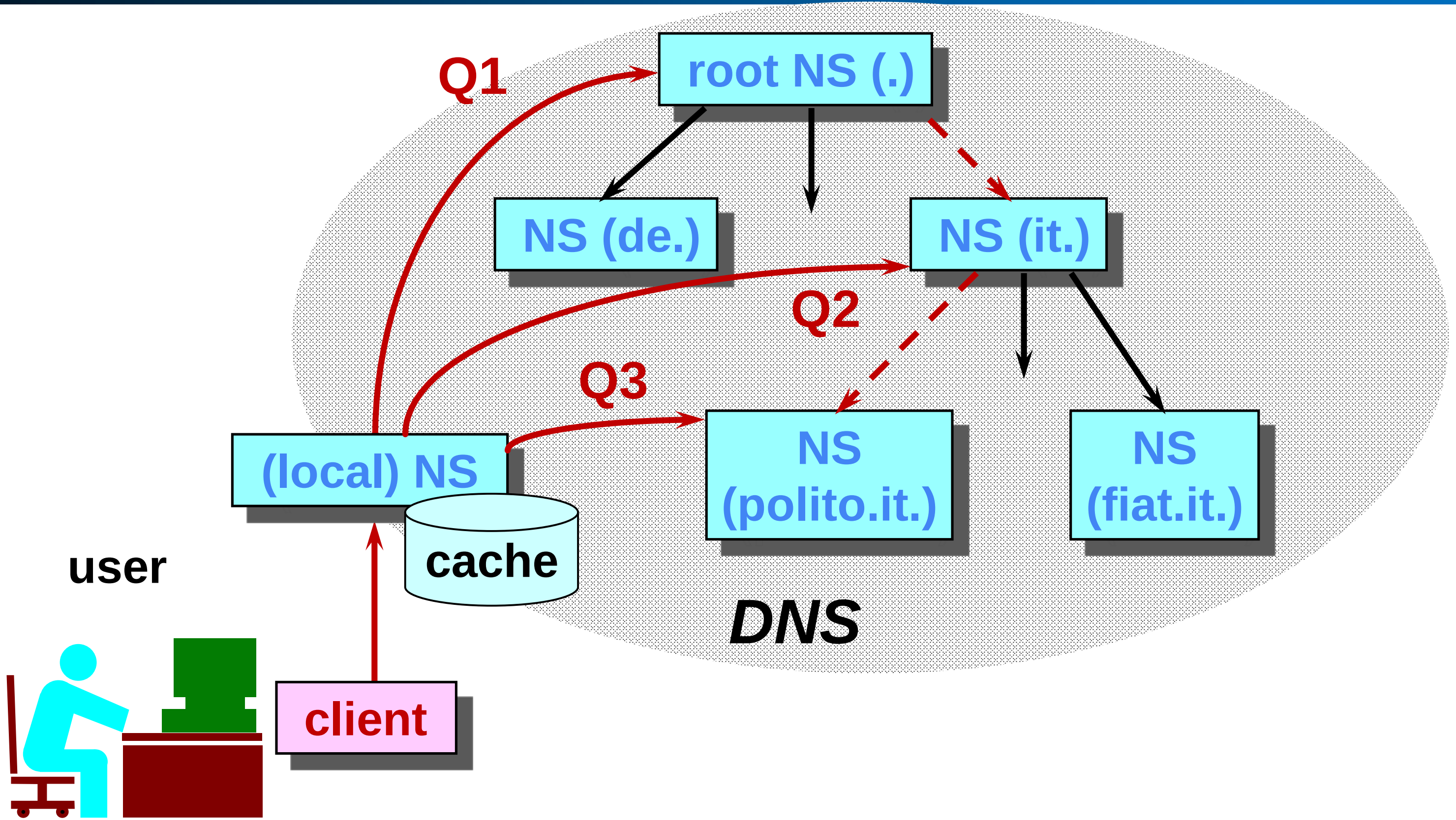
SYN cookie

- **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

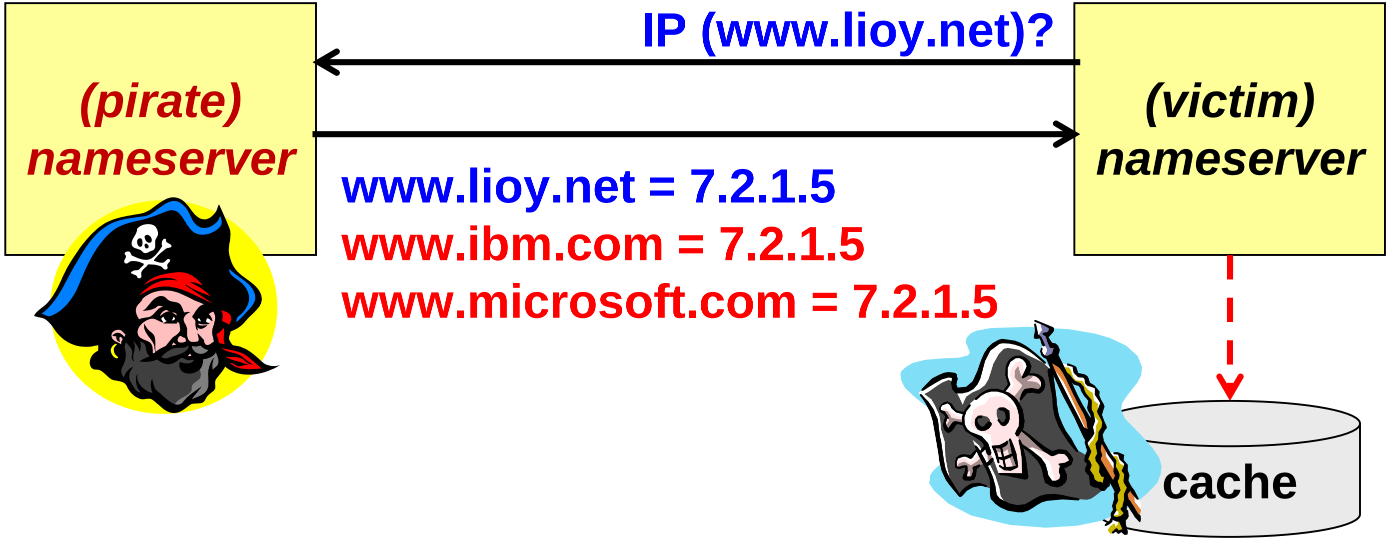
- DNS (Domain Name System)
- translation:
  - from names to IP addresses
  - from IP addresses to names
- vital service
- queries over port 53/UDP
- zone transfers over port 53/TCP
- no security
- DNS-SEC - under development

DNS architecture (iterative query)



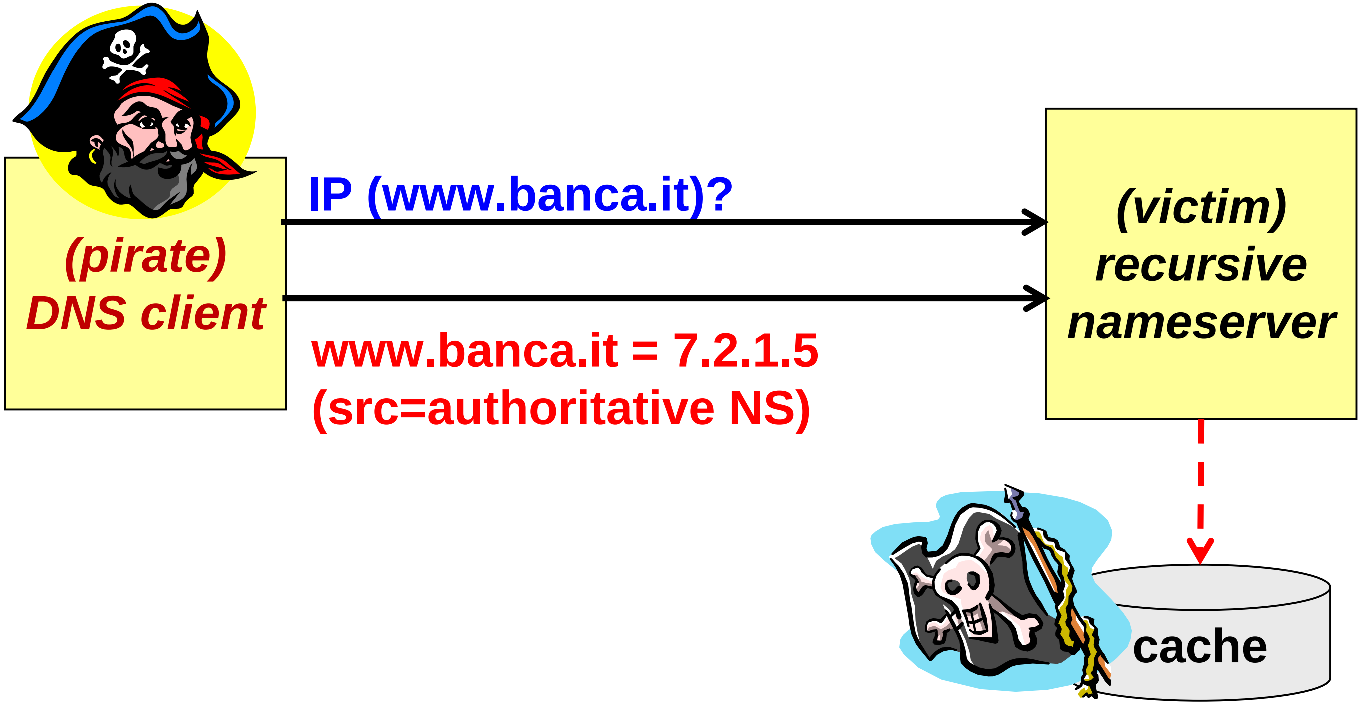
DNS cache poisoning

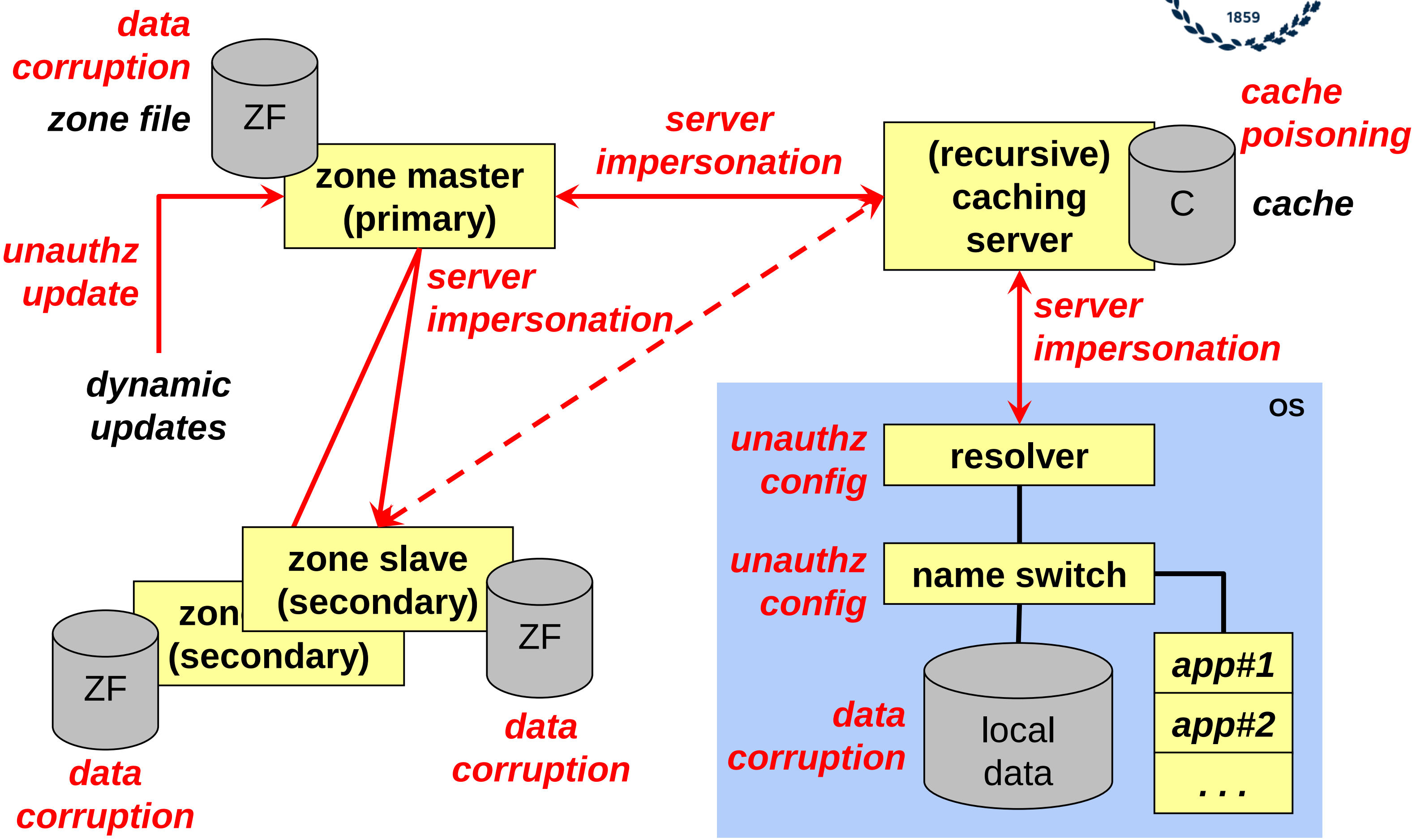
- attract the victim to make a query on my NS
- provide answers also to queries never done to push / overwrite the victim's cache



DNS cache poisoning (2nd version)

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




data corruption
zone file
ZF
zone master (primary)
server impersonation
(recursive) caching server
C
cache poisoning
cache
unauthz update
server impersonation
server impersonation
dynamic updates
OS
unauthz config
resolver
unauthz config
name switch
zone slave (secondary)
zon
(secondary)
ZF
ZF
local data
app#1
app#2
. . .
data corruption
data corruption
data corruption

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)
  - details in: "Fresh Phish", IEEE Spectrum, oct'08, doi: 10.1109/MSPEC.2008.4635052
- **(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**

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**
  - hierachical 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?

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's still evident that it's a DNS exchange
- **DNS-over-HTTPS (DoH) (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)**