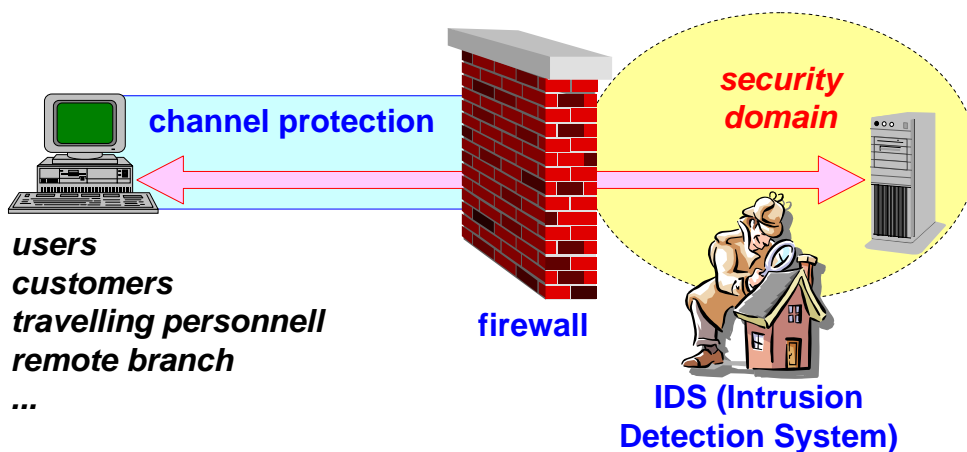


# Network security

**Antonio Lioy**  
< [lioy @ polito.it](mailto:lioy@polito.it) >

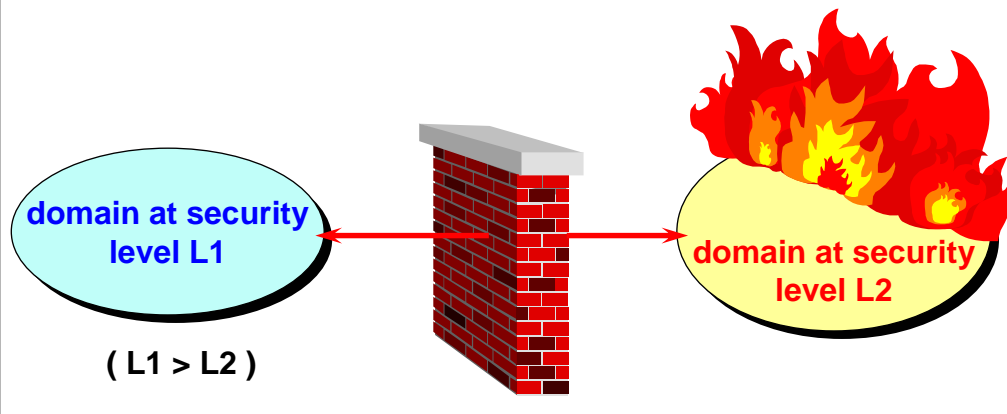
**Politecnico di Torino**  
**Dip. di Automatica e Informatica**

## Network security: basic components



## What is a "firewall"?

- firewall = anti-fire wall (not "wall of fire")
- controlled connection between two networks at different security levels = perimeter protection



### THE THREE FIREWALL BASIC RULES

- I. the FW must be the unique contact point between the internal and external networks
- II. only the "authorized" traffic can traverse the FW
- III. the FW itself must be an highly secure system

*D.Cheswick  
S.Bellovin*

## Authorization policies

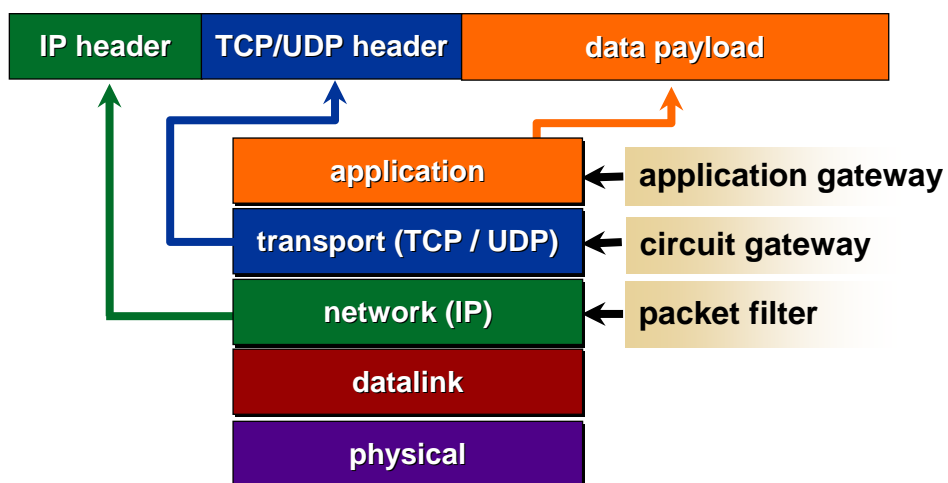
“All that is not explicitly permitted, is forbidden”

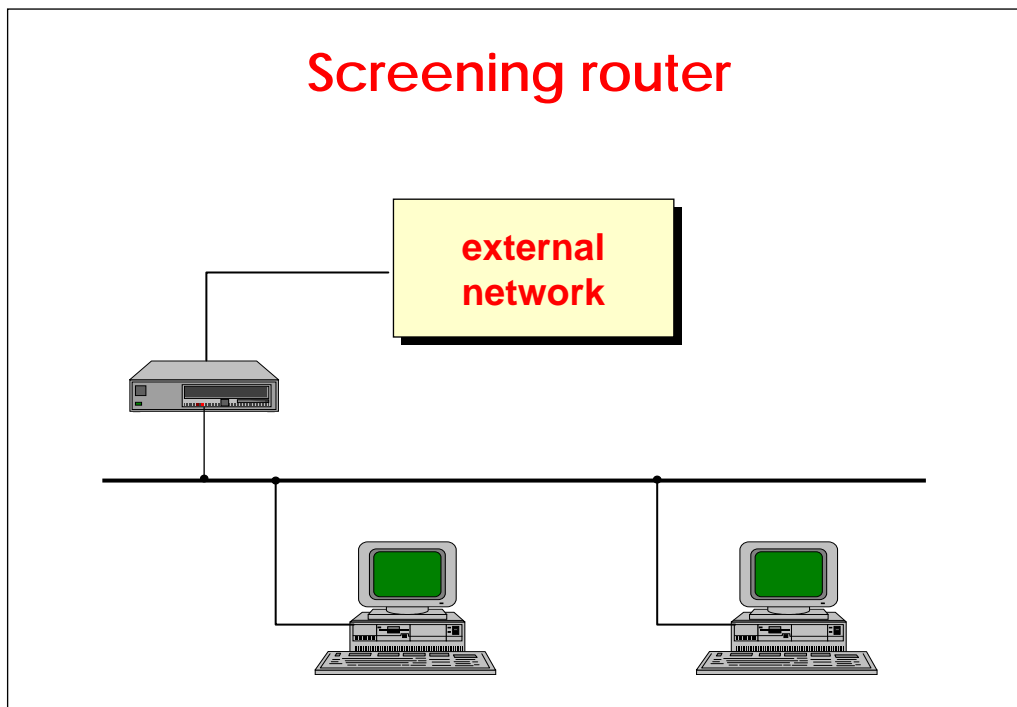
- greater security
- more difficult to manage

“All that is not explicitly forbidden, is permitted”

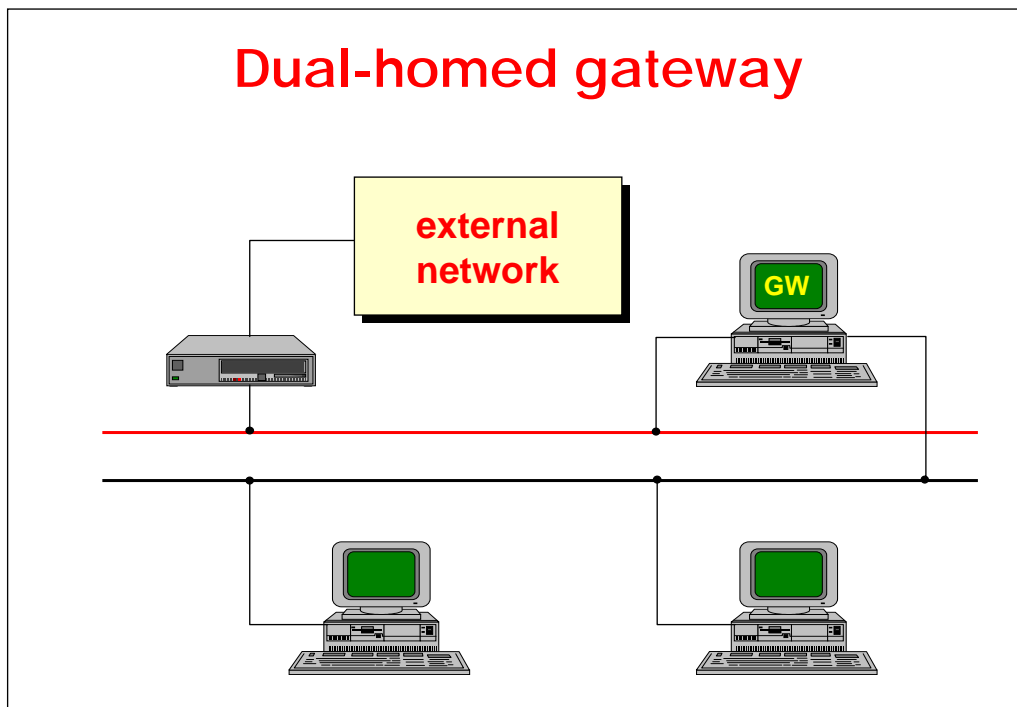
- lower security (open gates)
- easier to manage

## At what level the control is performed?

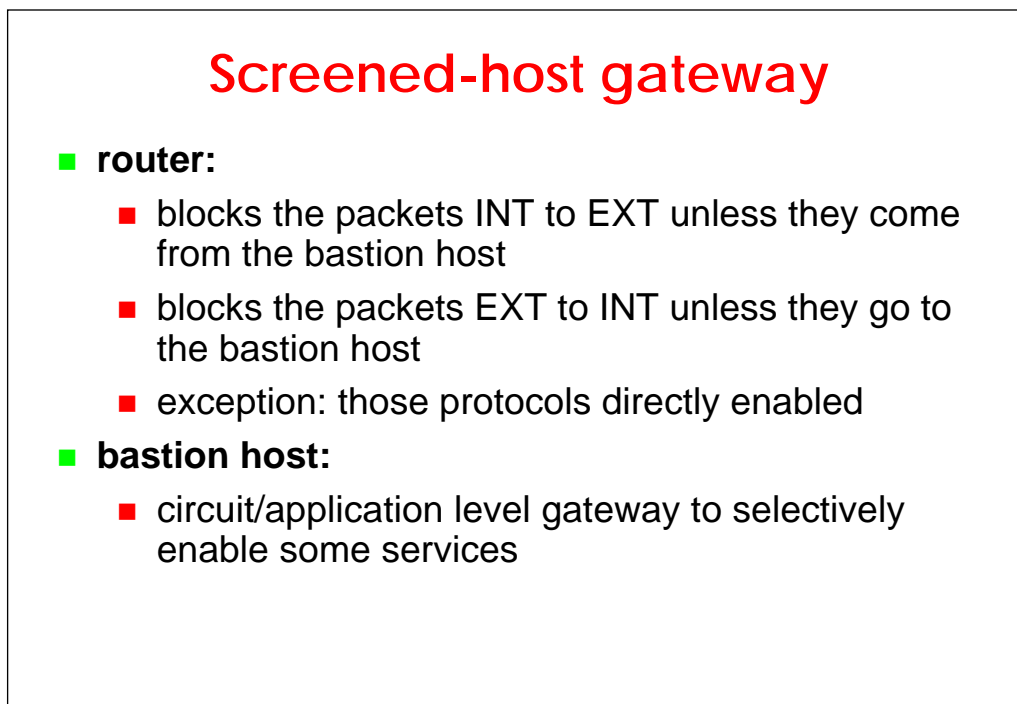
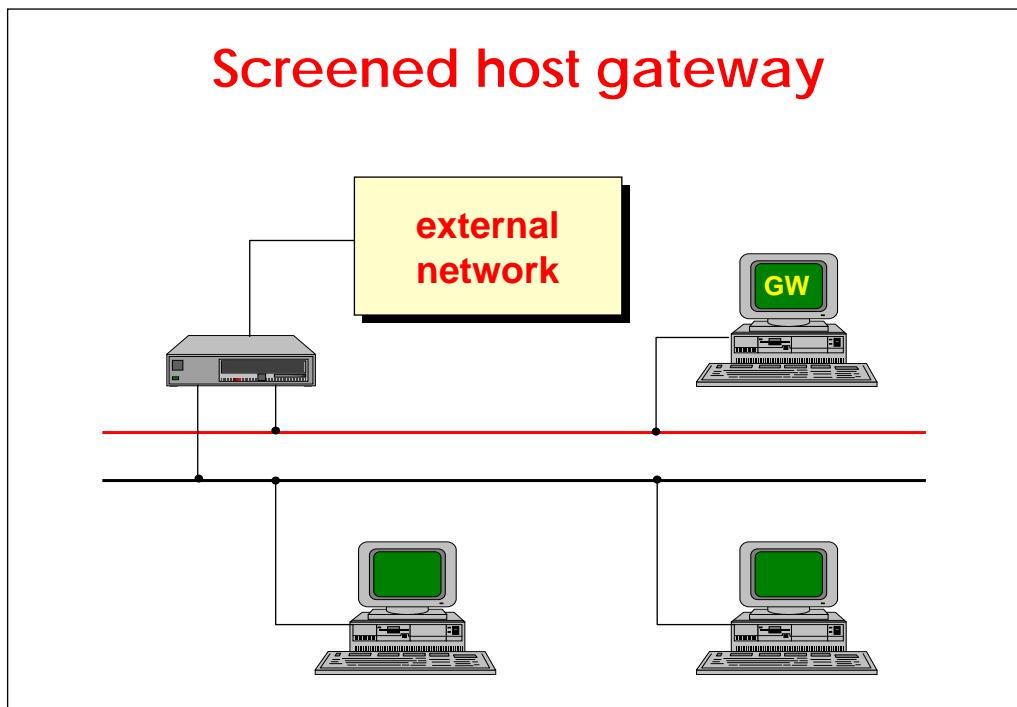




- ### Screening router
- the router is used to filter the traffic (mainly at IP level, partly at transport level – e.g. port numbers)
  - dedicated hardware not needed
  - proxies not needed (hence no change to applications is required)
  - easy and cheap ... but not very strong!

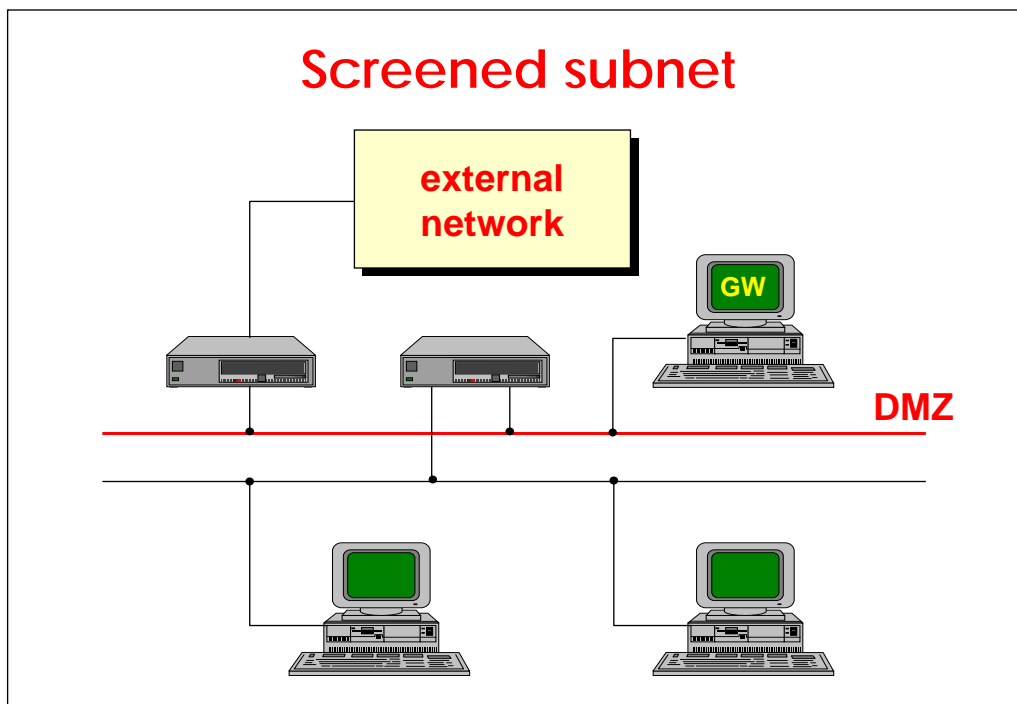


- ### Dual-homed gateway
- easy to be built
  - few additional hardware
  - possible to mask the internal network
  - not flexible
  - big overhead



## Screened-host gateway

- **more expensive**
- **more flexible:**
  - selection of hosts/services to be more thoroughly controlled
- **more complex to manage:**
  - two systems rather than one
- **masking possible only for those hosts/protocols that cross the bastion (but if the router has NAT functionality)**

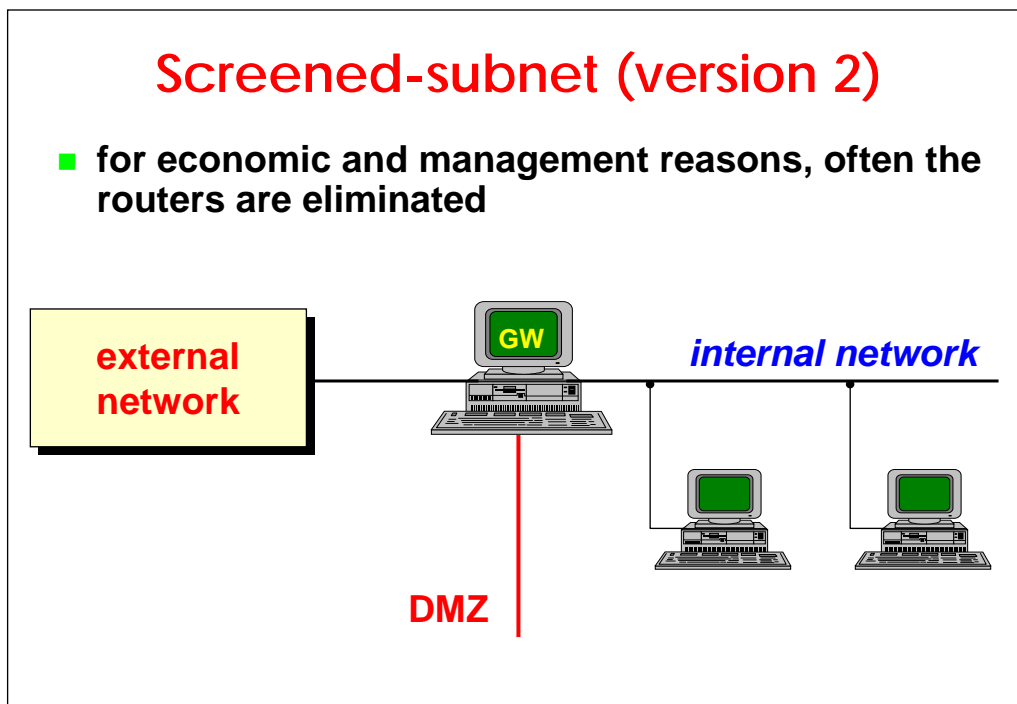


## Screened subnet

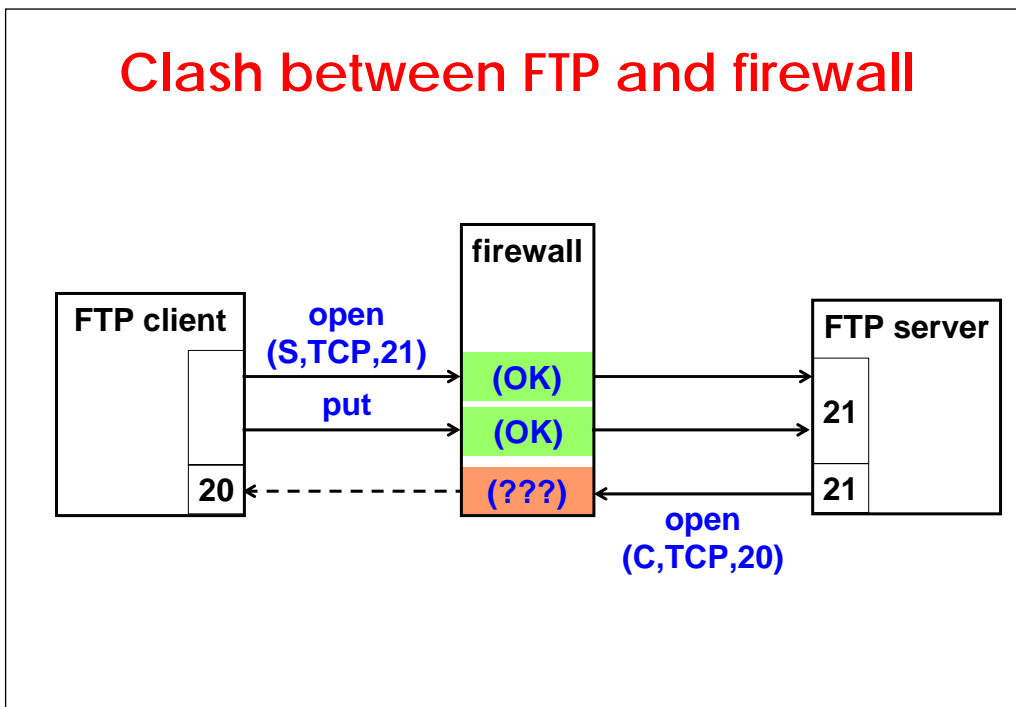
- DMZ (De-Militarized Zone)
- besides the gateway, the external network hosts other machines (typically the public servers):
  - Web
  - remote access
  - . . .
- routing policies can be configured so that the internal network be unknown
- expensive solution

## Screened-subnet (version 2)

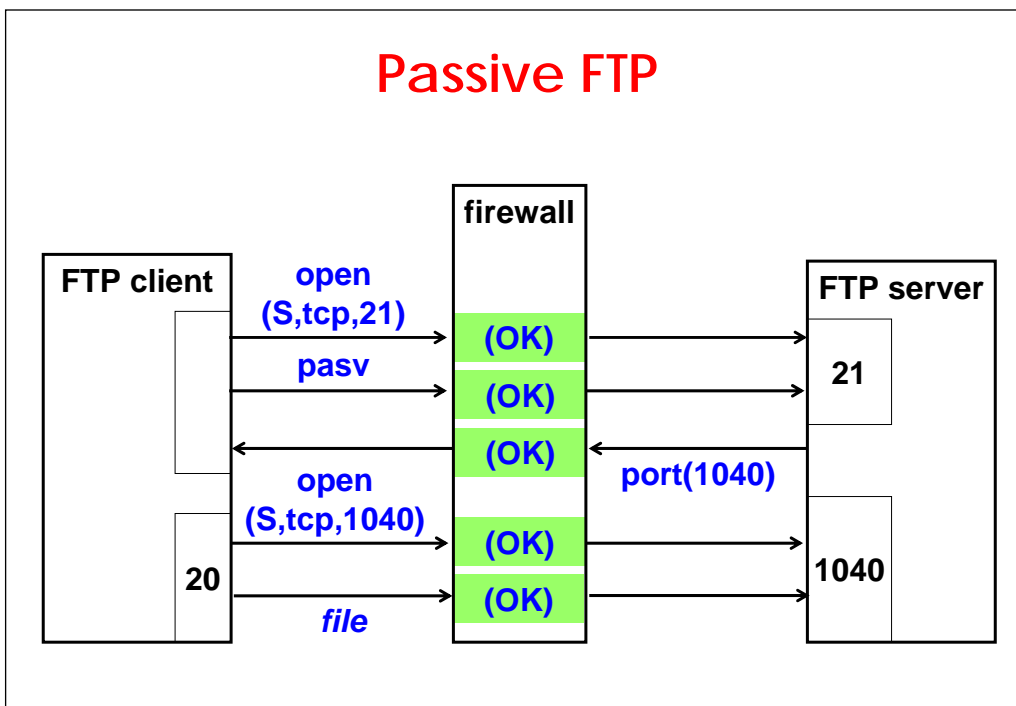
- for economic and management reasons, often the routers are eliminated



## Clash between FTP and firewall

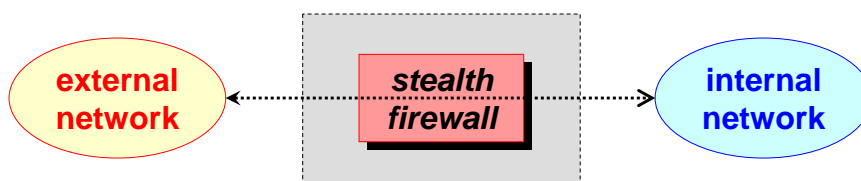


## Passive FTP



## Stealth firewall

- firewall with no network address, so that it cannot be directly attacked
- packets are intercepted physically, by setting the NIC in promiscuous mode



## Intrusion Detection System (IDS)

- **definition:**
  - a system to identify individuals that use a computer or a network without authorization
  - extended also to identify authorized individuals that violate their permissions
- **hypothesis:**
  - the behaviour “pattern” of non authorized individuals is different from that of the authorized ones

## IDS: functional classification

- **passive IDS:**
  - use a cryptographic checksum (e.g. tripwire)
  - pattern recognition (“attack signature”)
- **active IDS:**
  - “learning” = static analysis of system behaviour
  - “monitoring” = active analysis of traffic, sequences, actions
  - “reaction” = comparison with statistical parameters (reaction activated after a threshold)

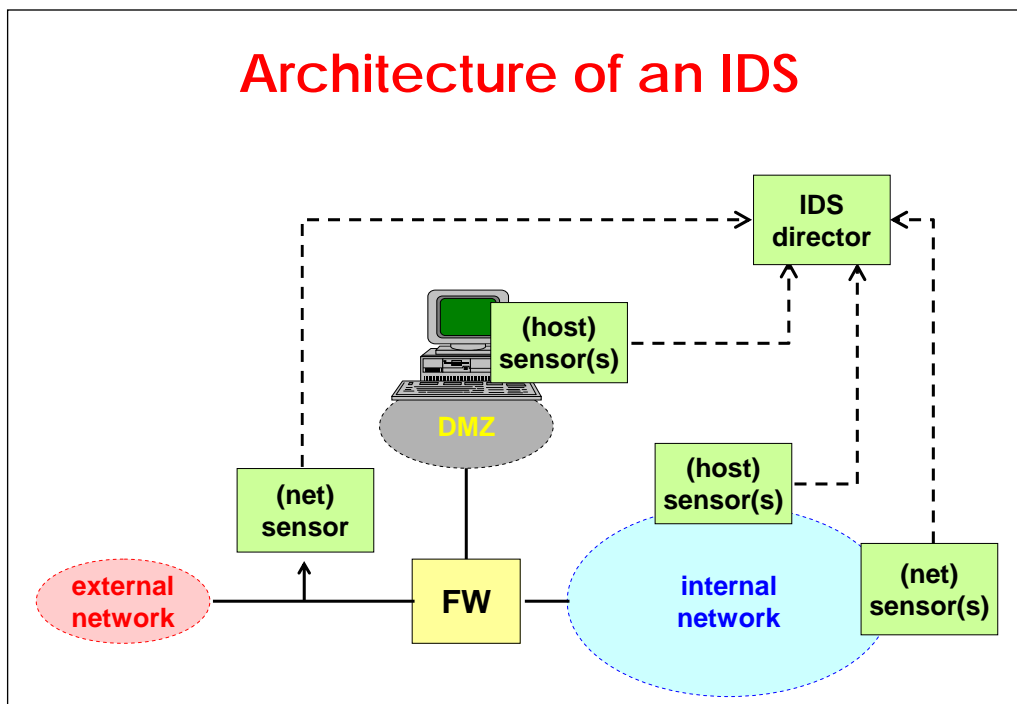
## IDS: topological classification

- **HIDS (host-based IDS)**
  - log analysis (O.S. or application logs)
  - monitoring tools (internal to the O.S.)
- **NIDS (network-based IDS)**
  - network traffic monitoring tools
  - often integrated into routers/switches

## Components of a NIDS

- **sensor**
  - traffic and log control to identify specific patterns
  - creates the appropriate security events
  - can interact with the system (ACLs, TCP reset, ... )
- **director**
  - sensor coordination
  - security database management
- **IDS message system**
  - for secure and reliable communication between the components of the IDS

## Architecture of an IDS



## Channel protection

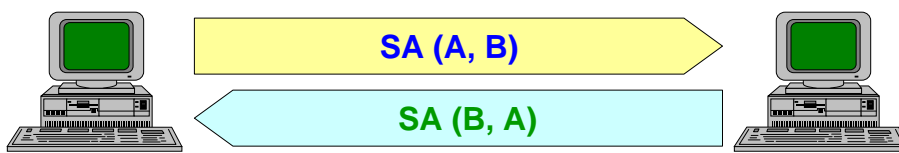
- **level 1 (physical) – only for military environments**
- **level 2 (data link) – only for point-to-point links**
- **level 3 (network):**
  - first general end-to-end level
  - transparent to applications (no modification needed but no control given)
- **level 4 (transport) or level 5 (session):**
  - secure logic channel
  - semi-transparent to applications (some modification needed but full control given)

## IPsec

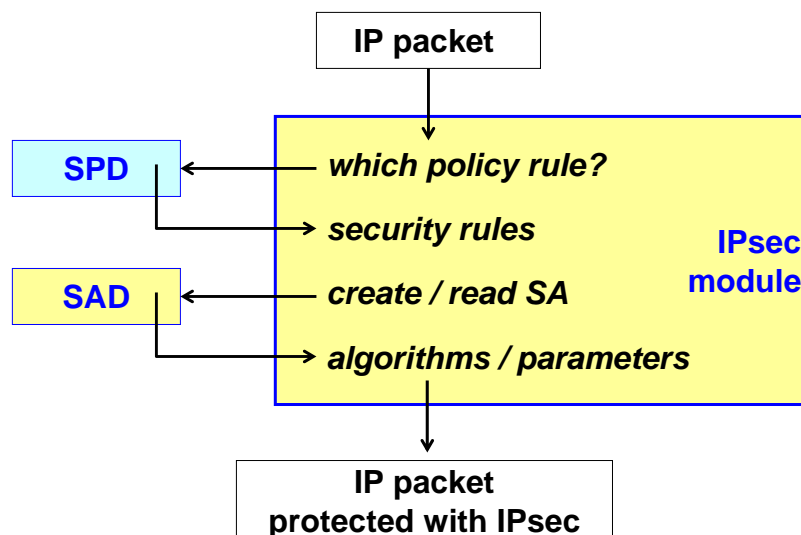
- **IETF network-level security architecture for both IPv4 and IPv6**
- **basic applications: secure VPN, secure end-to-end channel, secure remote access**
- **defines two formats (also called protocols):**
  - AH (Authentication Header)  
integrity, authentication, no replay
  - ESP (Encapsulating Security Payload)  
confidentiality (+AH)
- **key exchange protocol:**
  - IKE (Internet Key Exchange)

## IPsec Security Association (SA)

- IPsec logical “secure connection”
- two SAs needed for full protection of a bidirectional communication between two IP systems

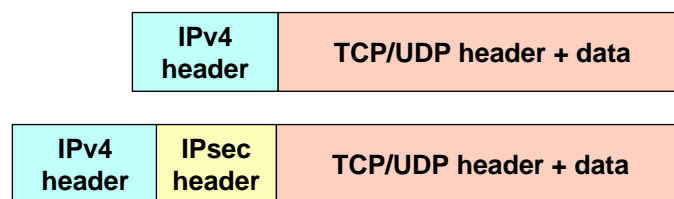


## How IPsec works (output)



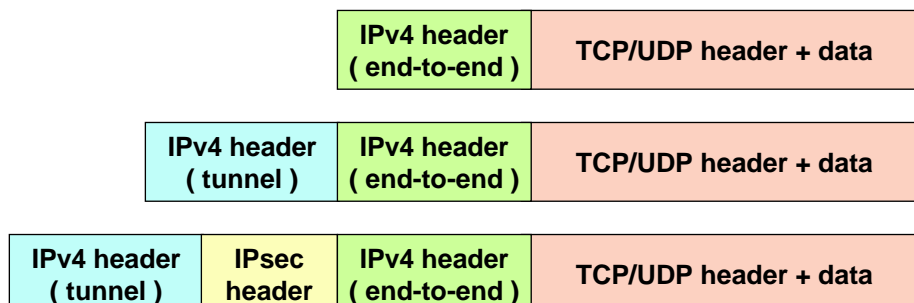
## IPsec in transport mode

- used by hosts, not gateways (exception: traffic for the gateway itself - SNMP, ICMP)
- light but it does not protect the variable IP header fields

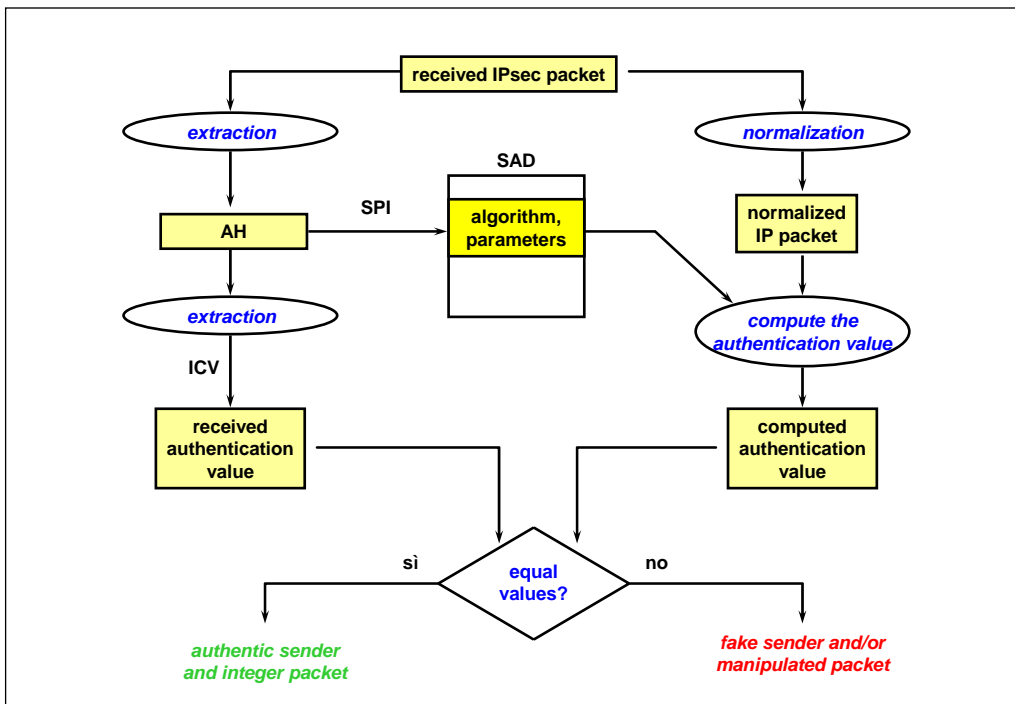
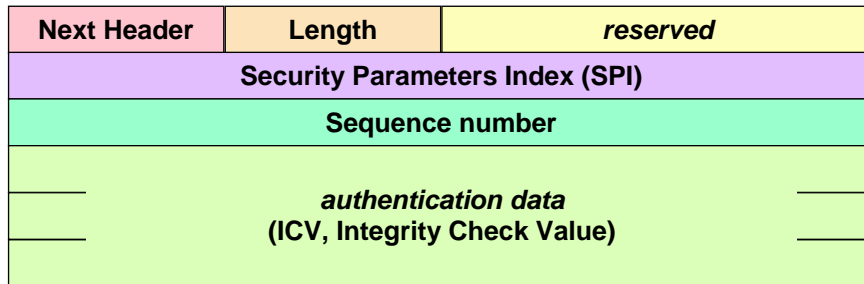


## IPsec in tunnel mode

- typically used by gateways
- heavy but it protects the variable fields of the IP header



## AH - format (RFC-2402)

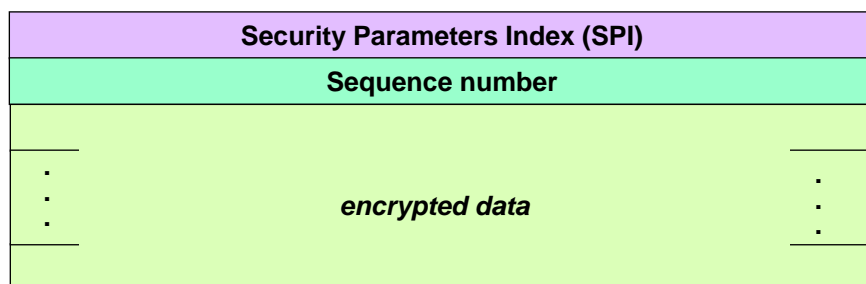


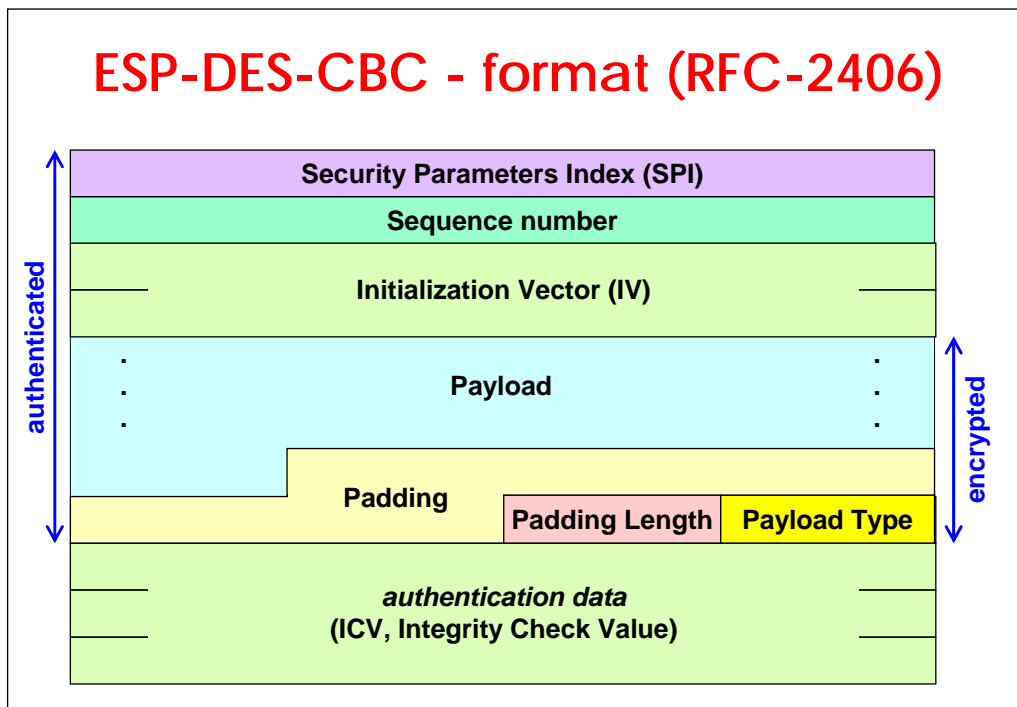
## HMAC-MD5-96

- given  $M$  normalize it to generate  $M'$
- pad  $M'$  to 128 bit (by adding zero-valued bytes) to generate  $M'p$
- pad the key  $K$  to 128 bit (by adding zero-valued bytes) to generate  $Kp$
- given  $ip = 00110110$  and  $op = 01011010$  (repeated up to 128 bit) compute the authentication base:  

$$B = \text{md5} ( (Kp \oplus op) \parallel \text{md5} ( (Kp \oplus ip) \parallel M'p ) )$$
- ICV = 96 leftmost bit di  $B$

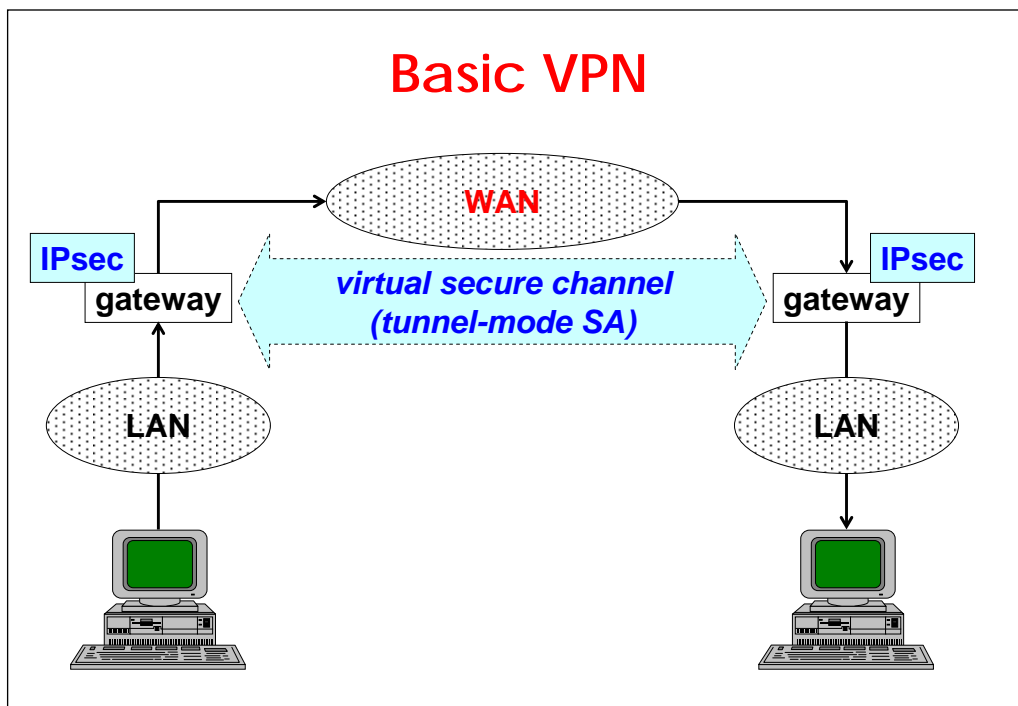
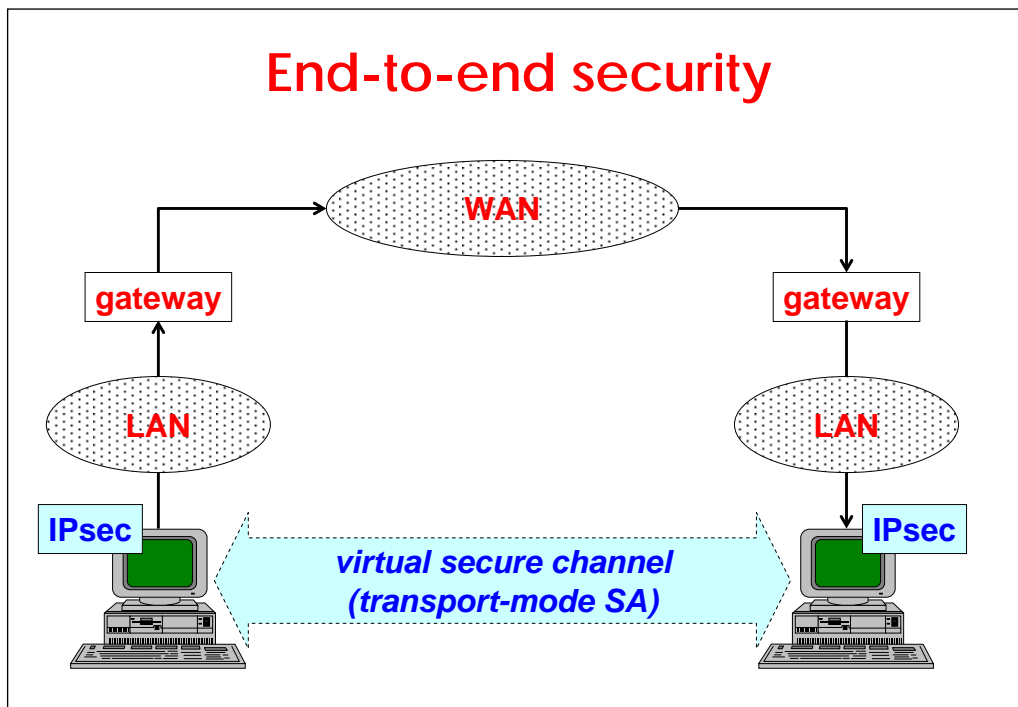
## ESP - format (RFC-2406)

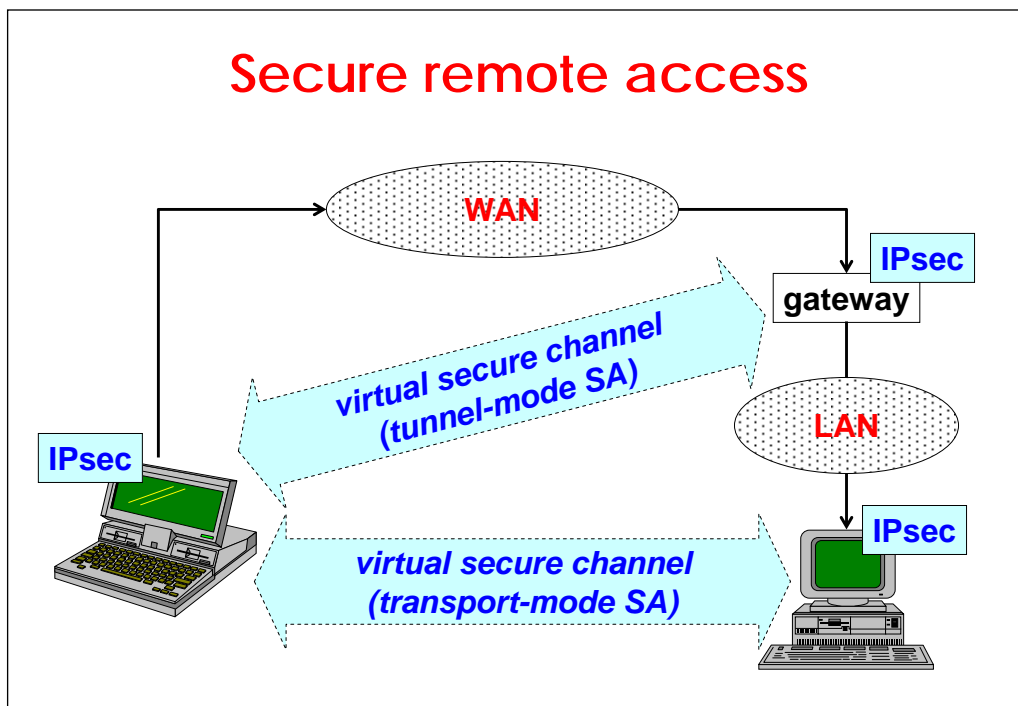
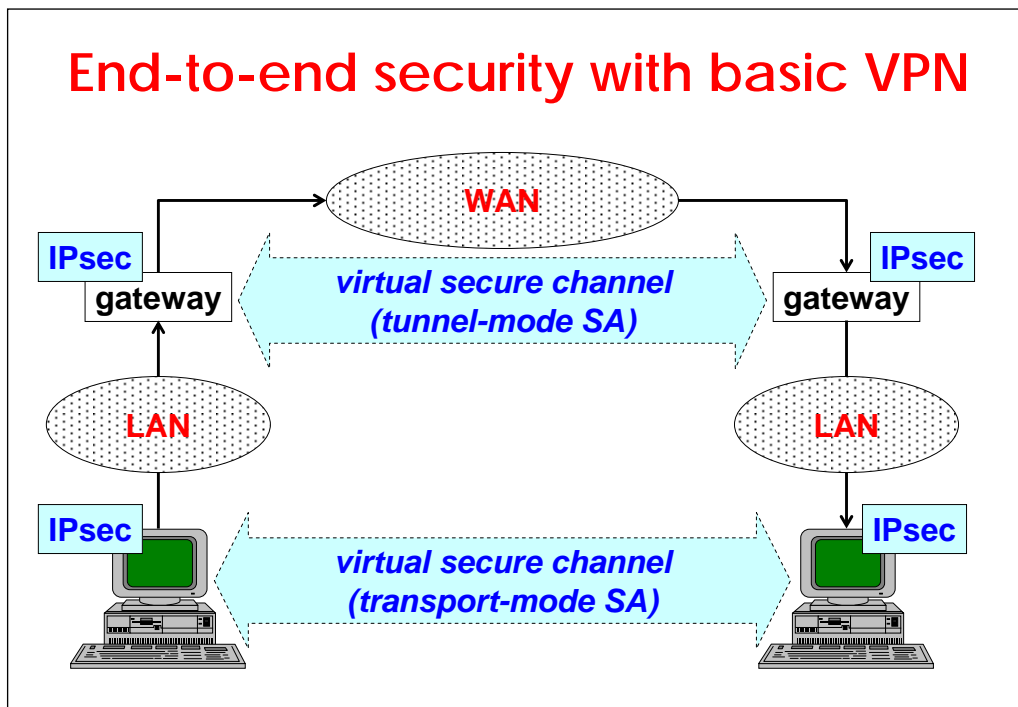




## Some implementation details

- **sequence number:**
  - no need for strictly sequential numbers (protection from replay only)
  - minimum window of 32 packets (64 suggested)
- **NULL algorithm:**
  - for authentication
  - for encryption (RFC-2410)
  - offers a protection vs. performance trade-off

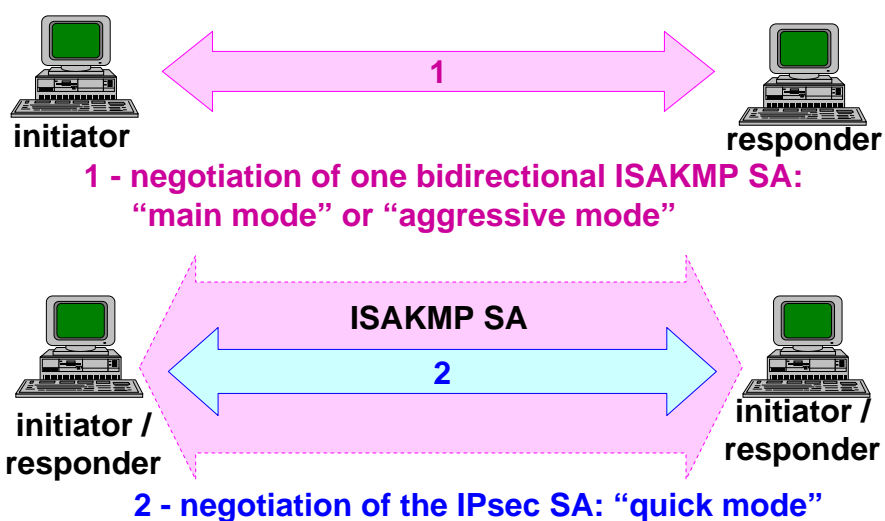




## IPsec key management

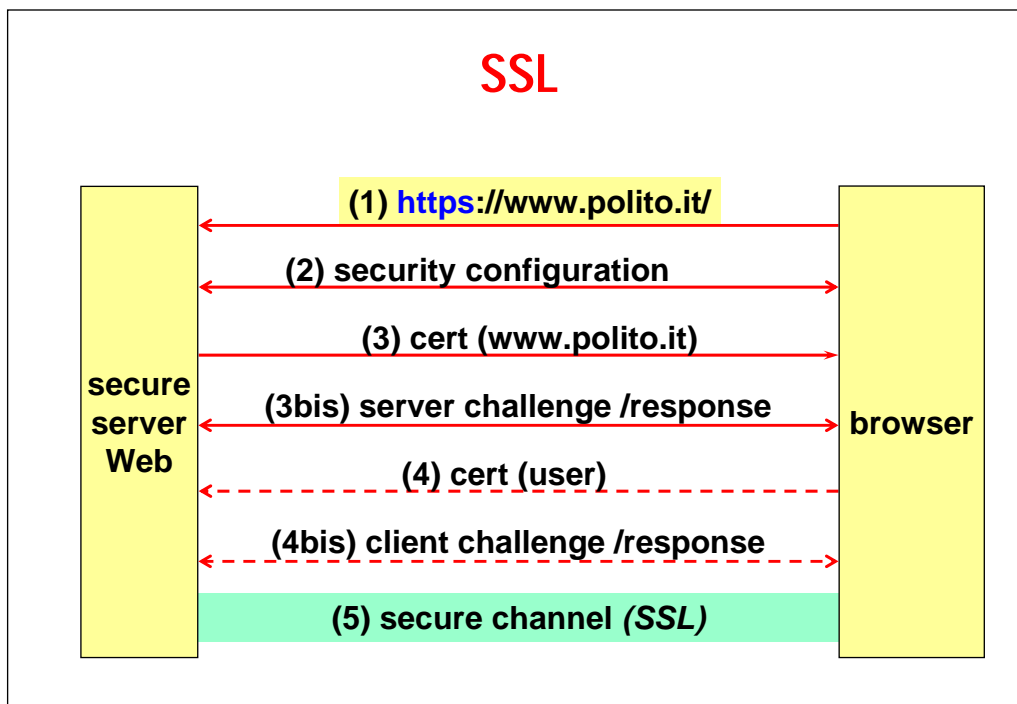
- to provide to IPsec parties:
  - peer authentication (when establishing the SA)
  - symmetric keys (for packet authentication and/or encryption)
- **ISAKMP (Internet Security Association and Key Management Protocol, RFC-2408)**
  - to negotiate, establish, modify and delete the SA
- **OAKLEY (RFC-2412)**
  - key determination between authenticated parties
- **IKE (Internet Key Exchange, RFC-2409)**
  - = ISAKMP + OAKLEY

## IKE in action



## SSL (Secure Socket Layer)

- proposed by Netscape
- secure session layer on top of a reliable transport protocol:
  - data confidentiality (symmetric cryptography)
  - data integrity and authentication (keyed-digest)
  - peer authentication = server or server+client
  - protection from data replay and data filtering
- several applications (HTTPS, LDAPS, FTPS, SMTPS, TELNETS, ...)
- standardized as TLS (RFC-2246)



## Session-id

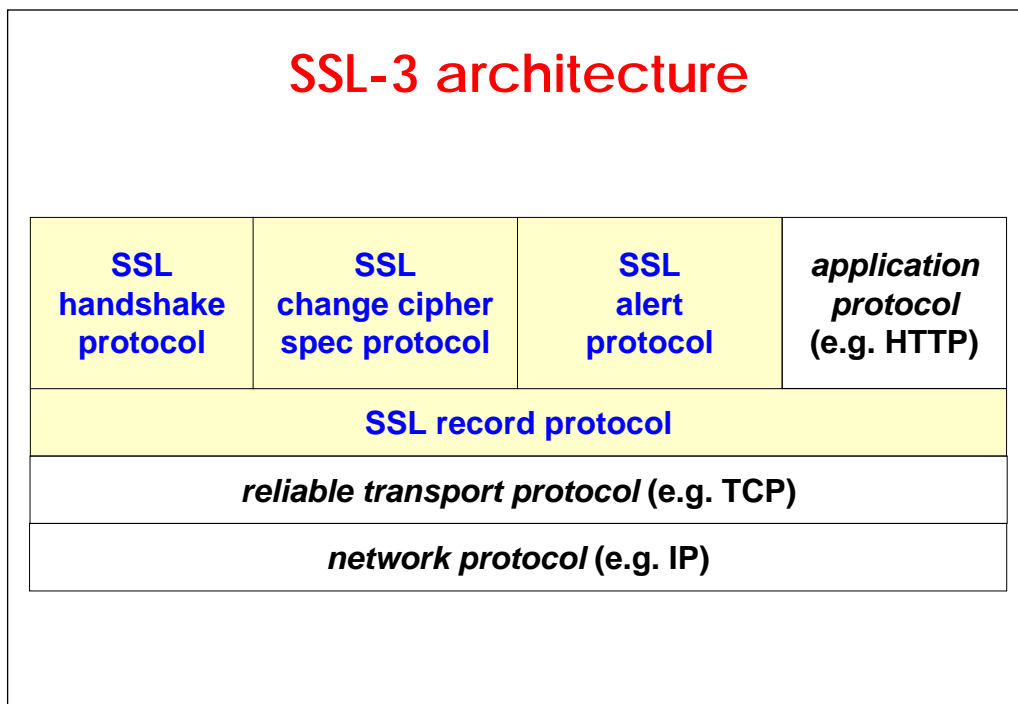
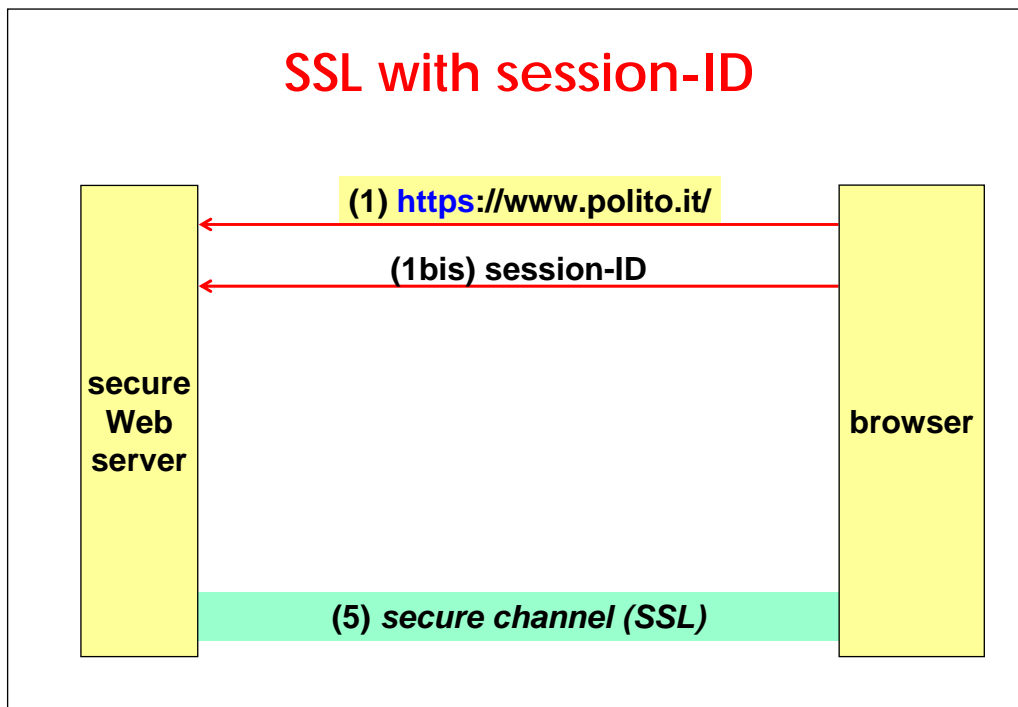
### Typical web transaction:

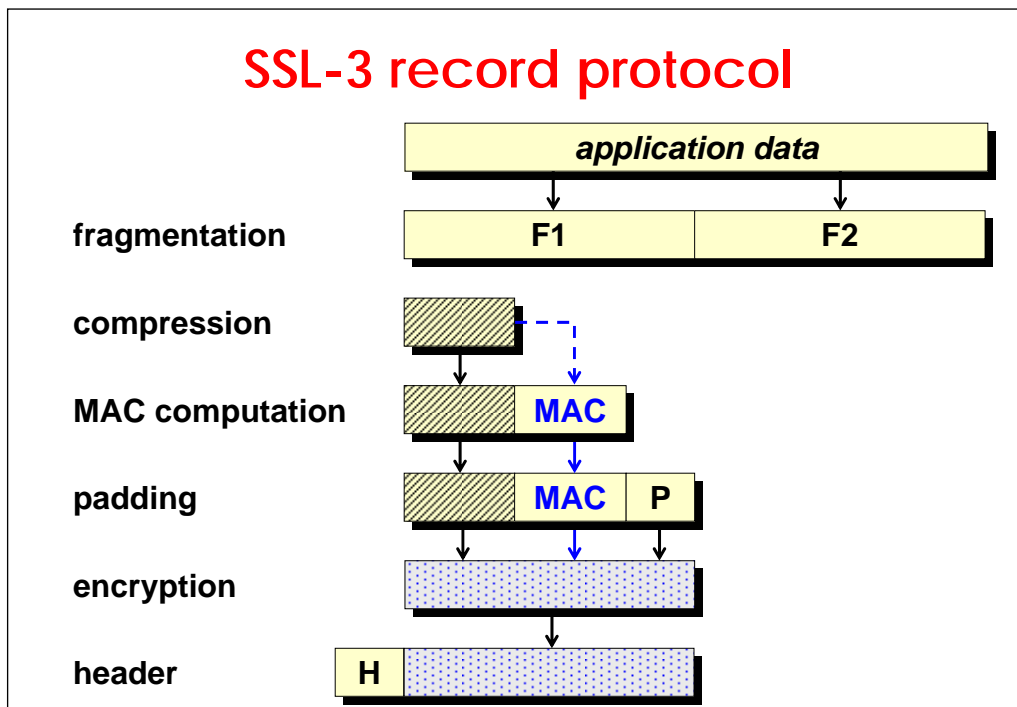
- 1. open, 2. GET page.htm, 3. page.htm, 4. close
- 1. open, 2. GET home.gif, 3. home.gif, 4. close
- 1. open, 2. GET logo.gif, 3. logo.gif, 4. close
- 1. open, 2. GET back.jpg, 3. back.jpg, 4. close
- 1. open, 2. GET music.mid, 3. music.mid, 4. close

Too much overhead if the cryptographic parameters are re-negotiated every time.

## Session-id

- to avoid the re-negotiation for each new connection, the SSL server provides a session identifier
- if the client provides a valid *session-id* the negotiation phase is skipped and the communication proceeds on a secure SSL channel
- the server can refuse to re-use a session identifier (always, or after a timeout = the session lifetime)





## SSL-3: novelties with respect to SSL-2

- **data compression:**
  - optional
  - before encryption (afterwards it is useless ...)
- **optional data encryption:**
  - to provide just authentication and integrity
- **re-negotiation of the connection:**
  - periodic change of the keys
  - change of the algorithms

## HTTP security

- **HTTP/1.0 security mechanisms:**
  - “address-based” = server access controlled based on the client’s IP address
  - “password-based” (or Basic Authentication Scheme) = server access controlled based on username and password, Base64 encoded
- **both schemas are highly insecure (because HTTP assumes an underlying secure transport channel!)**
- **HTTP/1.1 introduces “digest authentication” based on a symmetric challenge-response**
- **RFC-2617 “HTTP authentication: basic and digest access authentication”**

## HTTP - basic authentication scheme

```
GET /path/to/protected/page
HTTP/1.0 401 Unauthorized - authentication failed
WWW-Authenticate: Basic realm="RealmName"
Authorization: Basic B64_encoded_username_password
HTTP/1.0 200 OK
Server: NCSA/1.3
MIME-version: 1.0
Content-type: text/html
<HTML> protected page ... </HTML>
```

## WWW security

- **SSL channel:**
  - protects transactions
  - protects application passwords
- **password (for Basic Authentication):**
  - of the HTTP service
  - of the hosting O.S. (e.g.. Windows domain or Unix)
- **ACL for access control:**
  - according to the authentication type and objects to be protected (O.S. users, file system protections, ...)

## SSL client authentication for application-level protection

- **SSL client authentication identifies the user that opened the channel (with no need for username and password)**
- **some web servers support a (semi-)automatic mapping from credentials (extracted from the X.509 client certificate) and users (of the web server and/or O.S.)**

## SSH (Secure SHell)

- a complete substitute for R-commands (rsh, rlogin, rcp) + X11 forwarding
- channel encryption
- various types of authentication:
  - by IP address (= rlogin + crypto)
  - by asymmetric challenge, with host-based keys
  - by asymmetric challenge, with user-based keys
- protocol not yet published as RFC

## Electronic payment systems

- failure of e-currency due to technical and political problems (e.g. DigiCash bankrupt)
- currently the most common payment method is by transmitting a credit card number on a SSL channel ...
- ... which is not a complete fraud protection: VISA Europe declares that 50% of the fraud attempts originate from Internet transactions, although they are only 2% of its total transactions!

## Security of Internet payments based on credit card

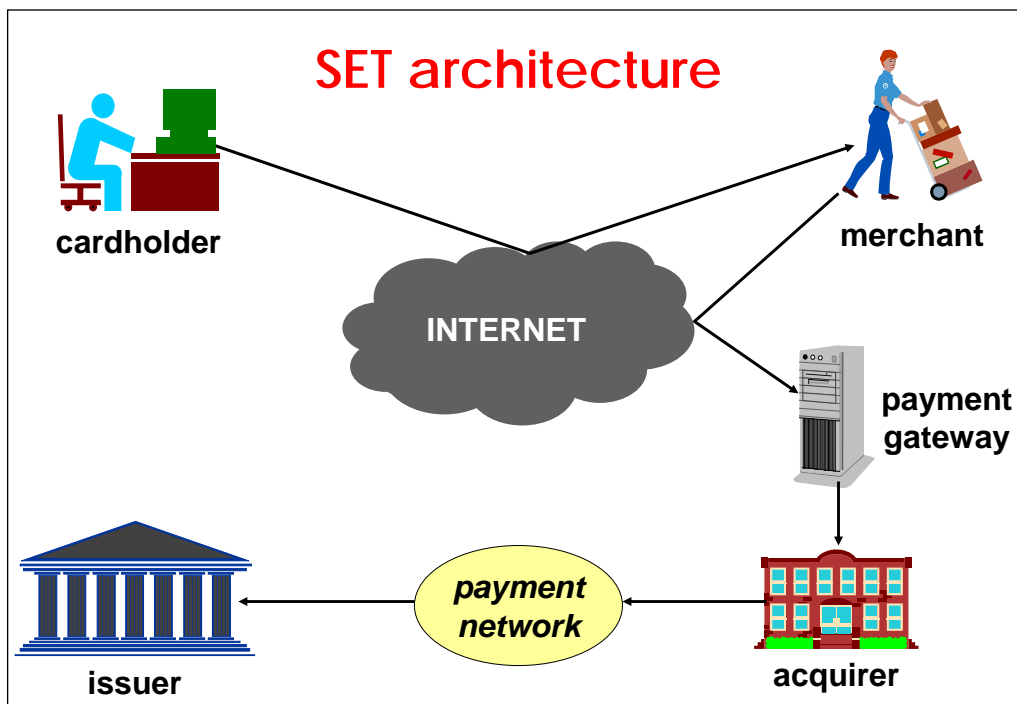
- **STT**  
(Secure Transaction Technology)  
VISA + Microsoft
- **SEPP**  
(Secure Electronic Payment Protocol)  
Mastercard, IBM, Netscape, GTE, CyberCash
- **SET = STT + SEPP**  
(Secure Electronic Transaction)

## SET

- SET is not a payment system but a set of protocols to securely operate in an open unprotected network environment an existing credit card payment infrastructure
- uses X.509v3 certificates with private SET-only extensions
- guarantees the user privacy because every actor comes to know only the data relevant for his side

## SET technical characteristics

- version 1.0 (may 1997)
- digest: SHA-1
- symmetric encryption: DES
- key exchange: RSA
- digital signature: RSA with SHA-1
  
- [www.setco.org](http://www.setco.org)



## SET double signature

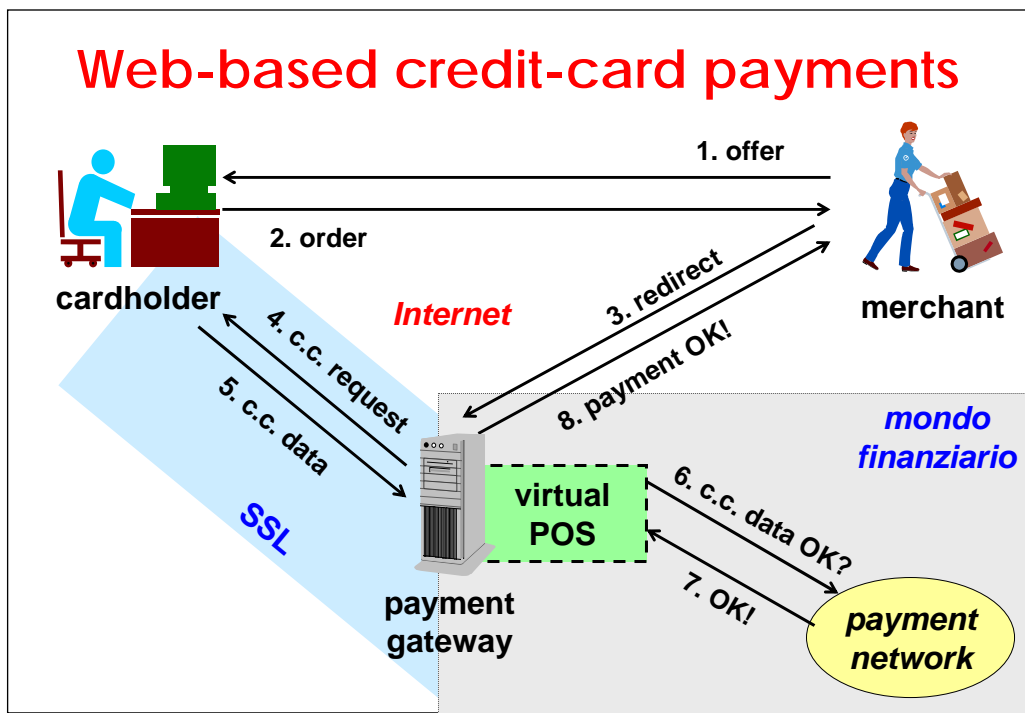
- to guarantee user privacy with respect to both the merchant and the payment system (acquirer + issuer) SET uses **double digital signature**
- the merchant has no knowledge of the payment details
- the bank has no knowledge of the items bought
- only the user can prove the association between the payment and the merchandise

## SET double signature: details

- PI: Purchase Information (payment)
- OI: Order Information (merchandise)
- $DS = E ( H( H(PI),H(OI) ), U_{kpri} )$
- DS+H(PI) to the merchant
- merchant knows OI and can compute  $(H(PI),H(OI))$  to verify its correspondence to the value extracted from the signature
- DS+H(OI) to the acquirer
- acquirer knows PI and can compute  $H(H(PI),H(OI))$  to verify its correspondence to the value extracted from the signature

## SET problems

- software very expensive (for the CA, the merchant and the acquirer)
- need for a special client-side application (SET wallet)
- complex procedure to issue the SET public-key certificates to the users
- version 2.0 should have been able to operate without the wallet (by using a browser and a SET wallet server)



## Web-based credit-card payments

### ■ foundations:

- the buyer holds a credit card
- the buyer has a SSL browser

### ■ outcomes:

- the effective security depends on the configuration of both the server and the client
- the payment gateway has all the information (payment + merchandise) while the merchant knows only the merchandise information

## Secure e-mail: S/MIME

- uses X.509v3 certificates for authentication, integrity and confidentiality

### *signed*

text
Excel table
Word doc
<b>digital signature in S/MIME format</b>

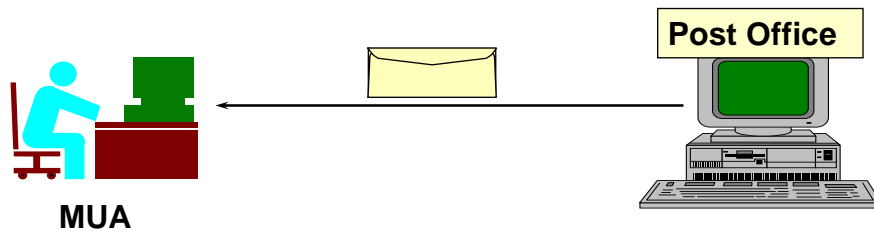
### *encrypted*

text
Excel table
Word doc
<b>encrypted envelope in S/MIME format</b>

### *signed and encrypted*

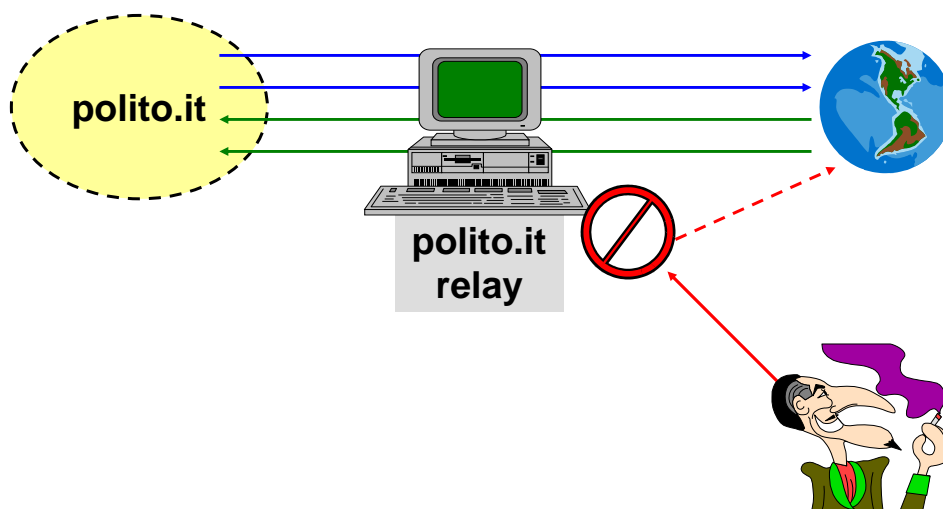
text
Excel table
Word doc
<b>digital signature in S/MIME format</b>
<b>encrypted envelope in S/MIME format</b>

## Secure e-mail download



- IMAP or POP over SSL/TLS
- IMAPS or POPS provides:
  - user authentication
  - server authentication
  - mail confidentiality/integrity during transfer

## Mail relay



## Anti-spamming measures

- **filter on the MUA IP address**
  - problem: mobile users
  - problem: IP spoofing
- **filter on “From” field**
  - problem: fake mail
- **use SMTP authentication**
  - problem: yet another password
- **SMTPS (=SMTP over SSL) with client authentication**
  - very strong solution