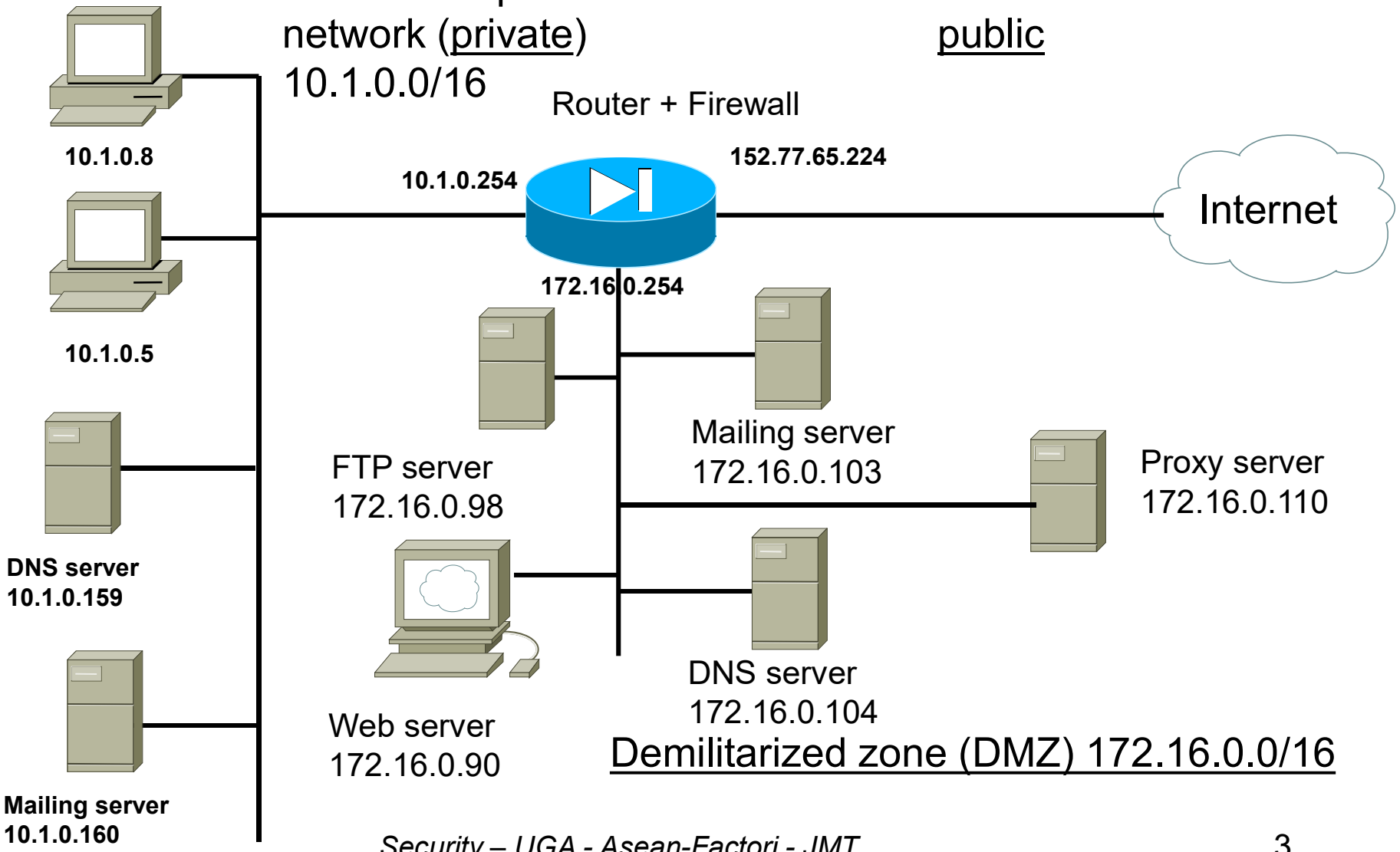
4.2. Cyber-security 2

Jean-Marc THIRIET

jean-marc.thiriet@univ-grenoble-alpes.fr

Security of the infrastructure DMZ, firewall

A network with a firewall/router…



Operation: inspection of each packet

- Source Address
- Destination Address
- Ports

- The decision to authorize or not depends on each inspected point
- Note: fast data processing
- Example of standard ACL on a Cisco router
  - To authorize the packets (permit)
  - To prohibit the packets (deny)

```
access-list 10 permit any 192.168.10.0
access-list 10 permit any 192.168.20.0
access-list 10 deny any 192.168.30.0
```

Stateless firewall: Filtering of packets by means of ACL (Access Control Lists)

- TCP/IP Data segmented in packets
  - Layer 3 of the TCP/IP model
- Examination of the contents of the packets and application of certain rules
  - Transmission of the packet
  - Removal of the packet
- Very widespread technology at the beginning of Internet
  - First line of defense
- Very much still used in the routers
- First line of defense, combined with other firewalls technologies

Examples of firewall lists (stateless)

	Source Address	Source Port	Destination Address	Destination Port	Action	Description
1	Any	Any	192.168.1.0	> 1023	Allow	Rule to allow return TCP Connections to internal subnet
2	192.168.1.1	Any	Any	Any	Deny	Prevent Firewall system itself from directly connecting to anything
3	Any	Any	192.168.1.1	Any	Deny	Prevent External users from directly accessing the Firewall system.
4	192.168.1.0	Any	Any	Any	Allow	Internal Users can access External servers
5	Any	Any	192.168.1.2	SMTP	Allow	Allow External Users to send email in
6	Any	Any	192.168.1.3	HTTP	Allow	Allow External Users to access WWW server
7	Any	Any	Any	Any	Deny	"Catch-All" Rule - Everything not previously allowed is explicitly denied

Stateful firewall: Dynamic ACL

- Dynamic filtering
  - **Stateful inspection firewall**: packet filters that take into consideration OSI-layer 4 (particularly TCP) => if a connection is authorized, every packet within this exchange will be implicitely accepted
  - Dynamic entries for responses to the TCP, UDP, ICMP requests
  - Does not require to keep open the static ports (the ports remain open only during the time of the session)
- Follow-up/monitoring of the TCP sequence numbers
  - Monitoring of the sequence numbers of the input and output packets to follow-up communication flows
  - Protection against “man in the middle” attacks and session hackings

Dynamic ACL

- Follow-up of specific applications (example of protocols)
  - Cu-SeeMe (port 7648): PTP videoconference
  - FTP (port 21)
  - HTTP (port 80 or 8080)
  - HTTPS (port 453)
  - DNS (Domain Name Server): port 53
  - H.323 (port 1720): multi-media communication (VoIP, video, audio)
  - ICMP: repairing of problems (administrator) + used by the pirates => to let pass only ICMP messages generated inside the network
  - MCGP (Media Control Gateway Protocol, port 2427): VoIP
  - MSRPC (Microsoft Remote Procedure Call Protocol, port 135): communication of inter-systems process

Dynamic ACL

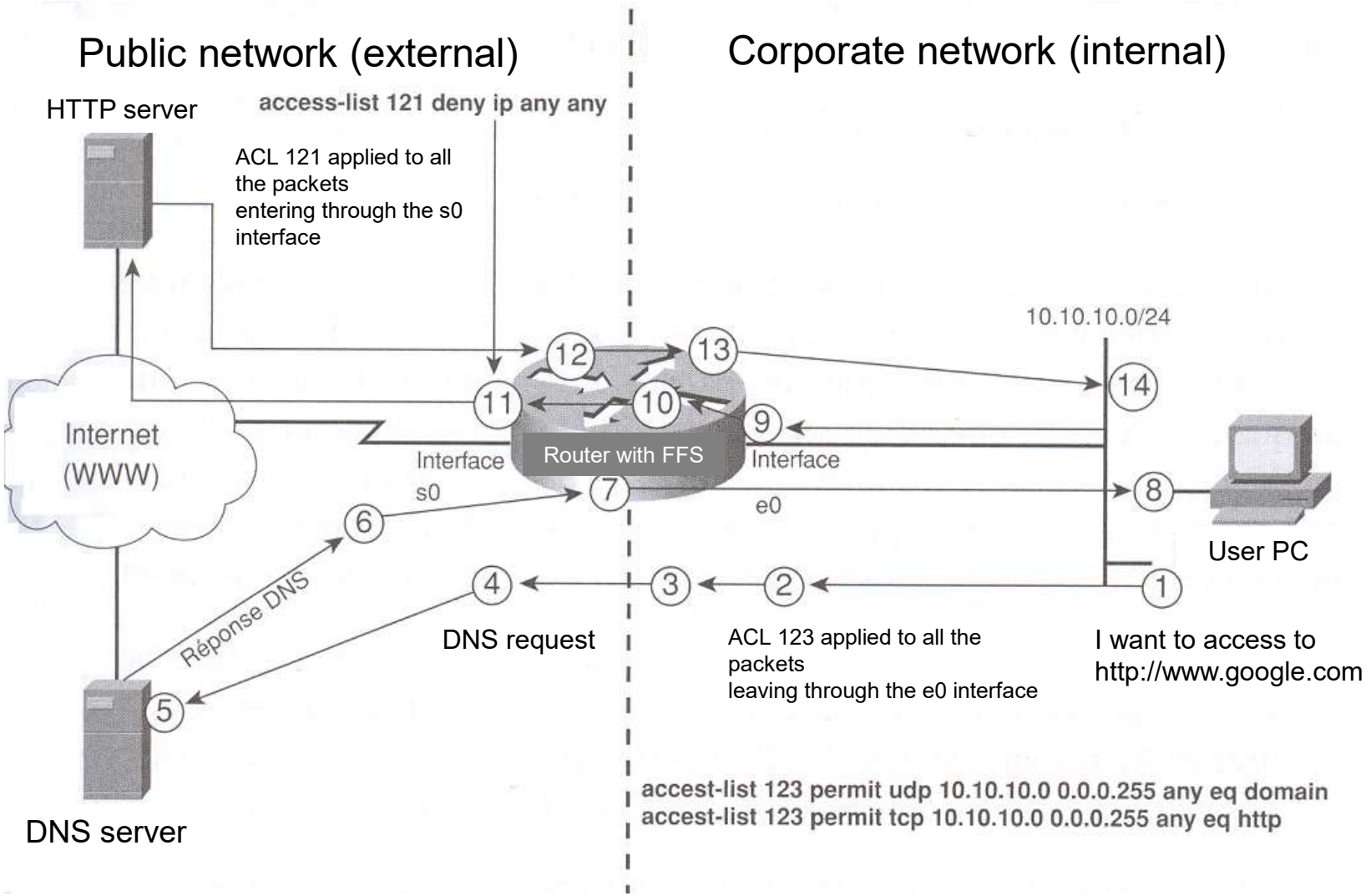
- NetShow (port 1755): Microsoft streaming
- R-EXEC (port 512): distant controls (Unix)
- R-SHELL (port 514): distant Shell (Unix)
- RTSP (Real-Time Streaming Protocol, port 544): streaming and VoIP
- SMTP (Simple Mail Transfer Protocol, port 25): mail
- SQLNet (port 1521): Communications clients-database
- Stream Works (port 1558): Real Networks Streaming
- Audio Real (port 7070): Real Networks Streaming
- TFTP (Trivial File Transfer Protocol, port 69): client-server file transfer
- VDOlive (port 7000): streaming

Example

- ACL
- Access-list 121 Permit TCP/UDP 10.10.10.0/24 any http
- Access-list 121 Permit TCP/UDP 10.10.10.0/24 DNS_SERVER dns

- Access-list 121 Permit TCP/UDP 10.10.10.0/24 any 80
- Access-list 121 Permit TCP/UDP 10.10.10.0/24 DNS_SERVER 53

Example (1/5)


Public network (external)
Corporate network (internal)
HTTP server
access-list 121 deny ip any any
ACL 121 applied to all the packets entering through the s0 interface
10.10.10.0/24
Internet (WWW)
Router with FFS
Interface s0
Interface e0
User PC
Réponse DNS
DNS request
ACL 123 applied to all the packets leaving through the e0 interface
I want to access to http://www.google.com
DNS server
accest-list 123 permit udp 10.10.10.0 0.0.0.255 any eq domain
accest-list 123 permit tcp 10.10.10.0 0.0.0.255 any eq http

Example (2/5)

- 1. The user types www.google.fr
  - The station emits a request for DNS name resolution to obtain the URL IP address
- 2. The DNS request packet (a UDP datagram) arrives on the router Ethernet internal interface
  - It is compared with the list “123” (filtering)
  - It is transmitted if authorized or removed
- 3. The authorized packet is controlled by the CBAC (Context-Based Access Control => contextual access control)
  - Inspection
  - Consignment of information in the table of states
    - source IP Address and port number
    - destination IP Address, port number and protocol
- 4. Creation of a temporary instruction `permit` on list 121
  - Authorization of the responding traffic by the destination host (DNS server)
  - Temporary instruction placed in front of the static instructions in the ACL

Example (3/5)

- 5. The DNS request packet (UDP 53 port) is transmitted to the DNS server
  - Response of the DNS server
  - ACL dynamic input kept during 5 seconds
- 6. Arrival of the DNS response packet
  - Comparison with the ACL n. 121
  - Authorized since it belongs to an established session
- 7. Inspection of the DNS response packet
  - Conservation of information until expiration of the timer (timer for the keeping of UDP sessions)
- 8. Arrival of the DNS response to the user PC and initiation by the PC of an HTTP session with google
  - HTTP is based on TCP, therefore the first packet comprises the SYN (synchronization) bit; this bit is activated to start the three-times negotiation process of TCP

Example (4/5)

- 9. HTTP packet is authorized
  - list 123 is authorizing HTTP port n. 80
- 10. Inspection of the output packet and consignment of information in the table of states
  - Source IP address and port
  - Destination IP address, port and protocol
- 11. Creation of a temporary instruction `permit` on list 121
  - Authorization of the traffic in response by the destination host (HTTP server)
  - Temporary instruction placed in front of the static instructions of the ACL
  - Maintenance of the entry during 30 seconds (time to receive a SYN-ACK packet, synchronization-acknowledgement from the Web server)
- 12. Reception of the packet coming from the Web server
  - Authorized by list 121 (because it belongs to an established session)

Example (5/5)

- 13. Inspection of the packet coming from the Web server
  - Elimination of the packet if there are specific violations of protocols
- In the case of HTTP and other protocols requiring several sessions
  - Continual update of the table of states
  - Continual update of the ACL
- Times of removal of temporary entries in the ACL
  - ICMP and UDP, with expiration of a timer (configurable duration)
  - TCP, five seconds after the exchange of FIN packets

Application firewalls

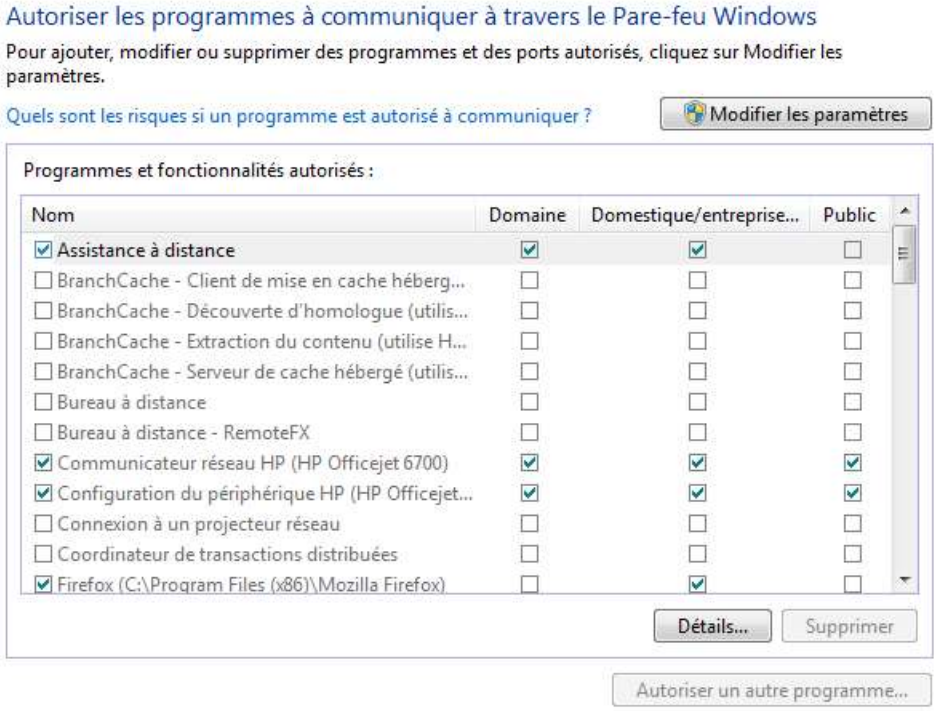
- Last generation of firewall
- Complete conformity of a packet to the expected protocol
- Ex : HTTP protocol only on the TCP port 80
- Need large calculation resources
- Problematics of some protocols not respecting strictly the layer-OSI model (some IP or TCP infos are managed at the application level)

Identifying firewalls

- Identification of connections crossing through the IP filter.
- Filtering rules per user and not only per IP or MAC addresses
- Possibility to monitor the network activity per user
- Dynamic rules based on a user authentication (ex Kerberos), the identity of her/his computer and the level of security (presence of an antivirus, of particular patches)

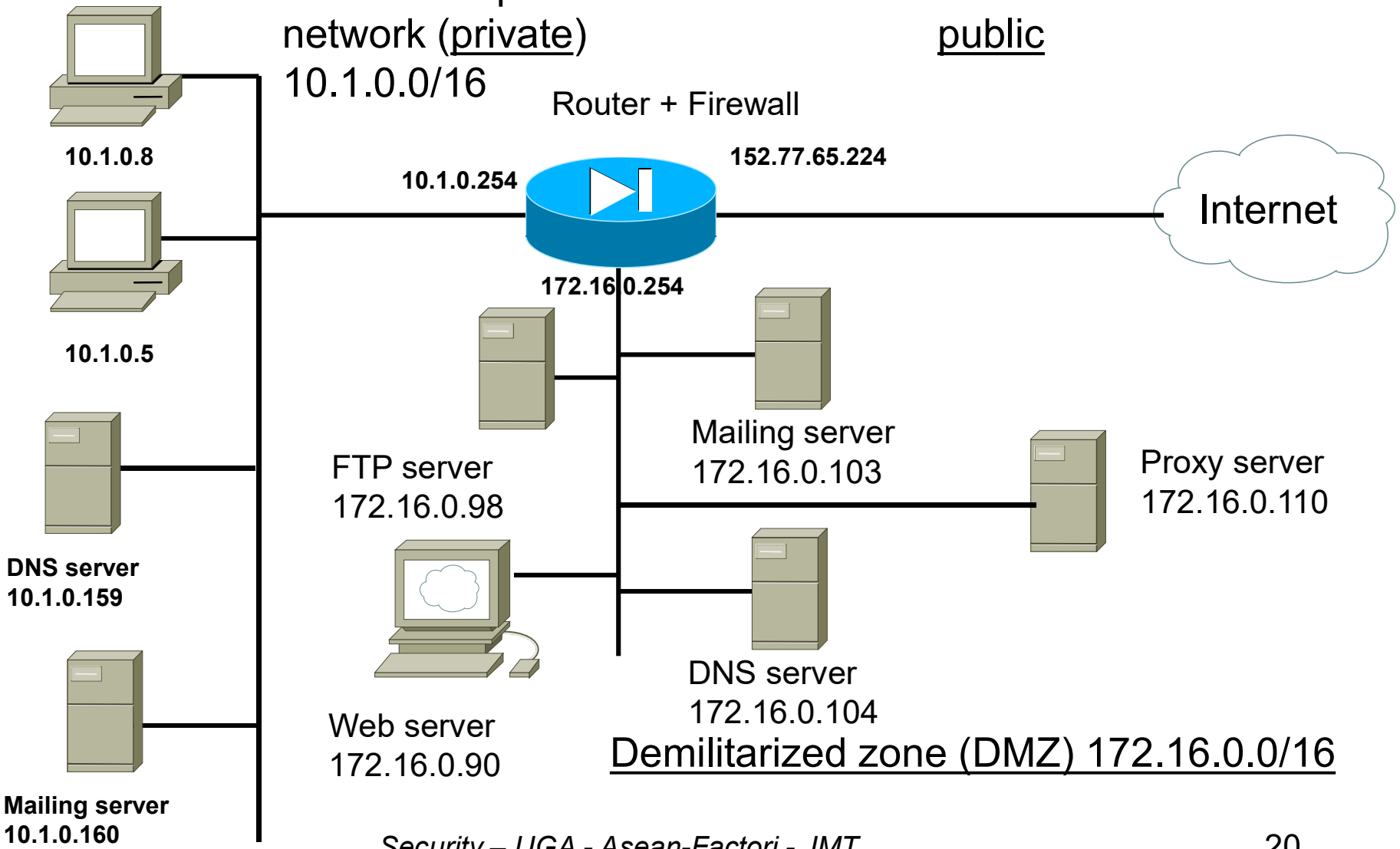
Personal firewalls

- Important element in a strategy of in-depth security
- Personal firewall
  - May be integrated to the OS (Windows, Mac…)
  - Ex of a configuration panel

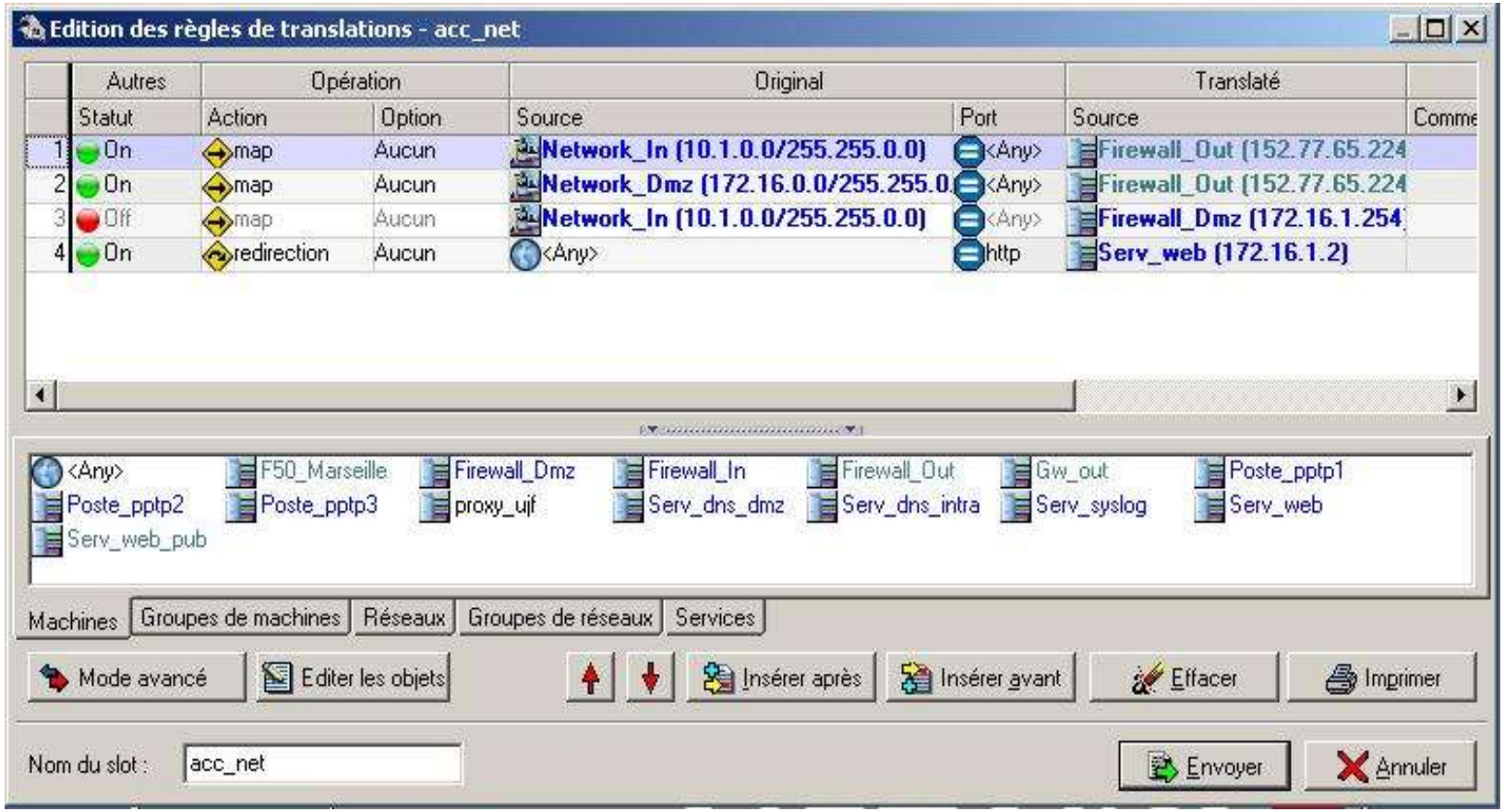


Type of firewalls

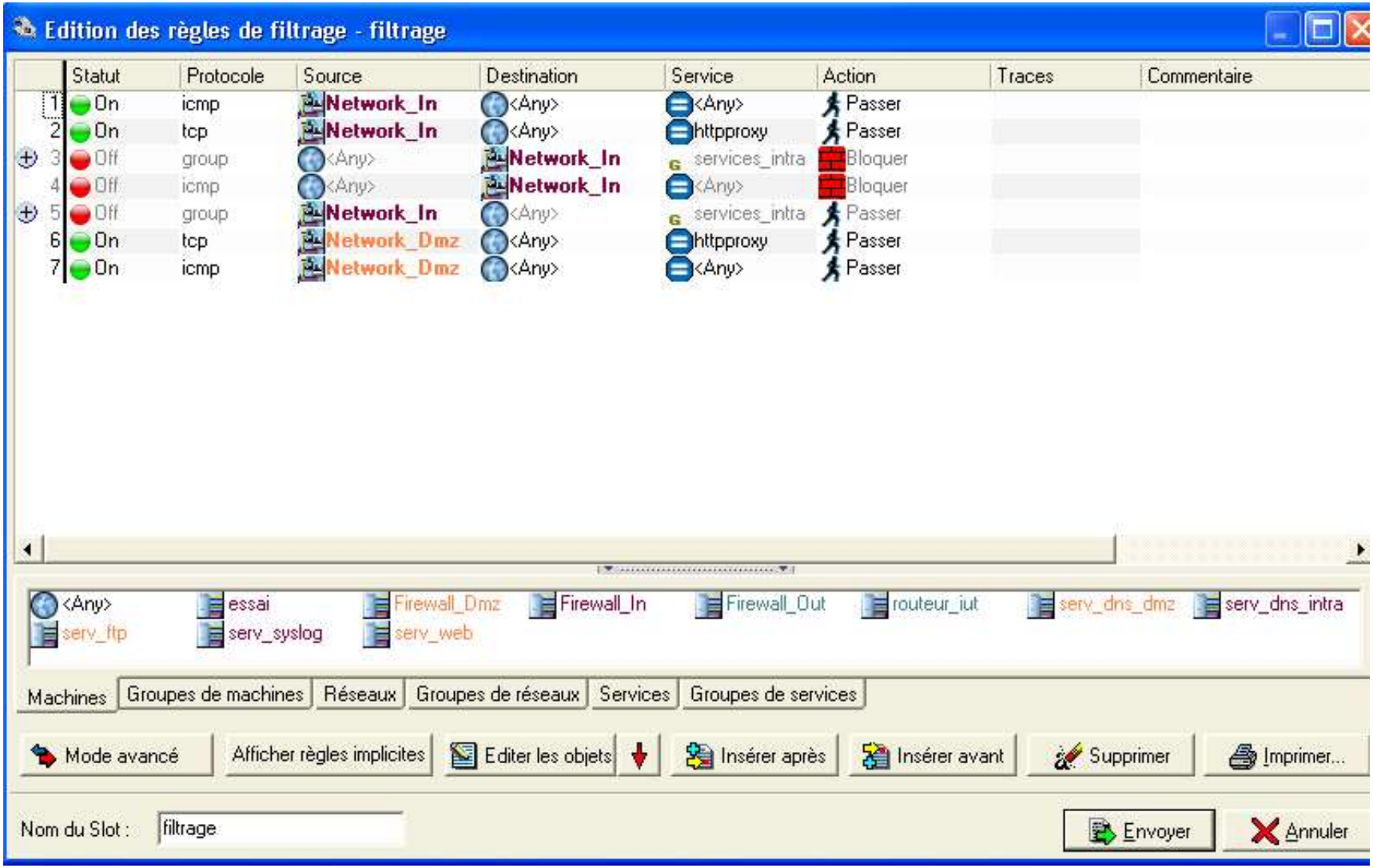
A network with a firewall/router…



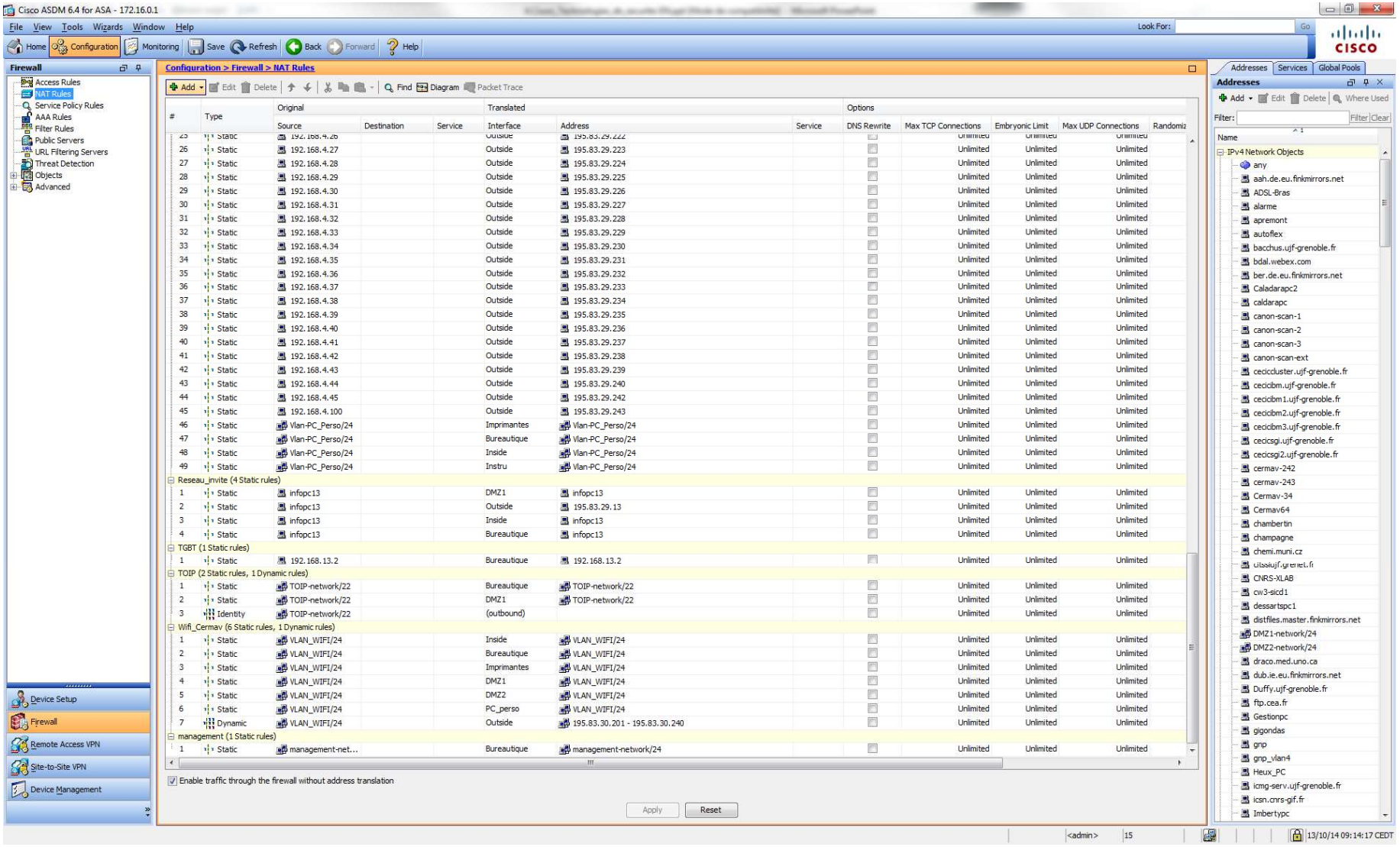
Translation (Netsacq)



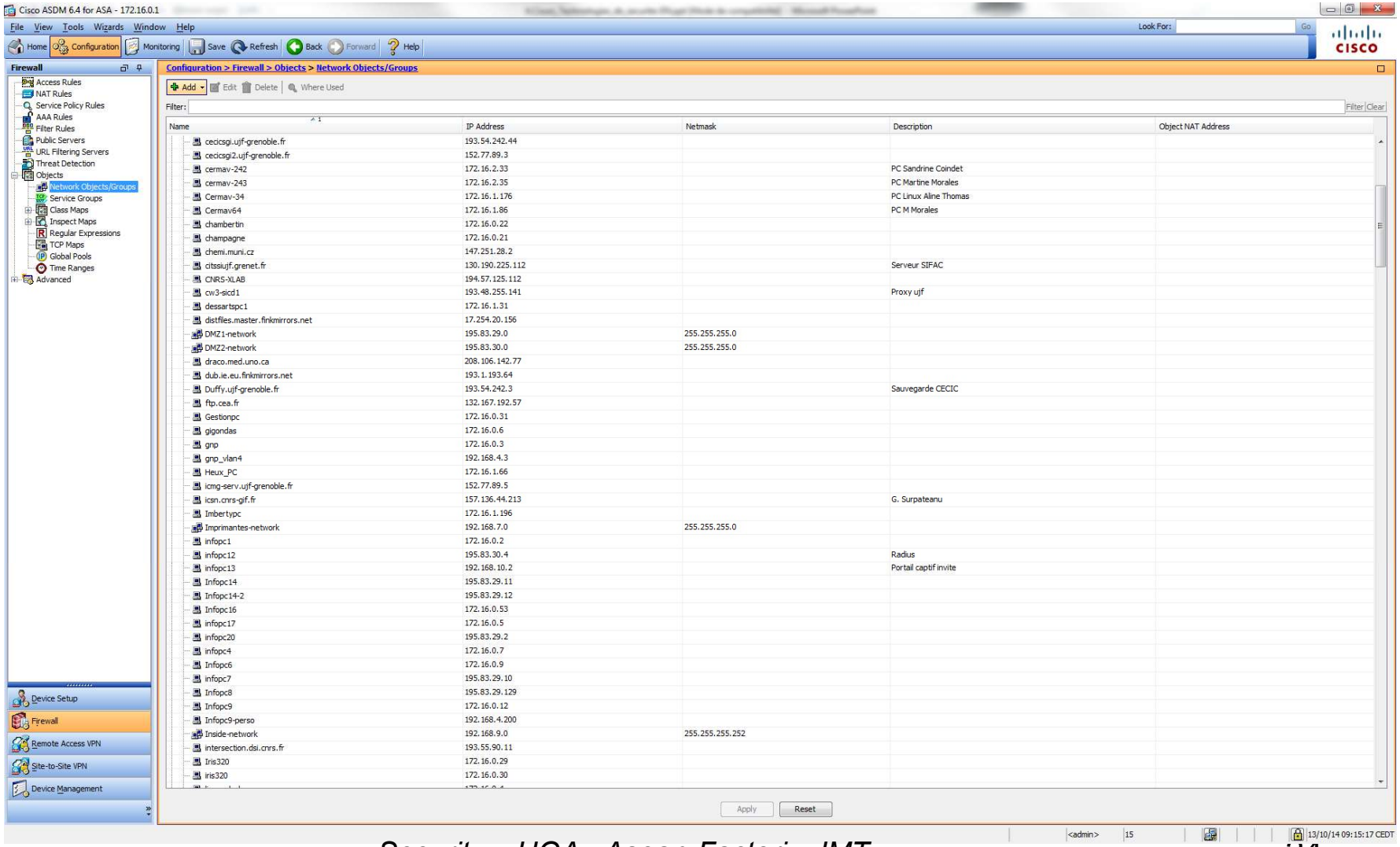
Filtering rules (Netascq)



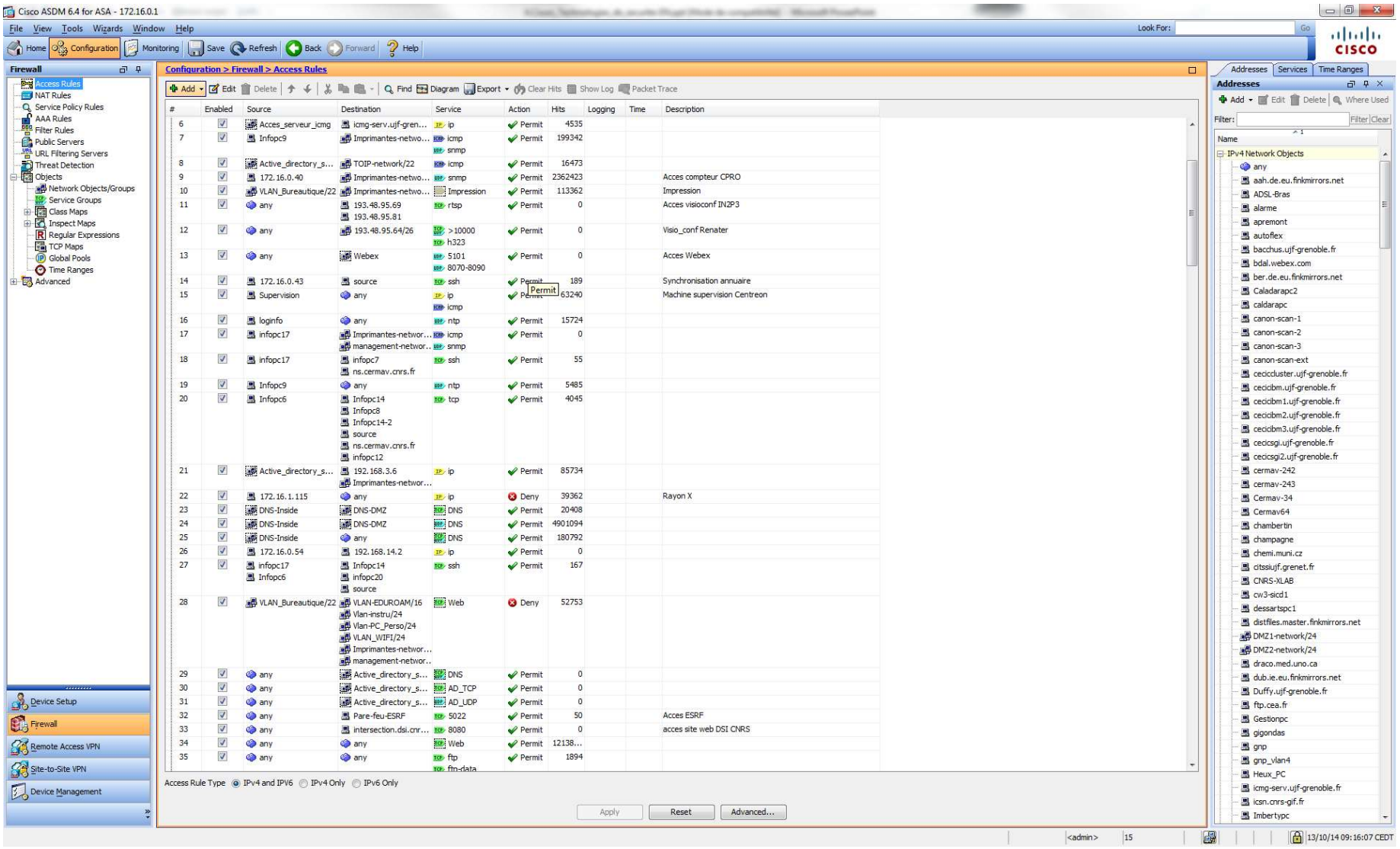
Translation (Cisco ASA)



Cisco ASA Firewall
Definition of the machines/hosts



Cisco ASA Filtering rules



Place of the firewalls

- Where should we put the firewalls?
  - At the connection interface between internal network and outside (Internet)
  - Between various portions of internal networks (large companies)
  - On each machine

Firewalls limitations

- Cannot prevent users or attackers using modems to reach inside the network
- Cannot prevent a misuse of the passwords (non respect of the passwords strategy by the users)
- Concentration of the traffic in only one point = bottleneck = source of fatal breakdown

Criteria for the good choice of a firewall

- Nature, type of applications (FTP, email, SNMP, Audio, Video)
- Distribution and Load Balancing (QoS)
- type of filtering (network level, application level)
- Records, logs, for audit purpose
- Tools, aids for administration
- Ability to support an encrypted tunnel (VPN)
- tools for monitoring, alarms, active audit
- Vulnerabilities: Intrusion => configuration changes, access, modification or erasure of traces of logging, viral infection
- Rating: cf Common Criteria organization (www.commoncriteriaportal.org)

Guiding principles for the configuration of a firewall

- Less privilege: do not grant the users with a higher level of rights that they need; to prohibit for example the peer-to-peer protocol within a company
- Default Prohibition: To prohibit everything by default: everything which is authorized should be explicitely authorized
- In-depth defense: to use the protection means at all the possible levels, for example by analyzing and filtering everything which can be analyzed at the level of the firewall. This principle prevents letting enter the network undesirable communications, even if another method of control is used more in-depth in the network
- Bottleneck: all the communications incoming and outgoing of the network must pass through the firewall. Other paths are strictly forbidden, such as for example unauthorized modems or access points
- Simplicity: the firewall filtering rules must be the simplest and most comprehensible as possible in order to avoid any error on behalf of the administrator or his successors (every rule should be documented and traceable)
- Participation of the users: the users must be involved in the firewall definition. They must indeed express their needs and receive in exchange the reasons and the objectives of the installation of such a device; the constraints related with the firewall will be accepted thus better.

DMZ, demilitarized zone (concept of perimetric security)

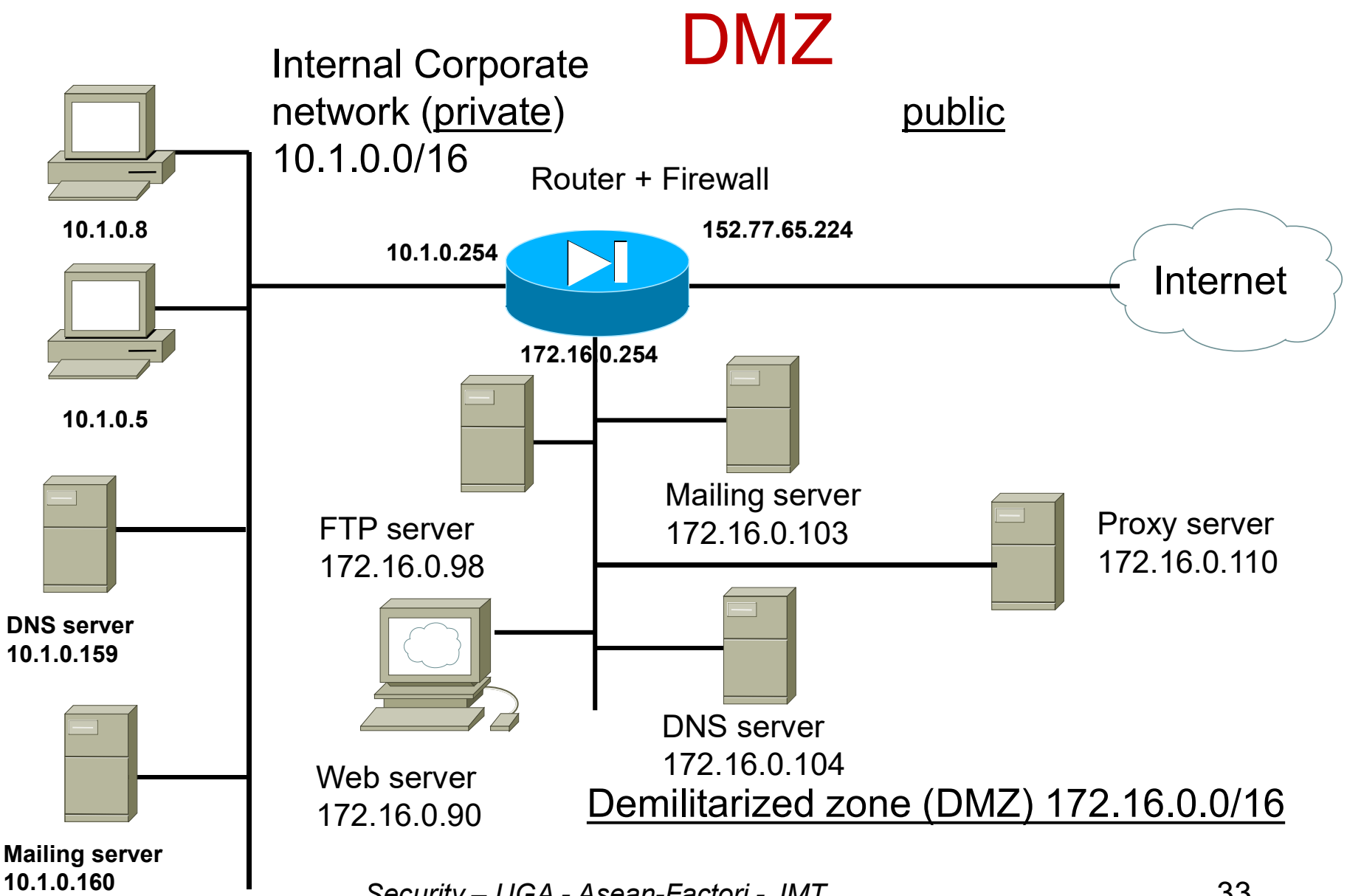
Demilitarized zone (perimetric security) (DMZ)

- Specific isolated zone of the internal network (between the public zone and the private zone)
  - Web server
  - Mailing server
  - FTP server
  - …
- This strategy allows the traffic coming from Internet to go in this zone, but not to penetrate elsewhere in the internal network
- Possibility of audit traffic exchanged with the DMZ
- Possibility of placing an intrusion detection system (IDS)

DMZ: its role

- To propose a zone
  - Receiving requests from outside
  - Does not allow direct communication from outside
  - Using its own addressing policy
- Access to the zone
  - Through the router from outside
  - Through the router + NAT from inside
- To realise a buffer zone
  - Can be corrupted
  - Does not reveal the presence of the local network

A network with a firewall/router…



Exercise 1

- We use a stateful firewall
- The machines from the inside network should be able to reach any machine in the DMZ or outside (for the mail)
- The machines from the DMZ should be able to reach any machine in outside BUT NOT inside (for the mail)
- Concerning http
  - Any machine from inside should NOT reach directly an http somewhere, but the request should be sent to the proxy machine (using the 3128 port)
  - Any machine from the DMZ should NOT reach directly an http somewhere, but the request should be sent to the proxy machine (using the 3128 port)
  - The proxy should be able to reach any http server (port 80) everywhere
- We should not forget the DNS aspects (port 53)

Exercise 1

- We use a stateful firewall
- The machines from the inside network should be able to reach any machine in the DMZ or outside (for the mail)
  - Access-list 1 permit mail 10.1.0.0/16 any eq 25
- The machines from the DMZ should be able to reach any machine in outside BUT NOT inside (for the mail)
  - Access-list 1 deny mail 172.16.0.0/16 10.1.0.0/16 eq 25 (should be before !)
  - Access-list 1 permit mail 172.16.0.0/16 any eq 25
- Concerning http
  - Any machine from inside should NOT reach directly an http somewhere, but the request should be sent to the proxy machine (using the 3128 port)
  - Access-list 1 permit tcp/udp 10.1.0.0/16 172.16.0.110 eq 3128
  - Any machine from the DMZ should NOT reach directly an http somewhere, but the request should be sent to the proxy machine (using the 3128 port)
  - No rule
  - The proxy should be able to reach any http server (port 80) everywhere
  - Access-list 1 permit tcp 172.16.0.110 any eq 80
- We should not forget the DNS aspects (port 53)
  - Access-list 1 permit tcp/udp 10.1.0.159 172.16.0.104 eq 53
  - Access-list 1 permit tcp/udp 172.16.0.104 a_specific_DNS_Server_outside eq 53
  - Access-list 1 deny any any any eq any

Exercise 2

- Let’s consider an architecture around a stateful firewall
- We wish to set up :
  1. All the machines of the internal network must ping the DMZ or the outside.
  2. All the machines in the DMZ must be able to ping outside but not on the internal network.
  3. All the machines from the inside must be able to reach http or https servers through the proxy.
  4. The DNS server of the internal network must be able to reach the DNS of the DMZ on port 53.
  5. The DNS server of the DMZ must be able to reach an external DNS (IP: 143.210.47.211).
- Actions to be carried out If necessary,
  - set up translation rules
  - Write filter rules and comment on them
- Audit of our security strategy
  - All the machines in the internal network have to be connected to the DMZ or to the outside. Is this a good strategy? Why is it a good strategy?
  - All the machines in the DMZ must ping the outside but not the internal network. Why this strategy?

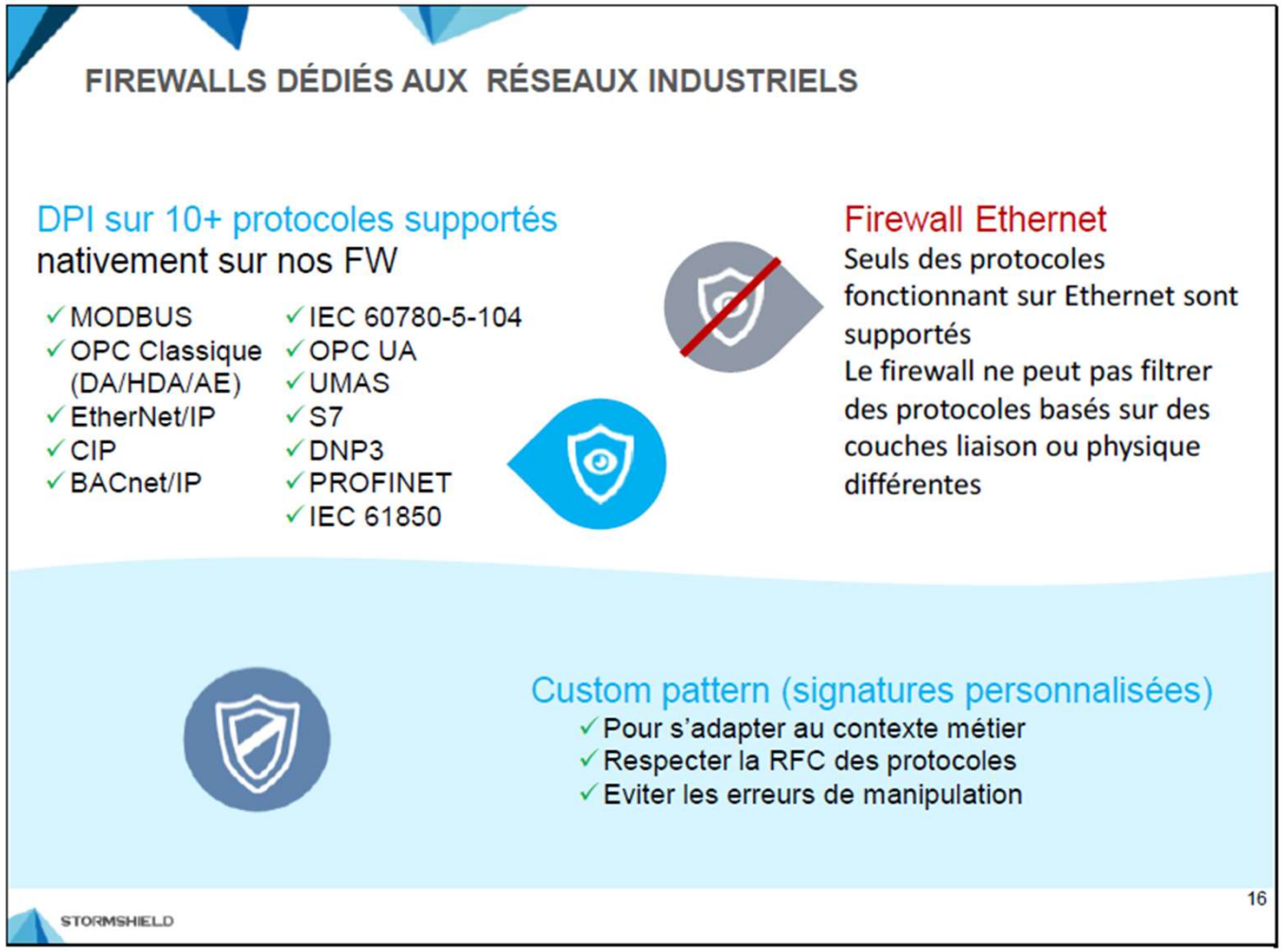
Règles de translation

- Elles sont nécessaires car nous avons utilisé des adresses privées
- 10.1.0.0/16 any 152.77.65.224 ; les machines du réseau interne sortent sur le réseau public en utilisant l'adresse publique unique 152.77.65.224
- 172.16.0.0/16 any 152.77.65.224 ; les machines de la DMZ sortent sur le réseau public en utilisant l'adresse publique unique 152.77.65.224

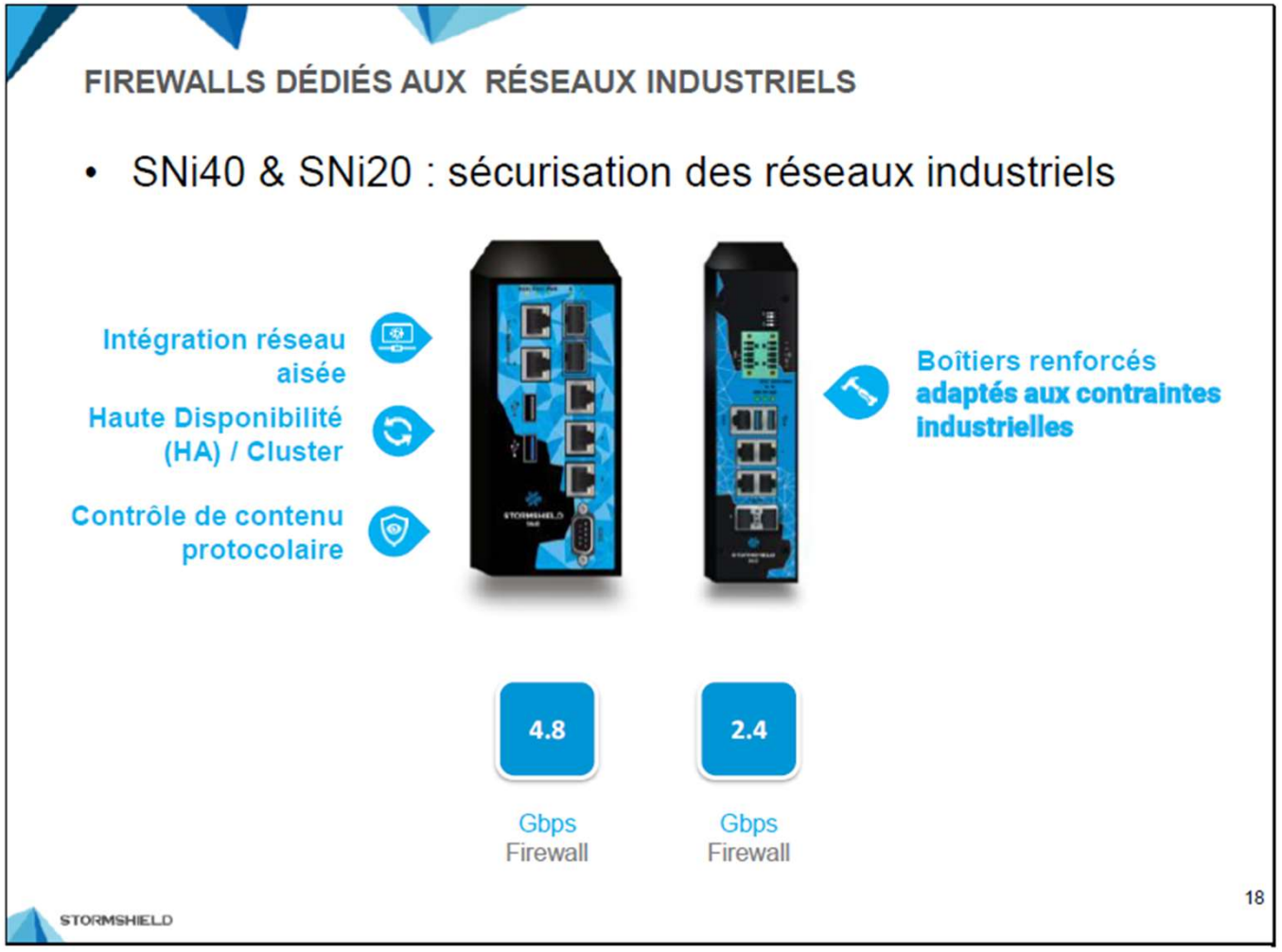
Règles de filtrage

Protocole	Source	Destination	Service (numéro port)	Action	Commentaire
ICMP	10.1.0.0/16	Any	Any	Pass	Réseau interne ping partout
ICMP	172.16.0.0/16	10.1.0.0/16	Any	Block	Pas de de DMZ vers réseau interne
ICMP	172.16.0.0/16	Any	Any	Pass	DMZ pingue partout
TCP	10.1.0.0/16	172.16.0.110	Httpproxy	Pass	Passage flux TCP réseau interne => proxy
TCP	172.16.0.110	Any	http, https	Pass	Passage flux TCP du proxy vers les serveurs http partout
TCP,UDP	10.1.0.159	172.16.0.104	Dns (port 53)	Pass	DNS Passe du réseau interne => DMZ
TCP,UDP	172.16.0.104	143.210.47.211	Dns (port 53)	Pass	DNS passe de DMZ vers DNS externe

Industrial firewall (from Stormshield)



Industrial network firewall



Cryptography

4.2. Ciphering and applications (Keys, cryptography)

4.2.1 Introduction

4.2.2 Symmetric cryptosystems
DES

4.2.3 Asymmetric cryptosystems
RSA

4.2.4 Comparison of encryption algorithms

4.2.5 Other applications of cryptography

4.2.6 Public Key Infrastructure

4.2.1 Introduction

4.2.1 Mission of cryptography

- To guarantee as well as possible
  - Confidentiality
  - Authenticity
  - Integrity

of data (or information) exchanged (or stored)

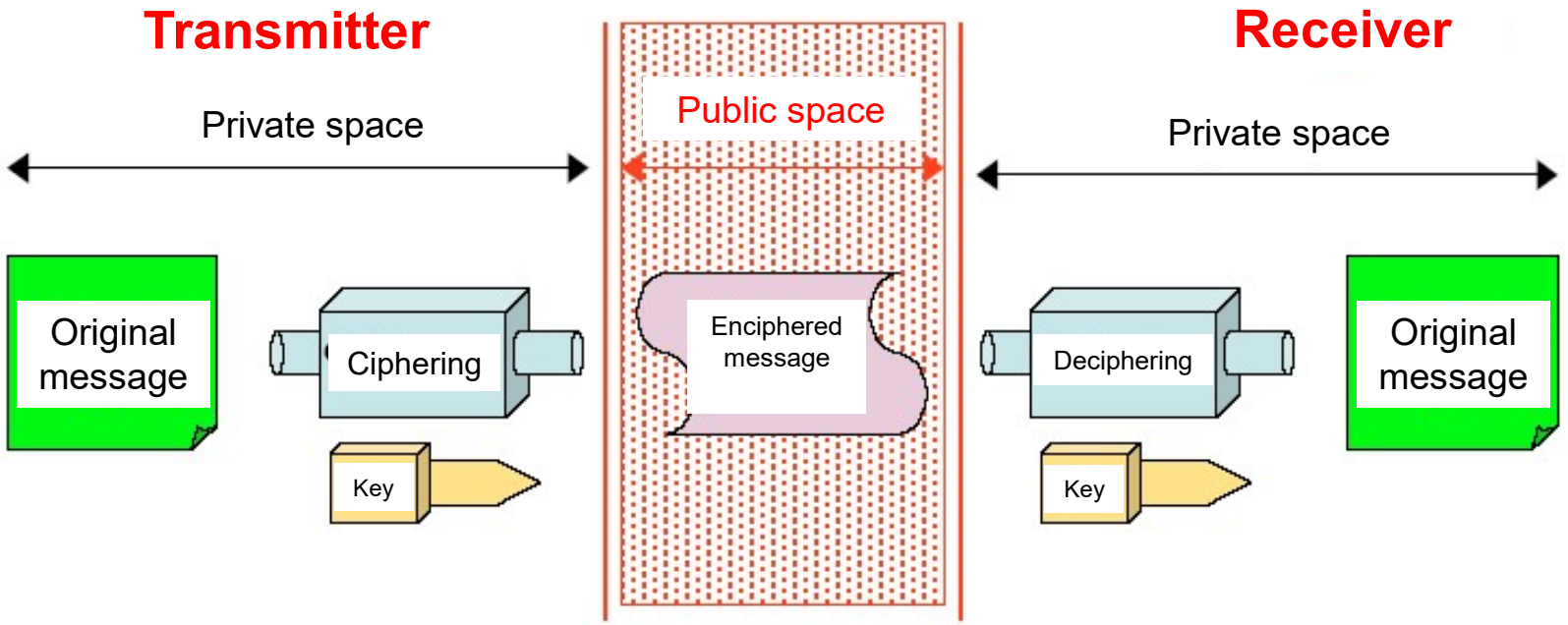
4.2.1 Cryptology and cryptography

- Cryptology
  - Science of enciphering
- Cryptography
  - Art to write the messages in an enciphered form
- Cryptanalyse
  - Attempt to decipher enciphered messages
- Modern cryptology
  - Based on digits
  - Use the results of arithmetic (some of these results are really old!!!)

4.2.1 Description of an enciphering system

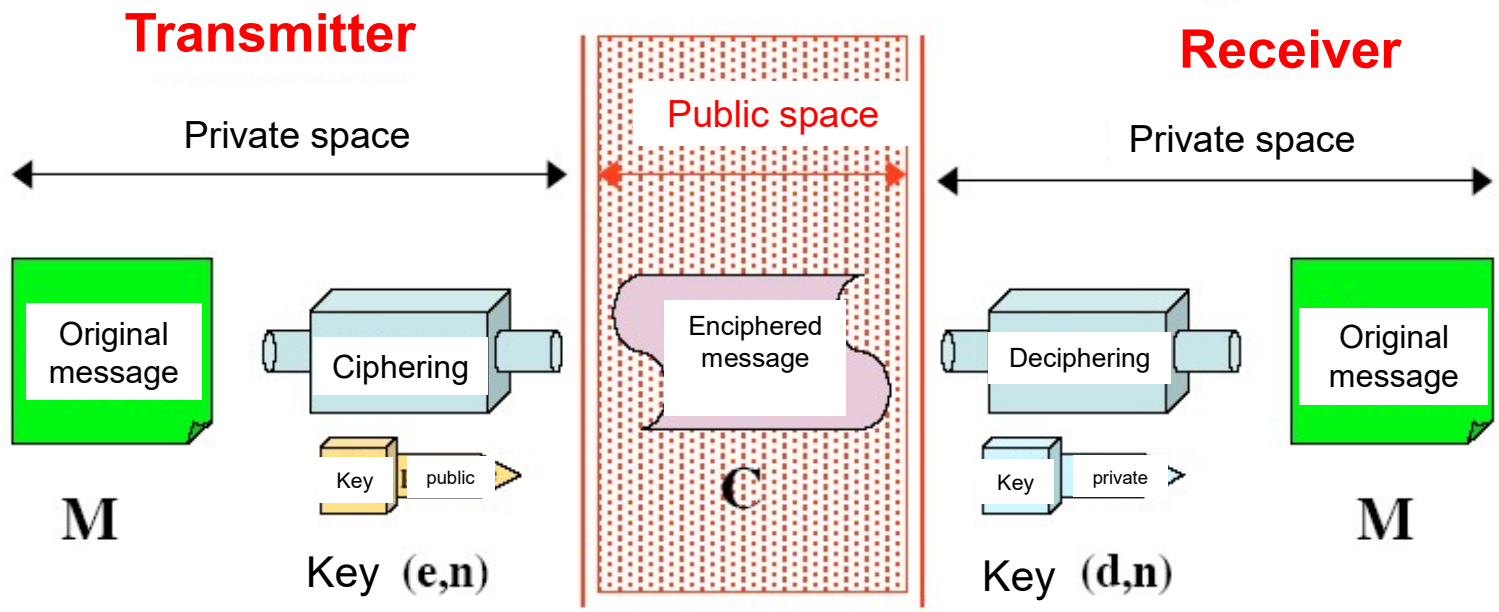
- Choice
  - Algorithm
  - One or more security keys
  - Transmission Media

4.2.1 Symmetric encryption



- ✓ The same key is used for enciphering and deciphering
- ✓ Problem: how to transfer the key

4.2.1 Asymmetric encryption



- Enciphering is achieved thanks to the public key
- Warrant that the owner of the private key ONLY can decipher the message

4.2.1 Size of the key

- Key enciphered on n bits => 2^n values
- The longest is the key
  - The most important is the number of possible keys
  - more time is necessary to compute and find the result
- A 40 bit-key (10^{12} different possibilities) => it has become now rather simple to break them
- Significant information => prefer a 128 bit-key (10^{38} possibilities) or a 256 bit-one
- Note: it is easier to find the key from a user or from the storage system than to find it thanks to deciphering

4.2.1 Robustness of the enciphering system

- Power of the algorithm (non-secret algorithm)
- Size of the key used
- Capacity to keep the secret keys in a protected way
- A system of enciphering is known as reliable, robust, sure, protected if it remains inviolable independently of the computing power or time available to an attacker
- It is known as operationally protected (*computational secure*) if its security depends on a series of realizable operations in theory, but unrealizable practically (too long processing times…)
- It is necessary to frequently change the enciphering key

4.2.2 Symmetric cryptosystems

4.2.2.1 Description

4.2.2.2 Examples

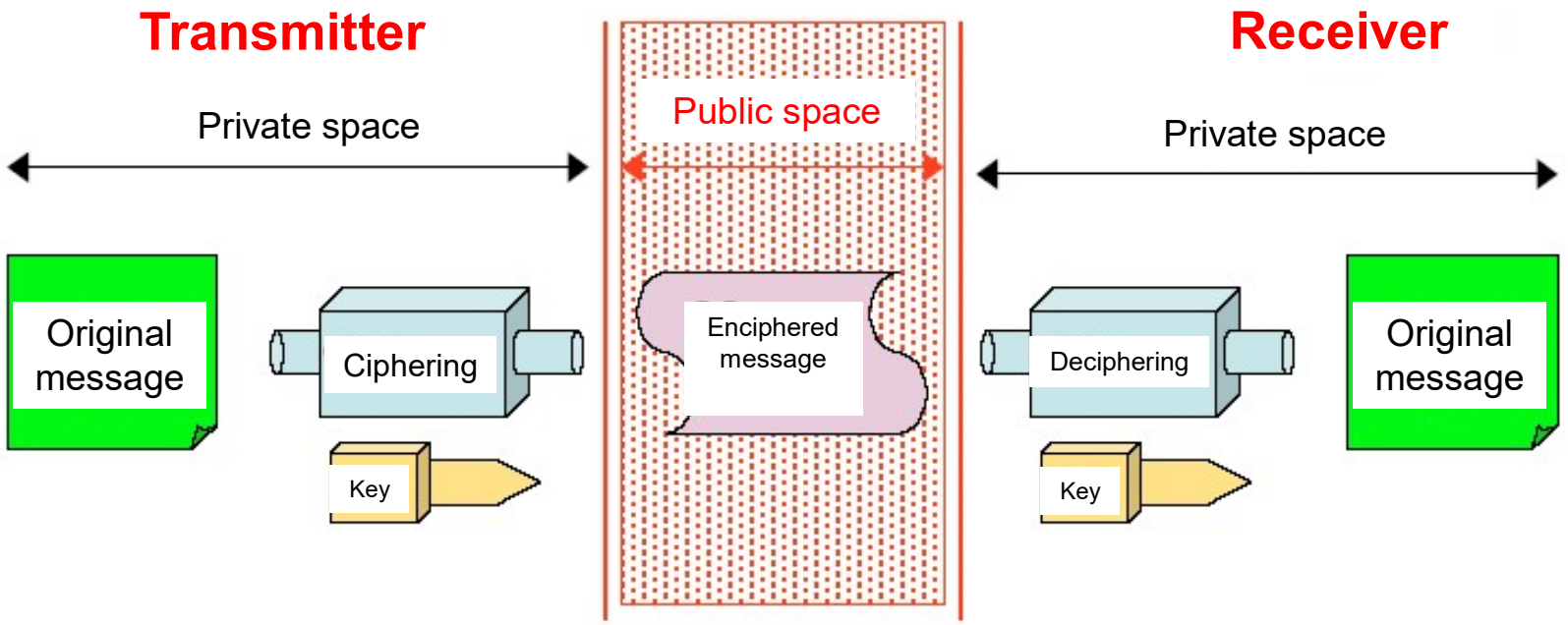
4.2.2.3 DES (Data Encryption Standard)

4.2.2.4 Other symmetric algorithms

4.2.2.1 Symmetric cryptosystems

- At the origin (once upon a time…), algorithms of enciphering (long continuations of operations on characters) => secretly preserved
- Today
  - Systems such as DES (Data Encryption Standard)
    - public algorithms enciphering and deciphering (known and fast)
    - Robust algorithms (it is almost impossible to break the algorithm) => only possibility: exhaustive research…

4.2.2.1 Symmetric cryptosystems



- ✓ The same key is used for enciphering and deciphering
- ✓ Problem: how to transfer the key

4.2.2.1 Symmetric cryptosystems

- Set of keys: K
- Algorithm for enciphering: E
- Algorithm for deciphering: D
- ∀ (whatever) a k key and the t message:

$$D(E(t,k),k)=t$$

4.2.2.2 Enciphering examples: substitutions (Caesar translation algorithm)

- Replacement of a letter by another
- Robustness?
  - Identical frequency contents
  - Can be easily “broken” easily starting from a message of 28 letters…
- Example: “right” shift of two letters
  - *Bonjour* => Dqplqwt
- Another example: shift of a letter towards the left
  - IBM => HAL (“2001: A Space Odyssey”)

4.2.2.2 Enciphering examples : poly-alphabetical codes (1/3)

- To code a character in several different ways according to its position
- One chooses a key which is used as an entrance point in a poly-alphabetical grid
- Each character of the key indicates an alphabet (K) grid, well defined
- To code a character from the plaintext, we should use the grid

4.2.2.2 Enciphering examples : poly-alphabetical codes (2/3)

- Let's consider an alphabet {A, B, C, D}

text t / key k	A B C D
A	C D B A
B	D C A B
C	C A B D
D	B D A C

plaintext: ABCB ACCB AACB B

Key: DBBC BAAC DDBB C

Encrypted text: BCAA DBBA BBAC A

- Require very large size keys not to be very vulnerable …

4.2.2.2 Enciphering examples : poly-alphabetical codes (3/3)

- Encrypt
- ACDBA with the key: BDBA

4.2.2.2 Enciphering examples : poly-alphabetical codes (3/3)

- Encrypt
- ACDBA with the key: BDBA

- Result:
- DABDD

4.2.2.2 Enciphering examples : Operations at the bit level

- Change of scale: character => bit
- Use of the “digital” techniques, mathematical jamming (“*brouillage*”) techniques
  - Permutations
  - Transpositions
  - Substitutions of forms (shapes, “*motifs*”)
- Use of the exclusive OR (or XOR) Boolean function => bijective operation + equals to its opposite

4.2.2.2 Enciphering examples: Operations at the bit level Distance permutations

- d1=1, d2 = 01, d3 = 001, d4= 0001…
- Distance permutation (di, dj)
- Example: TS
- Form substitution
- Example (d1, d2, d3, d4) substituted by (d2d3, d3d1, d1d4, d1d3) => increases the size of the data
- TS
- 54 53
- 0101 0100, 0101 0011
- d2 d2 d2 d4 d2 d3 d1
- d3d1 d3d1 d3d1 d1d3 d3d1 d1d4 d2d3
- 0011 0011 0011 1001 0011 10001 01001
- What is the size of the original message? The encrypted one?

- Then to decipher…

4.2.2.2 Example

- Encipher *BON* by substituting (d1, d2, d3, d4, d5, d6) by (d2d3, d3d1, d1d4, d1d3, d2d4, d5d6) with d1=1, d2 = 01, d3 = 001, d4= 0001…
- BON (each character is encoded as a 8-bit hexadecimal ASCII code)
- What is the size of the original message? The encrypted one?
- 42, 4F, 4E

4.2.2.2 Example

- Substitution (d1, d2, d3, d4, d5, d6) by (d2d3, d3d1, d1d4, d1d3, d2d4, d5d6)
- 42, 4F, 4E
- 0100 0010 0100 1111 0100 1110
- Encoding
- d2 d5 d3 d3 d1d1d1 d2 d3d1 d1, ! 0 is not taken into account…
- Encryption
- d3d1 d2d4 d1d4 d1d4 d2d3d2d3d2d3 d3d1d1d4 d2d3d2d3
- 0011 010001 10001 10001 010010100101001 001110001 0100101001
- What is the size of the original message? The encrypted one?
- 24 bits (3 bytes) for the original message, 54 bits for the encrypted one

4.2.2.2 Inversion of bits according to a random suite

- Purpose: Transforming each byte of a file F by reversing certain bits by operations of binary negation
- Let's consider a pseudo-random numbers suite (a_n)
- For each byte, the bits to be reversed are obtained by calculating the modulo 8 of the terms of the suite (a_n). The suites of modulo 8-numbers is called (b_n)
- If $b_{n+1} \leq b_n$ then one passes to the following byte => the nbr of bits which are reversed in a byte is random…

4.2.2.2 Inversion of bits according to a random suite : example 1

- (a_n) = (2, 14, 7, 11, 74, 25, 32, 37, 152, 99, 7) => (b_n) = (2, 6, 7, 3, 2, 1, 0, 5, 0, 3, 7)
- F= 01001010 10010101 00101001 00010100 11010110 11110001
- And
- F'= 01**1**0101**0**1 100**0**0101 00**0**01001 0**1**010100 **0**1010**0**10 **0**11**0**000**0**

Bit 2 Bit 6 Bit 3 Bit 2…

4.2.2.2 Inversion of bits according to a random suite : example 2

- *BON* with the random suite (a_n) = (3, 4, 11, 27, 32, 25, 12, 153, 77, 7) modulo 8
- 42, 4F, 4E

4.2.2.2 Inversion of bits according to a random suite : example 2

- *BON* with the random suite (a_n) = (3, 4, 11, 27, 32, 25, 12, 153, 77, 7) modulo 8
- 42, 4F, 4E
- 01000010 01001111 01001110
- (b_n) = (3,4,3,3,0,1,4,1,5,7)
- 01011010 01011111 01011110

4.2.2.2 Inversion of bits according to a random suite : example 3

- *BON* with the random suite (a_n) = (3, 4, 11, 27, 32, 25, 12, 153, 77, 7) **modulo 9**
- BON
- 42, 4F, 4E

4.2.2.2 Inversion of bits according to a random suite : example 3

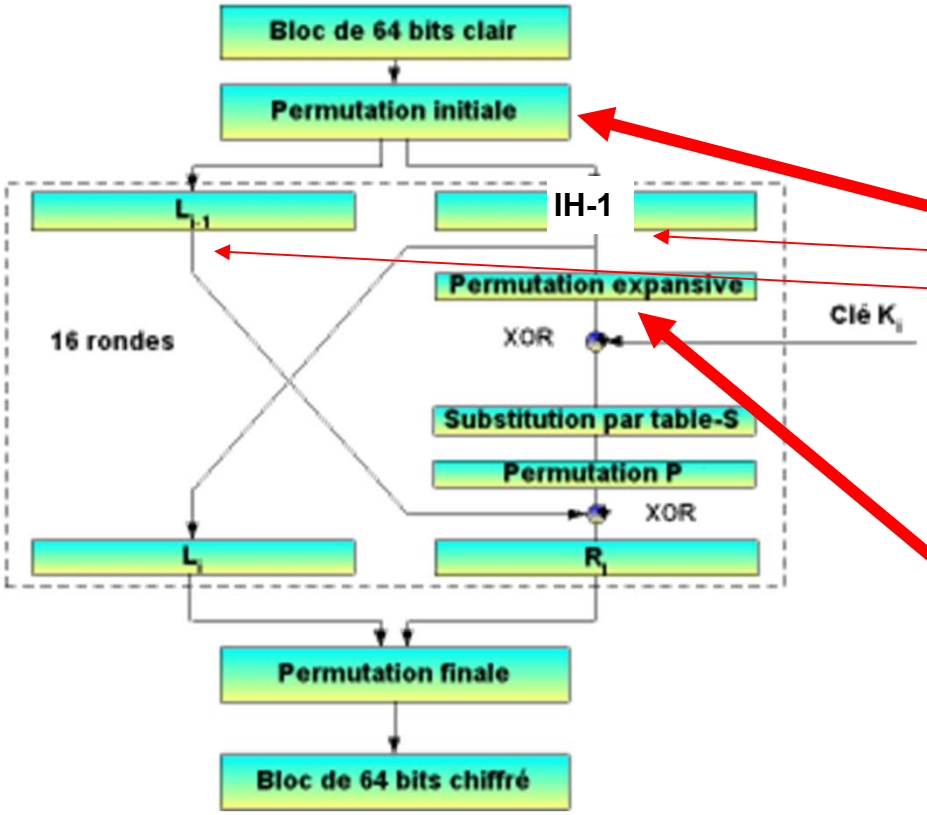
- *BON* with the random suite (a_n) = (3, 4, 11, 27, 32, 25, 12, 153, 77, 7) **modulo 9**
- BON
- 42, 4F, 4E
- $(b_n) = (a_n)$modulo 9 = (3, 4, 2, 0, 5, 7, 3, 0, 5, 7)
- 0100 0010 0 100 1111 01 00 1110
- 0101 1010 0 101 1111 01 10 1111

4.2.2.3 DES (Data Encryption Standard)

4.2.2.3 The standard algorithm for enciphering: IBM DES (Data Encryption Standard)

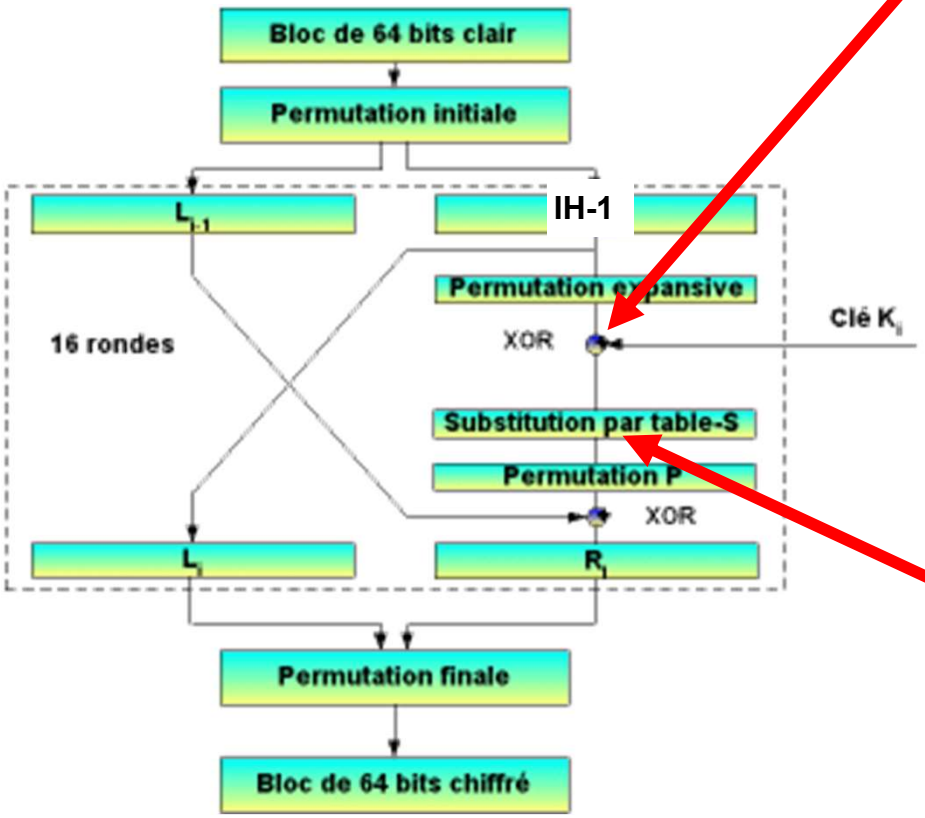
- Created 1977
- Algorithm for blocks encryption
- At first for classified or secret documents
- Today software and smart cards industry
- Enciphering and deciphering speed (rapidity)
  - Can be developed in less than 200 lines
  - Very fast on dedicated electronic charts
    - Smart cards
    - Electronic systems of telecommunications
- Implementation on Unix, Windows and MacOs available on Internet (chalmers.se/pub for example)

4.2.2.3 DES



- 1st step: Initial permutation
  - Each 64 bits-block of undergoes a permutation then is divided into two blocks (L_0 et R_0) of 32 bits
- *Beginning of the first iteration*
- 2nd step: expansive permutation
  - The 32 bits of R_0 enter a table of selection of bits, they are mixed and repeated. **48** bits are obtained

4.2.2.3 DES



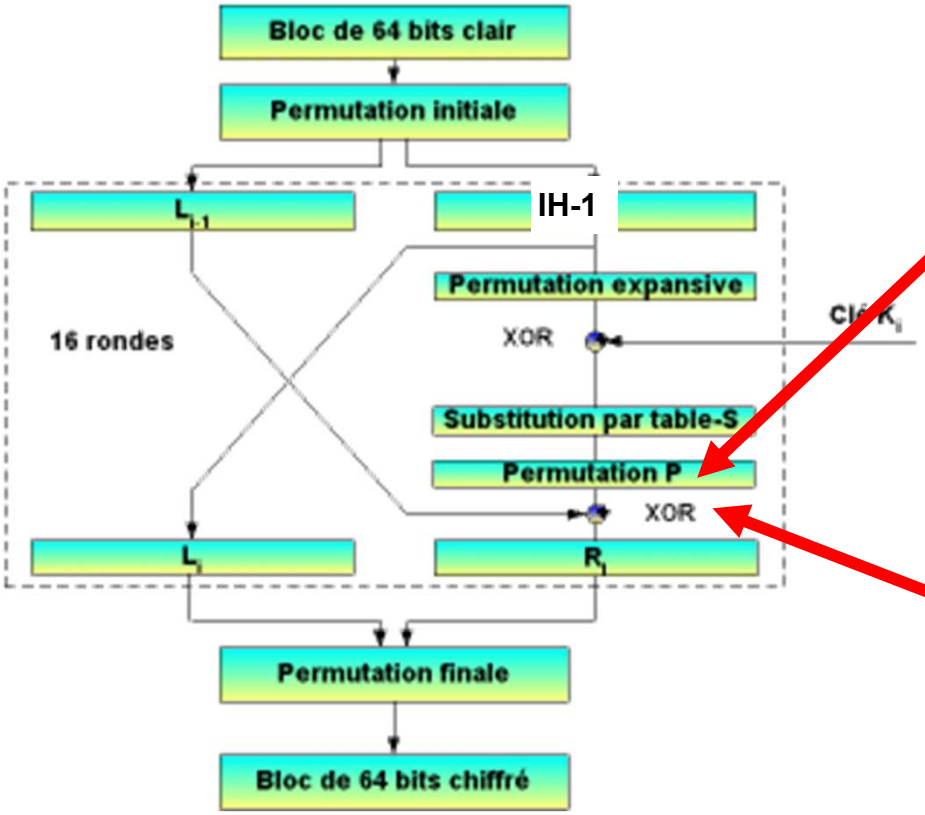
- 3rd step: Calculation of the Ki key
  - Calculation of the K_1 key (48 bits) from the origin key K
  - Transformation of the preceding block with a XOR with the K_1 key
- 4th step: substitution by S table
  - The preceding result is broken up into 8 R_{0i} blocks of 6 bits each (b_1, b_2, b_3, b_4, b_5, b_6)
  - This decomposition makes it possible to calculate a position in a table of selection (S) composed of 8 blocks of 16 columns and 4 rows
    - The number (b_1, b_6) represents the number of the row
    - The number ($b_2b_3b_4b_5$) represents the number of the column
  - We replace the R_{0i} block with the 4 bits-block found in the table => total of 32 bits for the 8 blocks

Table 1	{14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7}, {0,15,7,4,14,2,13,1,10,6,12,11,9,5,3,8}, {4,1,14,8,13,6,2,11,15,12,9,7,3,10,5,0}, {15,12,8,2,4,9,1,7,5,11,3,14,10,0,6,13},
Table 2	{15,1,8,14,6,11,3,4,9,7,2,13,12,0,5,10}, {3,13,4,7,15,2,8,14,12,0,1,10,6,9,11,5}, {0,14,7,11,10,4,13,1,5,8,12,6,9,3,2,15}, {13,8,10,1,3,15,4,2,11,6,7,12,0,5,14,9},
Table 3	{10,0,9,14,6,3,15,5,1,13,12,7,11,4,2,8}, {13,7,0,9,3,4,6,10,2,8,5,14,12,11,15,1}, {13,6,4,9,8,15,3,0,11,1,2,12,5,10,14,7}, {1,10,13,0,6,9,8,7,4,15,14,3,11,5,2,12},
Table 4	{7,13,14,3,0,6,9,10,1,2,8,5,11,12,4,15}, {13,8,11,5,6,15,0,3,4,7,2,12,1,10,14,9}, {10,6,9,0,12,11,7,13,15,1,3,14,5,2,8,4}, {3,15,0,6,10,1,13,8,9,4,5,11,12,7,2,14},
Table 5	{2,12,4,1,7,10,11,6,8,5,3,15,13,0,14,9}, {14,11,2,12,4,7,13,1,5,0,15,10,3,9,8,6}, {4,2,1,11,10,13,7,8,15,9,12,5,6,3,0,14}, {11,8,12,7,1,14,2,13,6,15,0,9,10,4,5,3},
Table 6	{12,1.10,15,9,2,6,8,0,13,3,4,14,7,5,11}, {10,15,4,2,7,12,9,5,6,1,13,14,0,11,3,8}, {9,14,15,5,2,8,12,3,7,0,4,10,1,13,11,6}, {4,3,2,12,9,5,15,10,11,14,1,7,6,0,8,13},
Table 7	{4,11,2,14,15,0,8,13,3,12,9,7,5,10,6,1}, {13,0,11,7,4,9,1,10,14,3,5,12,2,15,8,6}, {1,4,11,13,12,3,7,14,10,15,6,8,0,5,9,2}, {6,11,13,8,1,4,10,7,9,5,0,15,14,2,3,12},
Table 8	{13,2,8,4,6,15,11,1,10,9,3,14,5,0,12,7}, {1,15,13,8,10,3,7,4,12,5,6,11,0,14,9,2}, {7,11,4,1,9,12,14,2,0,6,10,13,15,3,5,8}, {2,1,14,7,4,10,8,13,15,12,9,0,3,5,6,11}.

Figure 10.2. *Table de substitution non-linéaire utilisée à l'étape 4.*

4.2.2.3 DES: nonlinear substitution table used for the 4th step

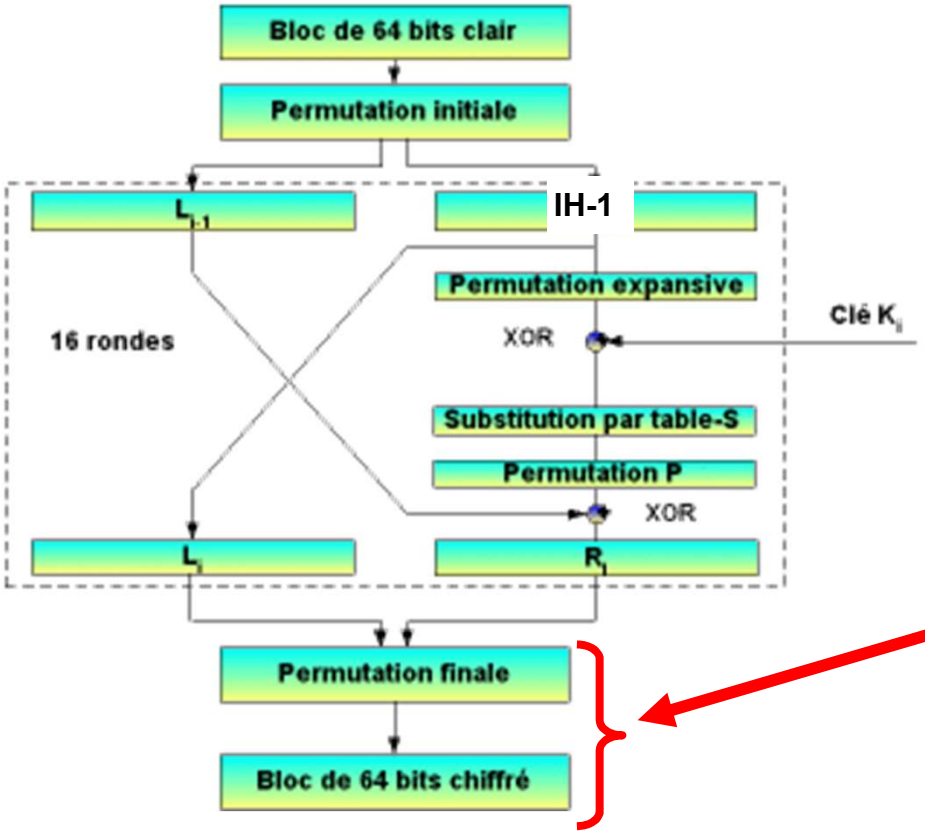
4.2.2.3 DES



- 5th stage: New permutation
  - The 32 bits are permuted in the following way:

    $b_{16}, b_7, b_{20}, b_{21}, b_{29}, b_{12}, b_{28}, b_{17}, b_1, b_{15}, b_{23}, b_{26}, b_5, b_{18}, b_{31}, b_{10}, b_2, b_8, b_{24}, b_{14}, b_{32}, b_{27}, b_3, b_9, b_{19}, b_{13}, b_{30}, b_6, b_{22}, b_{11}, b_4, b_{25}$
- 6th stage: exclusive OR
  - The result of step 5 is subjected to a XOR with L0 to form R1
  - L1=R0 is posed
- *End of the first iteration*

4.2.2.3 DES



- The procedure is then repeated 14 times (round 2 to 15) using the K_i key
- The 16th iteration ends with a final permutation

4.2.2.3 Calculation of the K_i key starting from the key of origin K

- The security of thus rests on the key K
- This key is an alphanumeric chain of 64 bits (8 bytes)
- These 8 bytes undergo a P_1 permutation where the first 8 bits (parity) are eliminated
- then form two blocks
  - $L_0 = (b_{57}, b_{49}, b_{41}, b_{33}, \ldots, b_k, b_{k-8}, \ldots)$
  - $R_0 = (b_{63}, b_{55}, b_{47}, b_{39}, \ldots, b_k, b_{k-8}, \ldots)$
- The key K_1 is obtained by shifting L_0 and R_0 of a bit towards the left => one obtains the blocks L_1 and R_1
  - The blocks L_1 and R_1 undergo a P_2 permutation which does not turn more than 48 bits => those form the K_1 key
- This calculation spreads to generate 15 other keys K_i starting from the blocks L_{i-1} and R_{i-1}. Attention the number of bits of decay of L_i and R_i varies as a function of i in the following way:

$$\{1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1\}$$

4.2.2.3 Decoding of DES

- DES is a symmetric process, the operations of coding are thus equal to their reverse => decoding uses the same operations as encoding, by using the same keys k_i, but beginning with k_{16} instead of k_1

4.2.2.3 Security of DES

- Non linearity of the functions of substitution (4th step), they are designed to resist a cryptanalyse. N.B a small change in these tables can ruin the security of DES
- The procedures used cause that the cryptanalyse fails in an attempt to decrypt thanks to a frequency analysis of the encrypted texts
- 16 iterations per block scramble the plaintext and propagate the jamming quasi uniformly
- The number of keys implies the practical impossibility, even with large samples, to find the key K starting from the encrypted text (time estimated at 500 years at the end of the years 1990).
- The possibility of using longer keys, with 2 times more bits (128 bits), the test of all the possibility would take 268.000.000 times more time…
- But DES security can be easily corrupted from 1997 using an exhaustive research (size of the key)

4.2.2.4 Other symmetric algorithms

- 3DES (triple DES, 168 bit-key)
  - It consists in using three times the DES algorithm with three keys k_1, k_2 and k_3 : $m'=DES_{k1}(DES_{k2}(DES_{k3}(m)))$
  - Alternative with 2 keys and by using twice the algo of encrypting and once the algo of decoding: $m'=DES_{k1}(DES^{-1}_{k2}(DES_{k1}(m)))$, this alternative is considered more secure
- DESX (DES XORed), GDES (Generalized DES), RDES (Randomized DES): comes from DES, using larger keys.
- AES (Advanced Encryption Standard) published in 1998, recommended from 2002
  - developed to replace DES and offer a better security
  - Use 128, 192 or 256 bit-keys
  - N.B. At the end of 2003, the American department of defense approved its authorization
  - Used in IPSec (secured IP) and IKE (Internet Key Exchange)
  - Considered as “unbreakable” and the surest system nowadays

AES (some aspects)

- DES obsolete at the end of the 90s:
  - key size too small,
  - execution time too long (accentuated with triple-DES),
  - existence of back doors from NSA (?)
- 1997: competition for AES by NIST (National Institute of Standards and Technology, USA)
  - Can use 128, 192 or 256 bit keys
  - Block size of at least 128 bits
  - Fast execution on as many platforms as possible
- Rijndael algorithm designed by Daemen and Rijmen, from UC Louvain (BE) in 2000
- Cryptanalysts are constantly working on attack algorithms to detect vulnerabilities. For example, of the 10 rounds of AES-128, if it is easy to break one round of the AES, there are no significant results today (2016) on more than 6 rounds

4.2.2.4 Other symmetric algorithms

- RC (Rivest Cipher) from RC2 to RC6
  - Algorithms with symmetric keys diffused by RSA Security Inc. (www.rsasecurity.com)
  - Use a variable key length (up to 2048 bits)
  - Used to make confidential the applicative flows
- IDEA (International Data Encryption Algorithm)
  - Key of 128 bits keys to code 64 bits-blocks of data
  - Used by the protected protocol of mail PGP (*Pretty Good Privacy*) http://sebsauvage.net/logiciels/pgp.html
- Twofish
  - Key size up to 256 bits
- Blowfish
  - Symmetric encryption algorithm developed in 1993, largely replaced by AES

4.2.2.4 Conclusion on symmetric systems

- Sure
- Fast

but

- Need to exchange the key
  - Problem of security during the transmission of the key

- Problems of the management of keys

4.2.3 Asymmetric cryptosystems

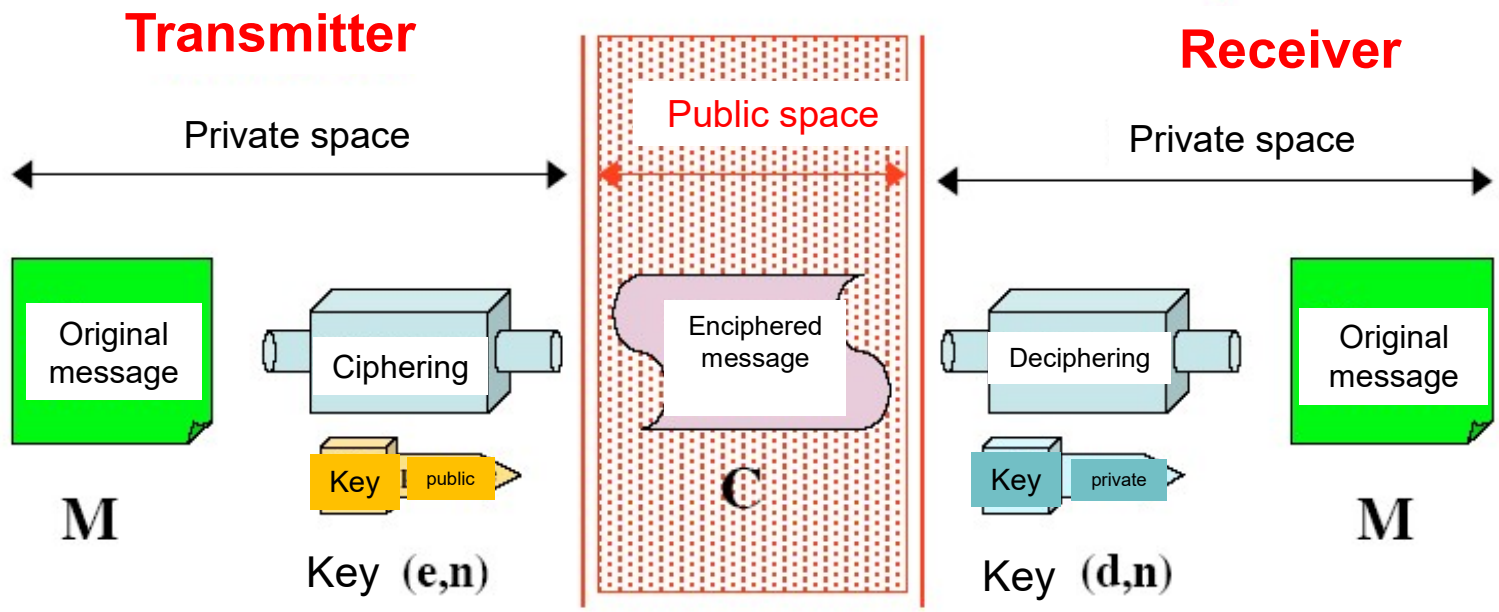
4.2.3.1 Principles
4.2.3.2 RSA (Rivest, Shamir and Adleman)
4.2.3.3 Other public keys-algorithms

4.2.3.1 Asymmetric ciphering

- Developed to resolve the problem of the distribution of the keys of the symmetric systems
- Use of a single couple of two complementary keys, calculated one compared to the other (public key and private key)
- the public key only can be known of all, the private key must be confidential and be treated like a secrecy
- Ex: one must know the public key of a recipient to send statistical data to him

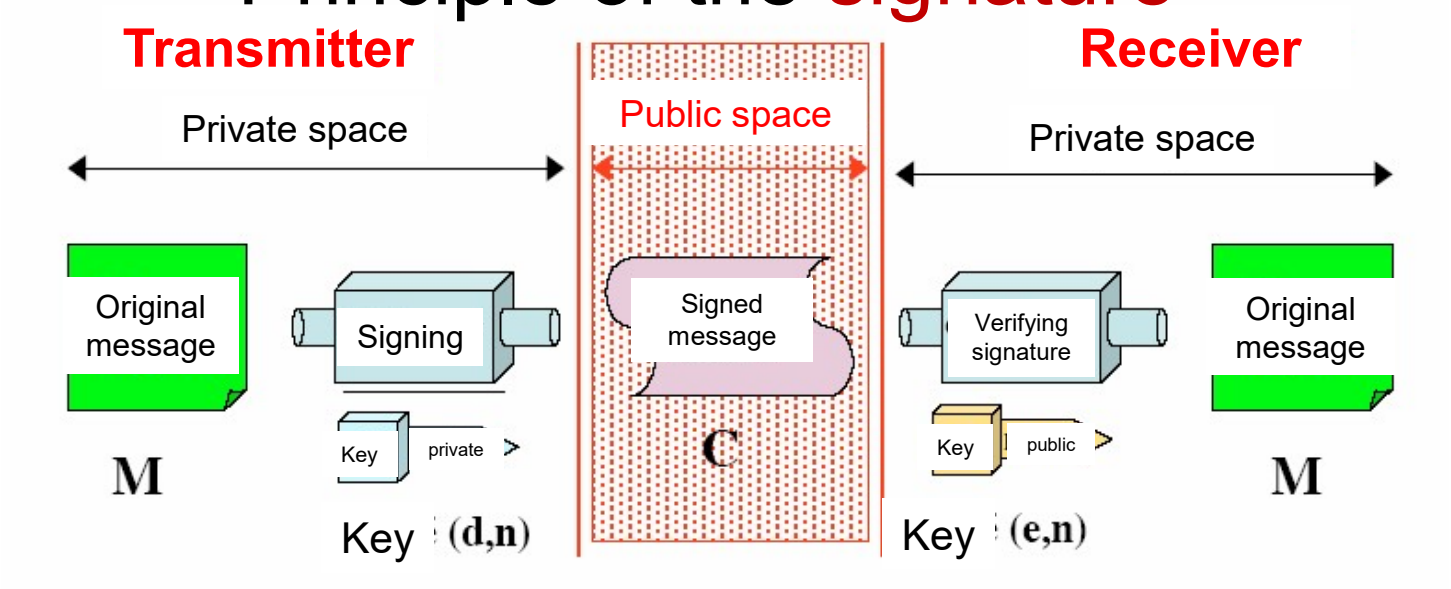
  ⇒ He will decipher them at reception with his private key, which he is alone to know
- Execution time of these algorithms produces important processing times processor

4.2.3.1 Asymmetric cryptography: Protection of the confidentiality



- ✓ Enciphering is achieved thank to the public key
- ✓ Warrant that the owner of the private key ONLY can decipher the message

4.2.3.1. Asymmetric cryptography: Principle of the signature



-to sign a message, the private key is used (the hash is encrypted with the private key)

-Garanty of authentication, but no confidentiality

- deciphering with the public key is a proof that the private key only was used for the signature

Combination

- Sender side
  - 1. Encryption using the public key of the receiver
  - 2. Signature using the private key of the sender
- Transmission
- Receiver side
  - 1. Verification of the signature using the public key of the sender
  - 2. Decryption using the private key of the receiver

4.2.3.1 Asymmetric systems: principles

- To allow the exchange of encrypted information without meeting to exchange the keys
- security is based on the fact it is practically impossible to solve a data-processing problem which is “difficult”, for which the search for a solution amounts to thousands, or even billion years.

4.2.3.1 Asymmetric systems: some concepts (1/2)

- Functions with single direction
  - According to the theory of the calculability and algorithmic complexity, a function has a single direction if:
    - The function f is calculable quickly
    - Its reverse f^{-1} needs a very long time to be calculated
  - Example of function with a single direction:

$a^p \bmod n$ => called exponential of the variable ***p***

Base a fixed

n is the product of two “large” prime numbers

the reverse function (called discrete logarithm) is calculable “with difficulty”

4.2.3.1 Asymmetric systems: some concepts (2/2)

- Trap functions or secret breach function
  - Function with one single direction except for any person knowing a secrecy, or a breach, allowing to calculate an fast inversion algorithm
  - Example of secret breach function:

$a^p \bmod n$ => called modular exponentiation of the variable ***a***

Power p fixed, n product of two prime numbers

The existence of a reciprocal algorithm which allows the calculation of the p^{th} roots modulo n of a is proved, but this algorithm is not known

On the other hand, if one knows the factorization of n (the breach), it is easy to reverse the modular exponentiation

4.2.3.1 Definition of asymmetric crypto-systems (according to **Whitfield DIFFIE and Martin HELLMAN**)

- It exists a public algorithm for coding: E (encoder)
- A secret algorithm for decoding: D (decoder)
- E and D are a function of the key K used
- m=D(E(m))
- D and E are calculable immediately for any person knowing the key K
- The knowledge of E should not make it possible to know D, if not the security of encoding is not guaranteed
  - The process is public and depends on the key =>the decoding algorithm D remains secret and impossible (in reasonable time) to calculate starting from E => E should be a secret breach function, the breach being the algorithm D

Factorisation (Gary Blackwood « Mysterious messages, a history of codes and ciphers », 2009

- Factoring is not easy
- Ex: 11 * 13 = 143 ; You have to determine what two prime numbers (or factors) can be multiplied to make that numbers. A small number like 143 can be factorised easily by trail and errors.

A high-speed computer can do the job, of course—eventually. In 1994, researchers at Oregon State University started with this number:

2219620528659701952660120743076100427390924
3570733965516770339373353207430502358024273
0327563320054080668946066967922195450939671
27330845624462896060302682123 17

They found it was the product of

1653723785156468892426140704164885399065774 3

muliplied by

4971867800323378818776339900596001648747 65
9834953921156974700575915322824191116704 32
0092701688428573103024883134912641 9

Using thirty computer stations, the task took them eight weeks.

- **Largest Known Prime Number:**
- **$2^{82\ 589\ 933}-1$**
- **Found in December 2018, composed of** 24 862 048 digits
- 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223…

4.2.3.2 RSA

4.2.3.2 RSA ciphering Protocol

- Proposed in 1977 by the cryptologists Rivest, Shamir and Adleman
- Based on the modular exponentiation (trap function)
- Main applications
  - Sending of confidential messages to a person
  - Authentication by any person of the message sent by an individual
  - Authentication by password (smart cards, bank cards)
- security based on the impossibility of carrying out the factorization of a large number of a few hundreds of digits in a reasonable time
  - The user selects two large prime numbers p and q, then multiplies them to obtain n=p.q (integer modulating the RSA protocol)

4.2.3.2 RSA (encryption, confidentiality)

- The algorithm is remarkable by its simplicity. It is based on the prime numbers.
- To encrypt a message:

c = m^e mod n

- To decrypt: **m = c^d mod n**
  - **m** = clear message
    c = encrypted message
    (e, n) constitutes the public key
    (d, n) constitutes the private key
    n is the result of the multiplication of 2 prime numbers
    ^ is the power function (a^b: a power b)
    mod is the operation of modulo (rest of the *integer division*)

4.2.3.2 RSA (digital signature to protect authentication and integrity ⇔ signature)

- The algorithm is remarkable by its simplicity. It is based on the prime numbers.
- To sign a message:

s = m^d mod n

- To verify the signature: **m = s^e mod n**
  - **m** = clear message
    s = signed message
    (e, n) constitutes the public key
    (d, n) constitutes the private key
    n is the result of the multiplication of 2 prime numbers
    ^ is the power function (a^b: a power b)
    mod is the operation of modulo (rest of the *integer division*)

4.2.3.2 RSA: Creation of a pair of keys

- It is simple, but the **e**, **d** and **n** should be chosen with care! And the calculation of these three numbers is delicate.
- Methodology:
  - The user selects two large prime numbers p and q, and multiplies them to obtain n=p.q (integer modulating the RSA protocol), We should choose p and q with equivalent sizes.

    It is advised that n is higher or equal to 512 bits
  - Take a number **e** which does not have any factor in common with

    (p-1) (q-1).
  - Calculate **d** such as **ed mod (p-1)(q-1) = 1** (We need to choose e randomly in order that e doesn't have any common factors with

    r=(p-1) (q-1))
- The couple **(e, n)** constitutes the public key.
- **(d, n)** is the private key.
- Various other rules are to be respected for the use of these prime numbers so that the algorithm cannot be "broken"

4.2.3.2 RSA: Operative rules

- We use the decimal codes of the characters to code (for example ASCII)
  - N.B. If we take other codes (ex: for instance hexadecimal), it is necessary to treat all calculations in the corresponding base
- We cut out the message in blocks which is composed of less figures than **n** (we add zeroes if it is needed to obtain the last block)

4.2.3.2 RSA: Example (1/4)

- https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html
- Let us start by creating our pair of keys:
  - Let us take 2 prime numbers randomly: **p = 127**, **q = 167**
  - Let us calculate **n = pq = 127 * 167 = 21209**
- We must choose **e** randomly such as **e** does not have any factor in common with r=**(p-1) (q-1)**:
  - **r=(p-1) (q-1) = (127-1) (167-1) = 20916**
- We need to find two numbers **e** and **d** whose product is a number equal to **1 mod r**. Below appears a list of some numbers which equal **1 mod r**.
- We must choose e randomly such as e does not have any factor in common with r=(p-1) (q-1):
- Candidates numbers (k.r+1), k app N* donc nbr mod r = 1 : 20917 41833 62749 83665 104581 125497 146413 167329
- 188245 209161 230077 250993 271909 292825 313741 334657
- 355573 376489 397405 418321 439237 460153 481069 501985
- 522901 543817 564733 585649 606565 627481

4.2.3.2 RSA: Example (2/4)

- Step 2. Find K a number equal to 1 mod r which can be factored:
  - Let's choose for instance K=20917
- Find the factors of K (see https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html)
  - 13*1609
- Let's take **e = 13**
- Let's choose **d** such as **13*d mod 20916 = 1**
  - We find **d = 1609**
- These are the keys:
  - The public key is **(e, n) = (13,21209)** (=enciphering key)
  - The private key is **(d, n) = (1609,21209)** (=deciphering key)

4.2.3.2 RSA: Example (3/4)

- https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html
- Let's encipher the message "**HELLO**". Let's take first the ASCII code (into decimal) of each character and one puts them end to end:
  - **m = 72-69-76-76-79**
- Then, it is necessary to cut out the message in blocks which is composed of less digits than **n**. **n** is composed of 4 digits, one thus will cut out our message in blocks of 3 digits:
  - 726 976 767 900
    (let's complete with zeros)
- Then one encrypt each one of these blocks:
  - 726^13 mod **21209** = 11600
    976^13 mod **21209** = 5705
    767^13 mod **21209** = 16590
    900^13 mod **21209** = 3565
- The encrypted message is **11600.5705.16590.3565**. One can decipher it with d:
  - 11600^1609 mod **21209** = 726
    5705^1609 mod **21209** = 976
    16590^1609 mod **21209** = 767
    3565^1609 mod **21209** = 900
- I.e. the digit suite: **726976767900**.
  We find the clear message: **72 69 76 76 79**: "**HELLO**".

4.2.3.2 RSA: Example (4/4)

- https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html
- Let's encipher the message "**HELLO**". Let's take first the ASCII code (into decimal) of each character and one puts them end to end:
  - **m = 72-69-76-76-79**
- Then, it is necessary to cut out the message in blocks which is composed of less digits than **n**. **n** is composed of 4 digits, one thus will cut out our message in blocks of 3 digits:
  - 726 976 767 900
    (let's complete with zeros)
- Then one encrypt each one of these blocks:
  - 726^13 mod **21209** = 11600
    976^13 mod **21209** = 5705
    767^13 mod **21209** = 16590
    900^13 mod **21209** = 3565
- The encrypted message is **11600.5705.16590.3565**. One can decipher it with d:
  - 11600^1609 mod **21209** = 726 (if 1 bit is corrupted 11601^1609 mod 21209 = 6051)
    5705^1609 mod **21209** = 976
    16590^1609 mod **21209** = 767
    3565^1609 mod **21209** = 900
- I.e. the digit suite: **726976767900**.
  We find the clear message: **72 69 76 76 79**: "**HELLO**".

4.2.3.2 Remarks on the RSA

- In practice, it is not so simple to program: Large prime numbers should be found (that can take very a long time to calculate)
- It is necessary to obtain prime numbers **p** and **q** which should be *really* random
- We should not use blocks as small as in the example before: it is necessary to be able to calculate powers and modulos on very large numbers
- In fact, one never uses the asymmetric algorithms to encrypt all the data, because it will take a too long time to calculate: one encrypts the data with a simple symmetric algorithm from which the key is drawn randomly, and it is this key which is exchanged through an asymmetric algorithm like the RSA

4.2.3.3 Other asymmetric algorithms

- gpg (GNU Privacy Guard)
- **ECC** (Elliptic Curve Cryptosystems, Encryption by Elliptic Curve). This system is based on a parametric curve which passes through a certain number of points with integer co-ordinates. It is not very developed yet, but it is promising
- **Diffie-Hellman** (Used in IKE (Internet Key Exchange) negotiations), more and more preferred than RSA. (Diffie-Hellman had quickly been adopted by the open-source community when RSA was not in the public domain yet)
- **El Gamal**: based on the calculation of discrete logarithms

4.2.4 Comparisons of the ciphering algorithms

Comparison

- Asymmetric ciphering more useful
  - No problem with the key transfer
  - Allow the message signature
  - Possibility to manage Public Key Infrastructure (PKI)
- Symmetric encryption is faster (La Recherche, June 2018)
  - AES allows enciphering many gigabytes per second on a recent processor
  - Asymmetric cryptography standards reach less than one megabyte per second (1000 to 10000 slower !)
- Generally the two strategies are combined
  - Interest: to use a protocol with public key to transmit the DES key => hybrid cryptography

Hybrid cryptography, generation of a sharing key Diffie-Helman strategy

- Two users will design a common key which will be useful for them only

ASYMMETRIC ASPECT

- They choose n the multiple of 2 prime numbers p and q and an integer a (a and n can be known (not confidential))
- Then each one chooses an integer X belonging to [1, n-1] and calculates the integer $Y=a^X \bmod n$
- We obtain two couples (X_1,Y_1) and (X_2, Y_2) where the values Y_1 and Y_2 will be published

HYBRID ASPECT

- Each one of them can then calculate the key $c=a^{X1X2} \bmod n$ because $c=(Y1^{X2} \bmod n)=(Y2^{X1} \bmod n)$
- R: Each one knows its own X only
- Security comes from the fact that it is impossible in a reasonable time to obtain the key C by the calculation of a discrete logarithm (unfeasible in a reasonable time taking into account the size of p and q)

SYMMETRIC ASPECT

- Users can now exchange encrypted data using a symmetric system with the common key c

Application

- User 1
- User 2

n

a

Application

- User 1
- X1 : private key

n

a

- User 2
- X2 : private key

Application

- User 1
- X1 : private key
- $Y1=a^{X1} \bmod n$: public key

n

a

- User 2
- X2 : private key
- $Y2=a^{X2} \bmod n$: public key

Application

- User 1
- X1 : private key
- $Y1=a^{X1} \bmod n$: public key
- Send Y1 to user 2
- Receive Y2

n

a

- User 2
- X2 : private key
- $Y2=a^{X2} \bmod n$: public key
- Send Y2 to user 1
- Receive Y1

Application

- User 1
- X1 : private key
- $Y1=a^{X1} \bmod n$: public key
- Send Y1 to user 2
- Receive Y2
- $c=(Y2^{X1} \bmod n)$

n

a

- User 2
- X2 : private key
- $Y2=a^{X2} \bmod n$: public key
- Send Y2 to user 1
- Receive Y1
- $c=(Y1^{X2} \bmod n)$

Application

- User 1
- X1 : private key
- $Y1=a^{X1} \bmod n$: public key
- Send Y1 to user 2
- Receive Y2
- $c=(Y2^{X1} \bmod n)$
- Key « c » in order to use the symetric system

n

a

- User 2
- X2 : private key
- $Y2=a^{X2} \bmod n$: public key
- Send Y2 to user 1
- Receive Y1
- $c=(Y1^{X2} \bmod n)$
- Key « c » » in order to use the symetric system

Private and public keys

- The public key is composed of two large prime numbers p and q (several hundreds of bits).
- The public key n is given by n =p*q.
- As n est very large, it is impossible to find all the possible factorisations.
- The knowledge of n does not allow to deduce the values of p and q.

Exercise

- Generate a shared key with your neighbor
- (ex:
  - a=3 et n=14 (public values (known)) n=2*7
  - X_1 = 4 (secret value known only by the participant on the left)
  - X_2 = 3 (secret value known only by the participant on the right)

Some considerations on breaking a 768-bit RSA key

- From an Inria document, 2010.
- Key used for bank cards
- To break the key, find the prime numbers which compose the key: it is a number composed of 232 figures (2^{768})…
- Need efficient algorithm
- Need large calculation capacities: use of Grid'5000 => 1544 computers with more than 5000 cores.
- Collaboration with CH, JP, NL, DE : on average 1700 cores used during one year of calculation…
- One week by using the supercomputer *Jaguar* (from *Oak Ridge National Laboratory*) if available (not such computers in Europe…)
- The purpose was to show if it is possible to break using grid of « classical » computers
- Next step: to break a 1024 bit-key => it should be possible around 2020
- Advise from ANSSI (2010):
  - Use at least 1536 bit-keys for applications until 2010
  - Use at least 2048 bit-keys for application beyond 2010

Other considerations regarding calculation time (2016)

- 2^{40}~10^{12} : operations possible on a personal computer
- 2^{56}~10^{16} : operations possible for a company or research lab
- 2^{64}~10^{19} operations possibly possible for NSA (US), GCHQ (GB), DGSI (FR)
- 2^{80}~10^{24} operations considered somewhat unfeasible
- 2^{128}~2.10^{38} operations impossible with current technologies
- 2^{256}~6.10^{76} almost physically impossible: a computer, perfect in the physical sense, would need the same level of energy as the sun for several years ...

These orders of magnitude will not be the same in 20 years (maybe about 2^{80} will be possible for a company ...)

Other considerations considering the size of the keys

- ”From "Randomness, the keystone of computer security": LA RECHERCHE, July-August 2019, D. Vergnaud
- Is perfect security just an illusion? In theory, no.
- Disposable mask encryption (invented by the American Gilbert Vernam in the early 20th century) shows that if you have a key as long as the message, drawn in a random and uniform way, and used only once, it is possible to achieve perfect security, i.e. the **view of a ciphertext does not reveal any information about the plain text to the attacker.** Since the key must be exchanged and never reused, the practical difficulties are immense for large-scale use of this method of cryptography.
- The unpredictability of keys remains essential to ensure the security of these modern algorithms.

4.2.5 Other applications of cryptography

4.2.5.1 Hashing

4.2.5.2 Signature

4.2.5.3 Hash for passwords

4.2.5.4 Certificates

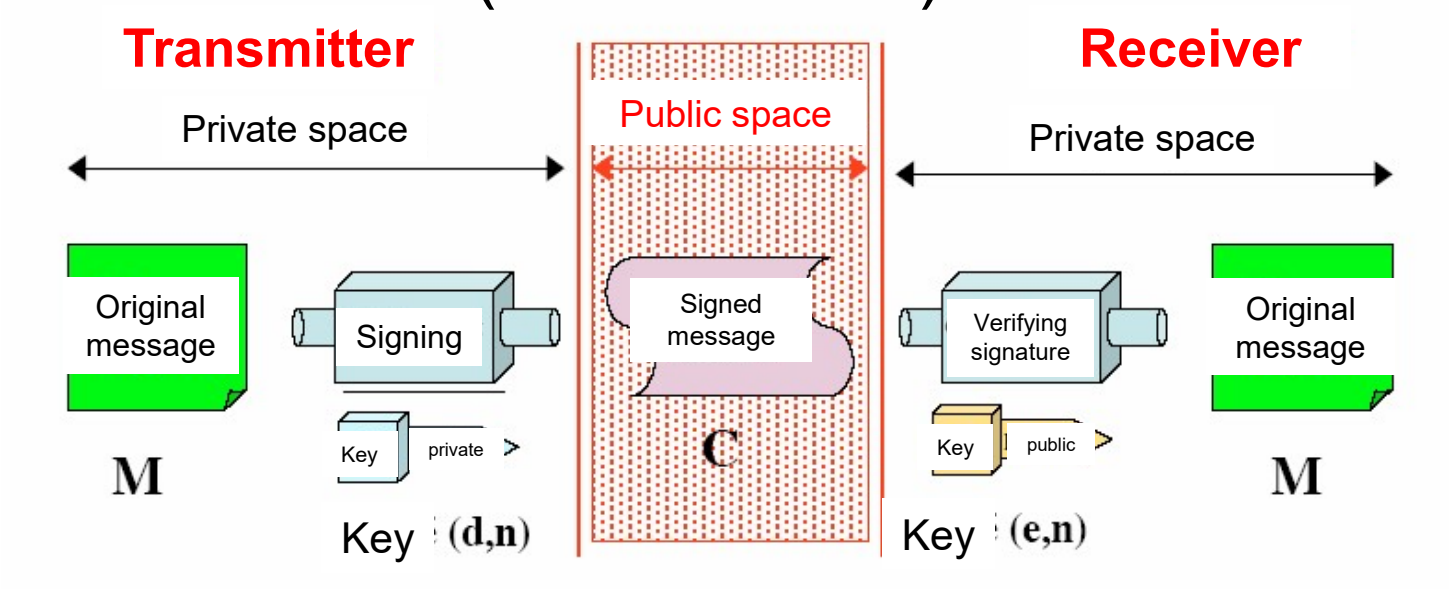
4.2.5.1 Hashing Functions

- How it works?
  - A hash is calculated by applying a mathematical algorithm to a set of data
  - The value obtained is generally shorter than the set of data
  - It is almost not possible to have two sets of data with the same hash (but it can be!)
  - It is impossible to obtain the data from the hash (non reversible function)
- Purpose
  - To guarantee the Authentication of the parts
  - To guarantee the integrity of the data
- Names
    - Hash (*fr. haché,* ***somme de contrôle***)
    - Fingerprint (*fr. empreinte*)
    - Digest (*fr. condensé*)

4.2.5.1 Algorithms for Hashing

- MD5 (message digest 5)
  - 1994 (by Ron Rivest), RFC 1321
  - irreversible print of 128 bits
  - allows to check the integrity of the message
  - http://www.isi.edu/in-notes/rfc1321.txt
- SHA-1 (Secure Hash Algorithm)
  - SHA-1 appeared in 1995
  - irreversible print of 160 bits
- SHA-2 (Last version of the standard: 2012)
  - Two functions: les fonctions, SHA-256 et SHA-512 (size of the hash), also truncated version of SHA 512: SHA-512/256 et SHA-512/224

4.2.5.2. Principles of signature (without hash)

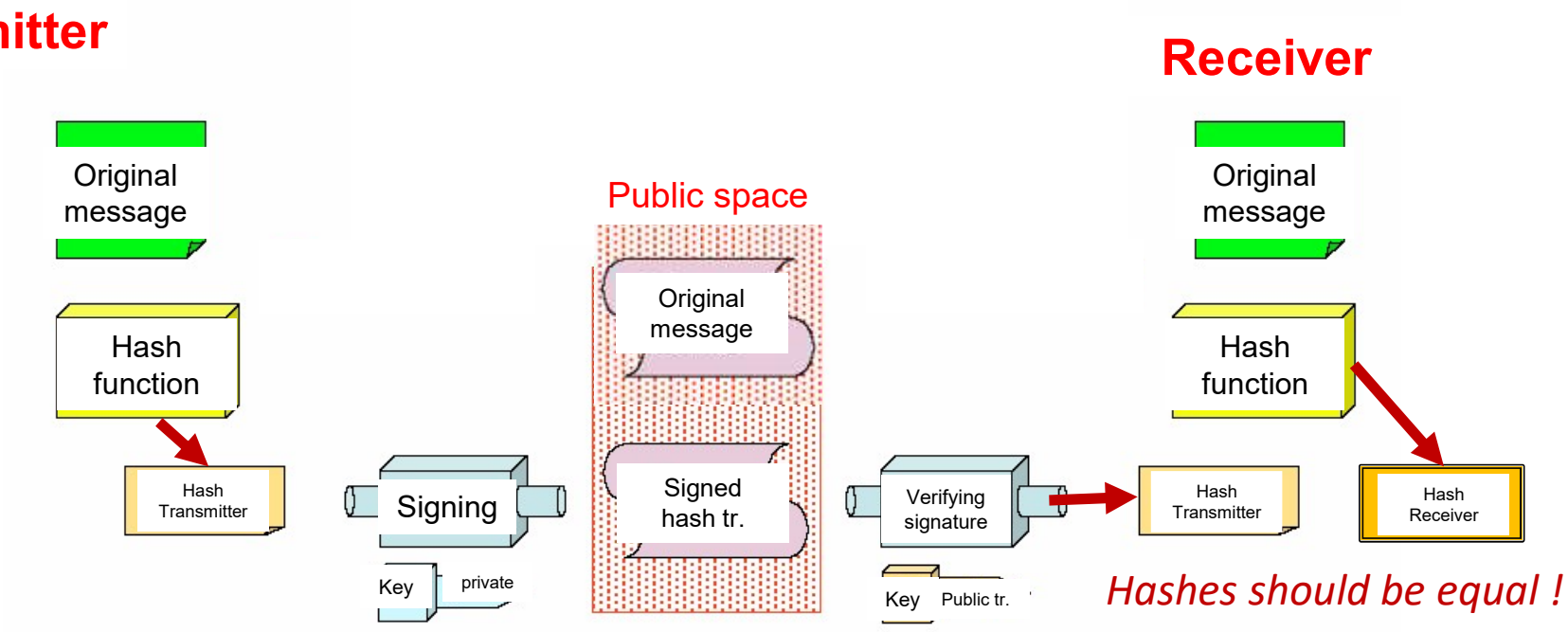


-to sign a message, the private key is used (the hash is encrypted with the private key)

-Garanty of authentication, but no confidentiality

- deciphering with the public key is a proof that the private key only was used for the signature

4.2.5.2. Signature (with hash)



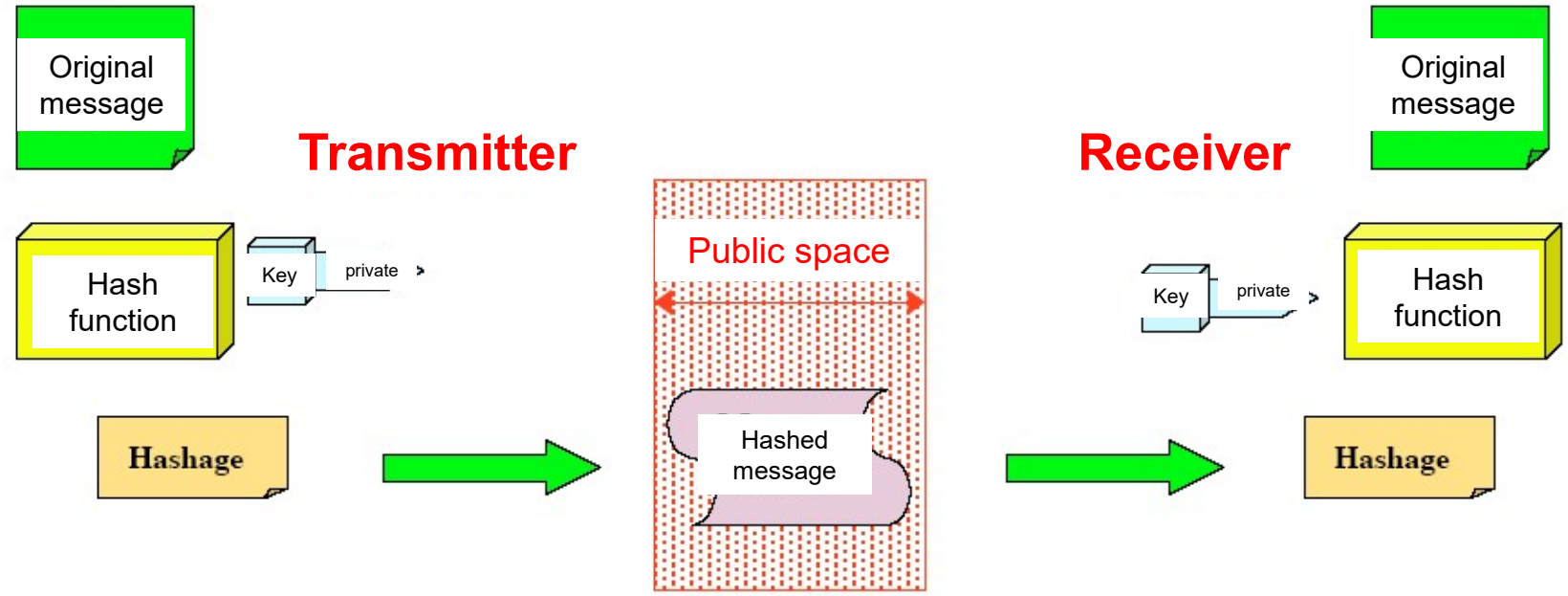
-It is not necessary to encrypt a complete message in order to sign it, it is enough to sign its hash only

-The robustness of the hashing procedure warrants that this is this document which has been signed

4.2.5.2 Authentication of the messages (1/2)

- If a symmetric ciphering is used to encipher the hash
  - It is an authentication, not a signature
  - Because the key needed to check the signature allows also to create it
  - If the key is known only by the two partners, it allows really to authentify the sender of a message.
- Ex: HMAC-SHA, HMAC-MD5

4.2.5.2 Authentication of the messages (2/2)



-For authentication, we do not encipher the hash, but use the key for calculation of the hash

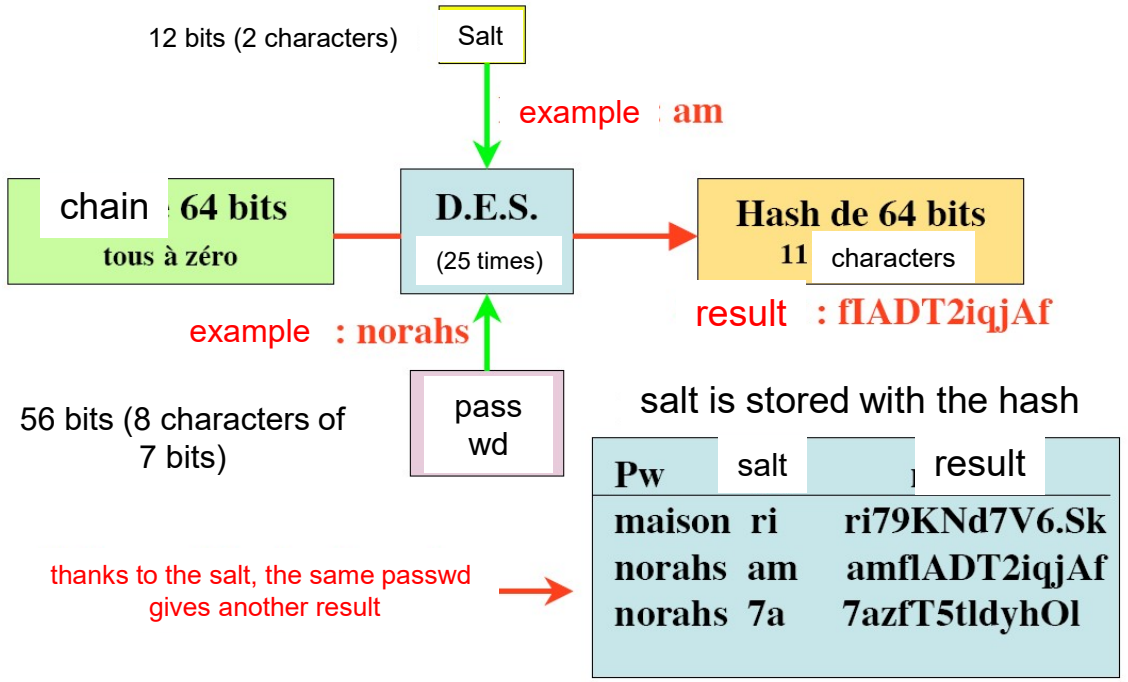
-Such a hash is called M.A.C. (Message Authentication Code)

4.2.5.3 Application of hashing: storage of passwords

- On a secure system, passwords are stored in an encrypted way (hash)
- By comparing the hash stored with the hash received (during the login) => the hashes have been created or not by the same passwords

4.2.5.3 Hash on a Unix system (½)

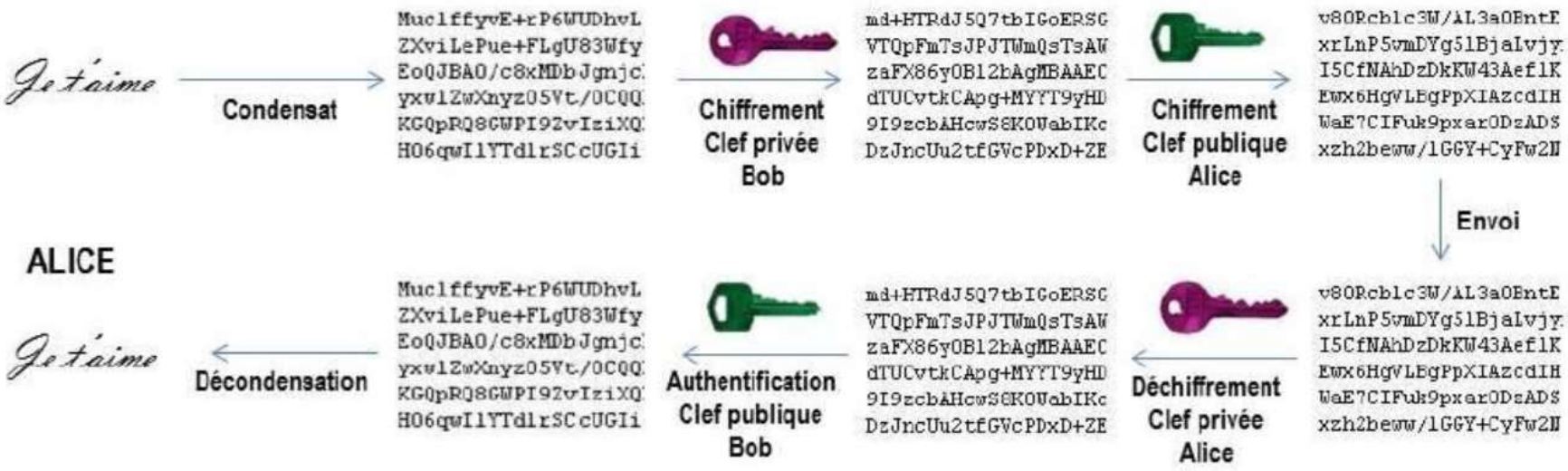
- Hash = encipher 25 times an empty character chain, with the password as a key
- For example, DES uses a 56-bits key, so 7 bits are taken from the passwd and the characters after the 8th are ignored
- A « salt » is added to avoid that two users using the same passwd (it could occurs) obtain the same hashes



4.2.5.3 Hash on a Unix system (2/2)

- Hash is generated thanks to the password and the « salt ». The generated hash is compared with the stored one
- When using a login through a network, hash and « salt » are obtained from a central server through a ciphered communication
- Storage
  - Before: logins and hash of the passwds in /etc/passwd => free reading access
  - Now: a specific file is used for the hash (can be read only by the administrator) => /etc/shadow
- In order to obtain /etc/shadow
  - Boot the computer with a disk or CD
  - Obtain the administrator passwd (exploit)

Chiffrement asymétrique



Ex Exhaustive research (attacks by brute force) of symmetrical keys

Knowing what the specialized machine “DES cracker” spends on average 4,5 days to find by an exhaustive research a 56 bits-key, how long will it take to find a 40 bits-key? a 112 bits-key?

Ex Exhaustive research (attacks by brute force) of symmetrical keys

Knowing what the specialized machine “DES cracker” spends on average 4,5 days to find by an exhaustive research a 56 bits-key, how long will it take to find a 40 bits-key? a 112 bits-key?

If it needs 4,5 days to decrypt 2^{56} keys, it means that for 2^{40} keys:

$$4{,}5 * \frac{2^{40}}{2^{56}} = 4{,}5/2^{16} = 4{,}5/65536 = 6{,}87.10^{-5}\ jours = 5{,}93s$$

And for a 112-bit key:

$$4{,}5 * 2^{112-56} = 4{,}5 * 2^{56} = 2{,}88.10^{17}\ jours = 7{,}89.10^{14}\ ans$$

If we consider the BigBang to be around 14.10^9 years,
How many Big Bangs? More than 56000 !!

Ex: **Symmetrical and asymmetrical ciphering (1/2)**

- A group of N people wishes to use a cryptographic system to exchange confidential information by pair of people. The information exchanged between two members of the group will not have to be able to be read by any other member. The group decides to use a symmetrical ciphering system.
- Which is the minimal number of symmetrical keys necessary?
- Give the name of a known symmetrical encryption algorithm.
- The group then decides to replace this system by an asymmetrical system.
- Which is the minimal number of couples of asymmetrical keys necessary so that each member can send and receive encrypted and/or signed information? If it is considered that each one can communicate with everyone, how many private and public keys each user will have it to hold (keep)?

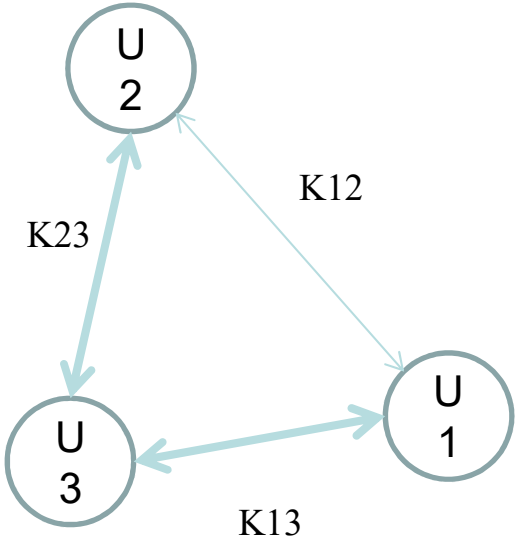
Ex: **Symmetrical and asymmetrical ciphering (2/2)**

- Bob wishes to send encrypted and signed information to Alice (Bob and Alice belong both to the group). Which key(s) Bob should use?
- Give the name of a known asymmetrical encryption algorithm.
- The group finally decides to use a hybrid system for the ciphering (i.e. which uses symmetrical and asymmetrical cryptography).
- Give the reasons why such a system can be efficient.

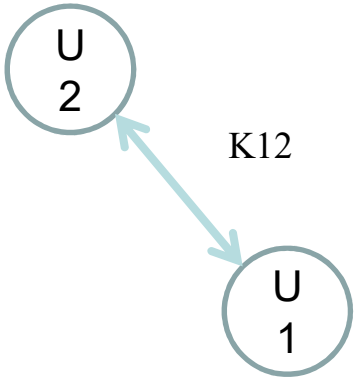
Ex: **Symmetrical and asymmetrical ciphering (1/2)**

- A group of N people wishes to use a cryptographic system to exchange confidential information by pair of people. The information exchanged between two members of the group will not have to be able to be read by any other member. The group decides to use a symmetrical ciphering system.
- Which is the minimal number of symmetrical keys necessary?
- See next slides
- Give the name of a known symmetrical encryption algorithm.
- For example DES, AES
- The group then decides to replace this system by an asymmetrical system.
- Which is the minimal number of couples of asymmetrical keys necessary so that each member can send and receive encrypted and/or signed information?
- n key pairs, each will have its secret key and will distribute its public key to all the others.
- If it is considered that each one can communicate with everyone, how many private and public keys each user will have it to hold (keep)?
- So everyone will hold the n-1 public keys of others + her/his own private key + her/his ownpublic key

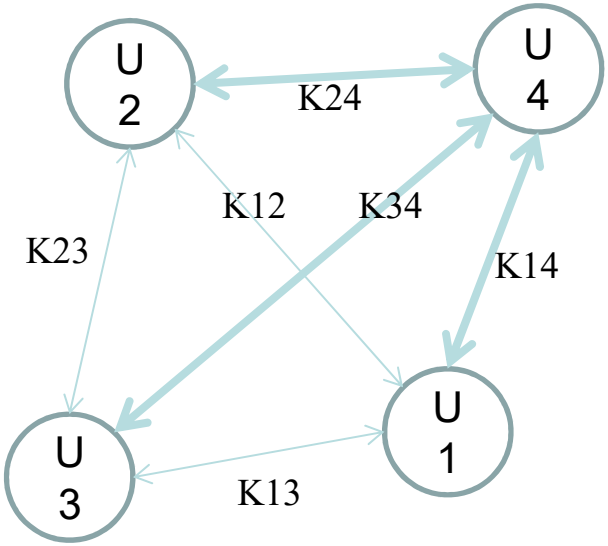
3 users, 3 keys



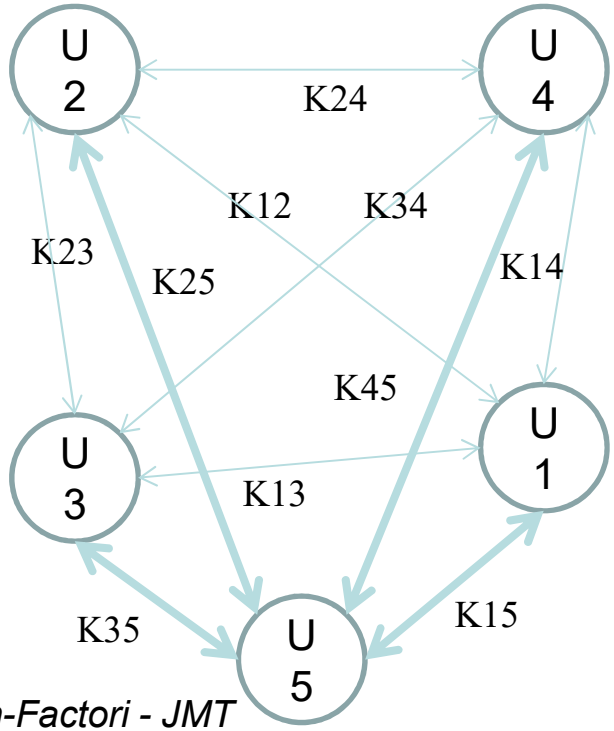
2 users, 1 key



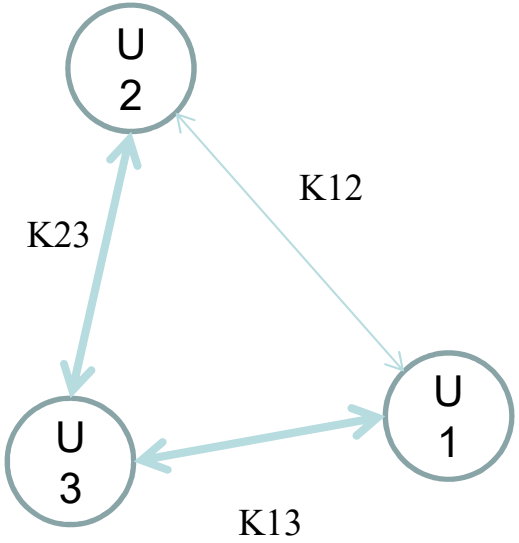
4 users, 6 keys



5 users, 10 keys

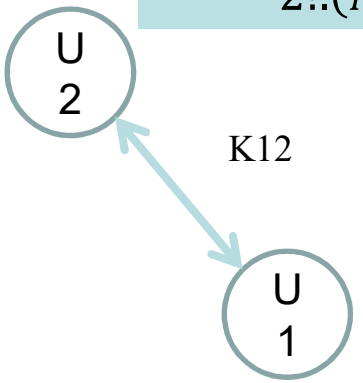


3 users, 3 keys

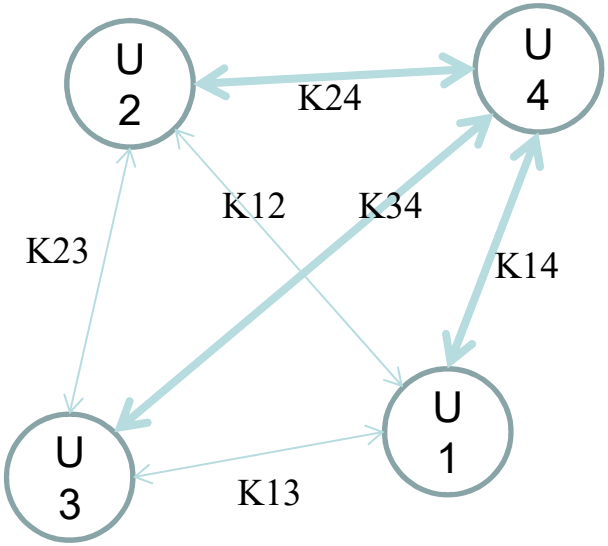


2 users, 1 key

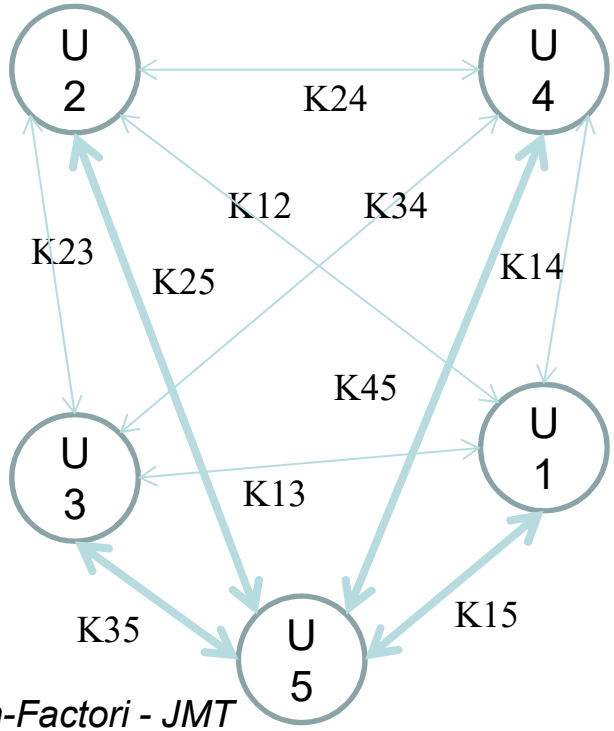
$$C_n^2 = \frac{N!}{2!.(N-2)!} = \frac{N.(N-1)}{2}$$



4 users, 6 keys



5 users, 10 keys



(rappels) le dénombrement

- Permutations
  - Tous les mélanges des objets
  - *Combien de jeux différents avec 32 cartes ?*
  - $32!=32*31*30*..*2*1=2,6.10^{35}$
- Arrangements $A_n^p=n!/(n-p)!$
  - Tirage d'une quantité d'objets dans l'ordre
  - *Combien de fois 3 chevaux dans l'ordre avec 10 chevaux au départ ?*
  - $10!/(10-3)! = 10*9*8=720$
- Combinaisons $C_n^p=n!/(n-p)!p!$
  - Tirage d'une quantité sans ordre
  - *Combien de possibilités de cinq numéros parmi 49 ?*
  - $49!/(49-5)!5!=(49*..*45)/(5*\ldots*1)=1906884$

Ex: **Symmetrical and asymmetrical ciphering (2/2)**

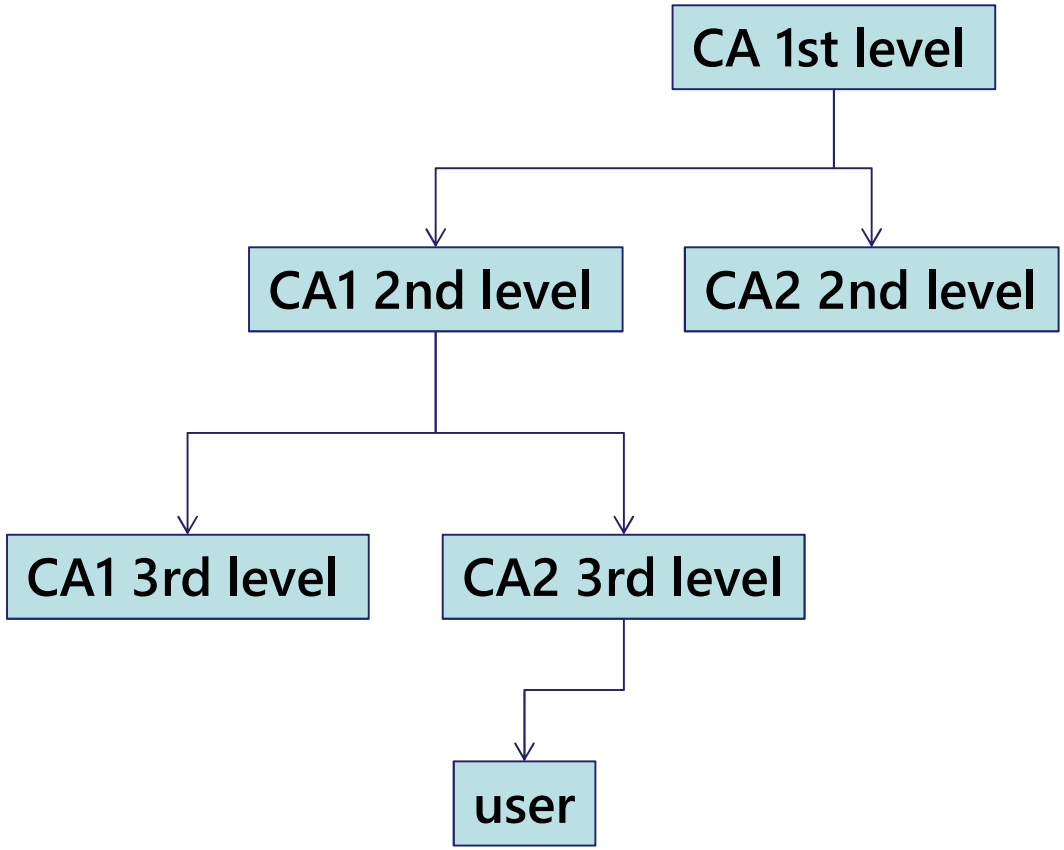
- Bob wishes to send encrypted and signed information to Alice (Bob and Alice belong both to the group). Which key(s) Bob should use?
- To encrypt, Bob will have to use Alice's public key. To sign, he will have to use his own private key.
- Give the name of a known asymmetrical encryption algorithm.
- RSA
- The group finally decides to use a hybrid system for the ciphering (i.e. which uses symmetrical and asymmetrical cryptography).
- Give the reasons why such a system can be efficient.
- To take benefit of both systems: fastness of symmetric systems coupled with an asymmetric aspect to guaranty the key exchange.
- The symmetric system should be used also if we want to digitally sign …

4.2.5.4 Certificates

4.2.5.4 Certificates

- Document which proves that a public key belongs to its owner
- It contains at least the following documents:
  - Public key
  - Identity of the entity owning the certificate
    - Name, firstname
    - IP address
    - E-mail address
  - Expiration date (not compulsory)
  - Signature of the certificate by a tierce/third party => this is a trusted partner (known company with good reputation)
  - Ex: https://premium.wpmudev.org/blog/ssl-certificate-authorities-reviewed/

Certification Authorities



4.2.5.4 Certification Authority (CA)

- CA creates and signs the certificates
- Each participant needs to present himself
  - Physical authentication of the participant
  - CA requests the participant to generate a public key-private key pair
  - CA creates a certificate with the identity of the participant, the CA public key, a expiration date and the signature of the CA
- With its certificate and the CA public key, the new participant can communicate with all the other participants certified by the same CA
- A CA can be private (company) or public
- A CA can ask another CA to certify its public key (hierarchy)
- Certificate format: (Open)PGP or X.509 v2 or v3 standards

4.2.5.4 Main parameters of a digital certificate according to X509v3 standard

1. Version of the certificate
2. Serial number
3. Algorithm used to sign the certificate
4. Name of the organization which managed the certificate
   - The couple serial number -name of the organization must be unique
5. Time of validity
6. Name of the owner of the certificate
7. Public key of the owner
8. Additional information concerning the owner or the ciphering mechanisms
9. Certificate signature
   - Signature and Algorithm and parameters used for the signature

4.2.5.4 Validation of the certificate

- To validate the received certificate, the customer must obtain the public key of the organization which created the certificate relating to the algorithm used to sign the certificate (field 3) and must decipher (verify) the signature contained in field 9.
- Using the information also contained in this field, the customer calculates the value of the digest (or *hash*) and compares the value found with that contained in the last field => if the two values correspond, the certificate is authenticated
- Then, the customer makes sure that the period of validity of the certificate is correct

4.2.5.4 Certificates directory

- To facilitate to access to certificate, CA proposes the use of a directory (LDAP, HTTP)
- To send an e-mail to User2, User1 ask the certificate to the directory, User1 does not need to be accessible (reachable)
- The directory can provide the certificate revoked list (CRL)
- A certificate can be revoked because it has been stolen, lost, or because its owner has moved in another company for instance

4.2.5.4 Limits of the certificates

- Various certification authorities exist
  - Impossible that there is only one CA (technical and strategic problems)
  - Interoperability of the authorities
  - Mutual recognition
  - Compatibility of the certificates and their validity
  - Limits inherent in the public key infrastructures (PKI)
    - Complexity, cost for an infrastructure deployment and management
    - High level of security necessary to the realization of the services
    - Validity, life duration, revokation of the certificates
  - Real lack of confidence of the users in the certification authorities which are generally outside the company and the services offered
    - Value of the certificates
    - Mechanisms and procedures of Authentication
    - Personal data protection, their identity, their transaction

4.2.6 Public Key Infrastructure (PKI)

4.2.6 Public Key Infrastructure (PKI)

- Setting of mechanisms necessary to the realization of asymmetric ciphering systems
  - PKI: Public Key Infrastructure
- Impossible to memorize all the public keys of all the potential correspondents of an Internet site
  - PKI = answers the need for knowing the keys in order to implement a public keys-asymmetric ciphering system

4.2.6 Function of a Public Key Infrastructure

- Generation of a single couple of keys (public key, private key)
  - Saving of the information necessary for its management
    - Filing (*fr. archivage*) of the keys
    - Procedure of covering in the event of loss by a user or of request for provision by the legal authorities
- Management of the digital certificates
  - Creation
  - Signature
  - Emission
  - Validation
  - Revocation
  - Renewal of the certificates
- Diffusion of the public keys to the resources which would request it and which would be entitled to obtain it
- Certification of the public keys (signature of the digital certificates)

4.2.7. Security protocols

4.2.7.1 IPv4 & IPv6, Security issues

4.2.7.2 security Protocols

- 4.2.7.2.1 Ipsec
- 4.2.7.2.2 SSL/TLS

4.2.7.3 Applications

- 4.2.7.3.1 VPN
- 4.2.7.3.2 RADIUS Server

4.2.7.1 IPv4 & IPv6, Security issues

4.2.7.1.1 IPv4

- Protocol: set of rules determining the format of the exchanged messages
- IPv4 (Internet protocol version 4, currently used) does not integrate any service for security
  - Does not allow the authentication neither of the source nor of the destination of a packet
  - Does not warranty the confidentiality of the data transported
  - Does not allow the confidentiality of implied IP addresses
- IPv4 is a protocol “without connection”, it does not guarantee:
  - The data has reached the destination (possible loss of data)
  - Delivery of data to the good destination
  - The correct scheduling (sequencing) of the data

4.2.7.1.1 IPv4

- No quality of service (no recovery after an error)
- => implementation of the TCP protocol (Transmission Control Protocol) at the 4th layer, TCP offers a reliable transport service in connected mode, but strictly speaking does not offer security services

4.2.7.1.2 IPv6

- The needs are taken into account in *IPnG* (Internet Protocol next Generation) or *IPv6*:
  - To handle a larger address range
  - To be able to make a dynamic allocation of bandwidth in order to be able to support multimedia applications
  - To take into consideration security aspects

Main IPv6 characteristics (RFC 2460)

- *(Recall: RFC = Request For Comment)*
- Support for a wide and hierarchically arranged addressing
- Addresses coded on 16 bytes (128 bits) instead of 4
- Representation in the form of hexadecimal numbers separated by two points every two bytes:
  - Example: 0123:: 4567:: 89ab:: cdef:: 0123:: 4567:: 89ab:: cdef
- Dynamic allocation of bandwidth for multimedia applications
- Creations of virtual IP networks
- Support procedures for authentication and ciphering
- Simplified headings of packets in order to facilitate and accelerate the routing

4.2.7.1.2 Difficulties for the use of IPv6

- Economic and technological problem for its deployment
- Modification of the address managements on internet
- Installation of systems supporting both IPV4 and IPV6 versions, synchronization of the migration of the versions

4.2.7.2 Security protocols

4.2.7.2.1 IPSec protocol

Security solution which is compatible with IPv4 and IPv6
Presentation
AH
ESP
IKE

IPSec strategies

4.2.7.2.1 IPSec (IP Security)

- IPSec is a set of protocols standardized by the IETF (Internet Engineering Task Force) which allows to ensure a data protection (IP layer of the TCP/IP model)
- The protection proposed by IPSec is based on cryptographic services and provides the following functions:
  - *"Anti-re-reading"*: an IP packet protected by IPSec and intercepted by a pirate could not be re-used in order to establish a new session
  - *Confidentiality*: enciphering of the data encapsulated in an IP packet in order to make sure that they cannot be read during their transfer
  - *Authentication*: allows to ensure that a received data comes from the expected IP host (with which the IP security was negotiated)
  - *Integrity*: the data were not modified during their transfer

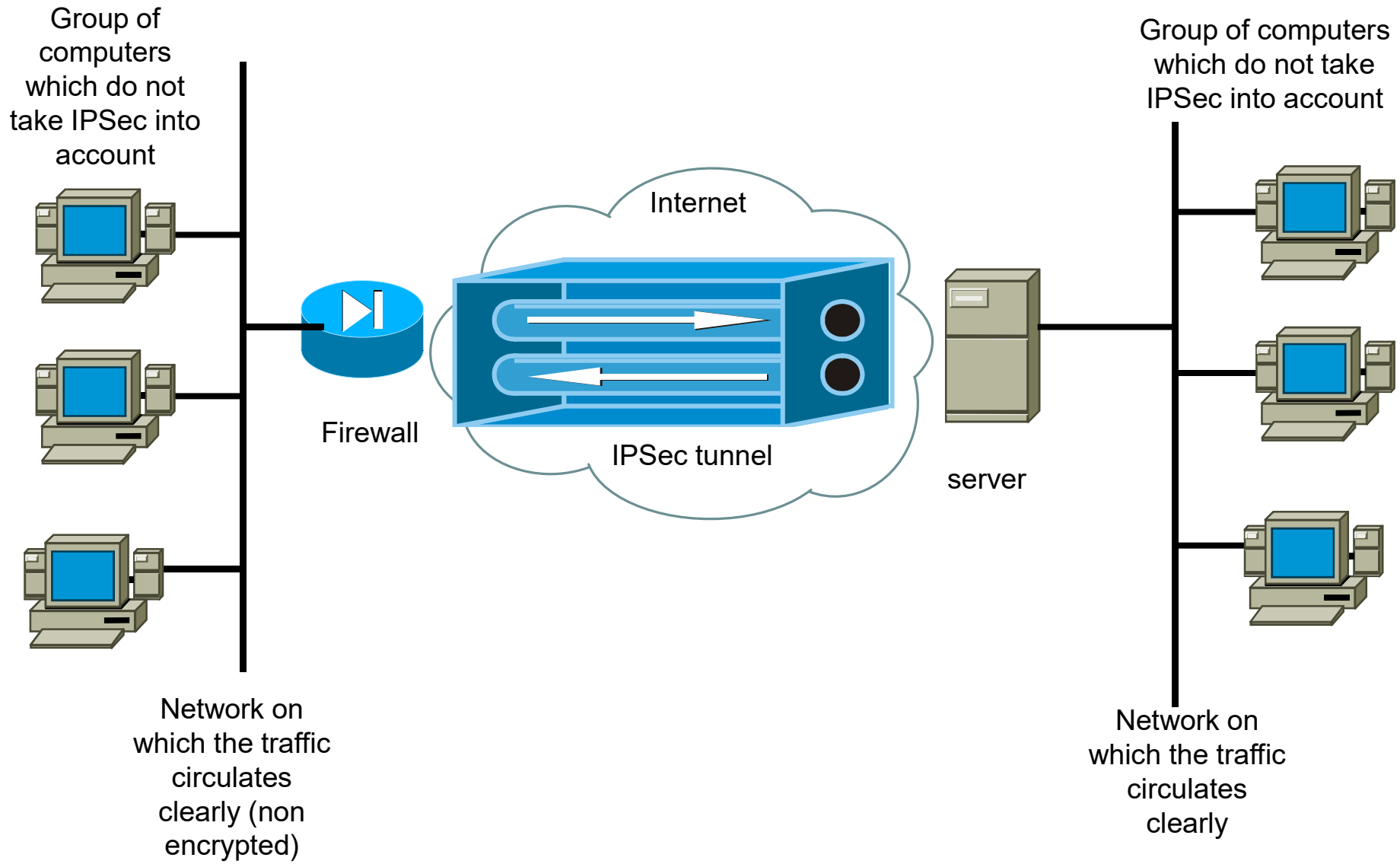
4.2.7.2.1 IPSec *Transport mode*

- Transport mode makes it possible to apply a security by IPSec from beginning to end
- Source and destination are the hosts taking charges of IPSec
  - Communication are safe from beginning to end
  - Blocking of certain types of traffic when the destination is open ports on a computer we would like to protect
    - Ex: a sensitive computer can have an IPSec strategy authorizing only a specific computer to reach this application, and blocking all the others

4.2.7.2.1 IPSec *Tunnel mode*

- Establishment of protected connections between two networks, when the gateways (firewall, router) are not able to use VPN technologies
- These are the gateways between the private network and the public network (Internet) which take the IPSec into account. The source and destination computers are not directly concerned

IPSec Tunnel mode, example



4.2.7.2.1 AH (Authentication Header) 1/3

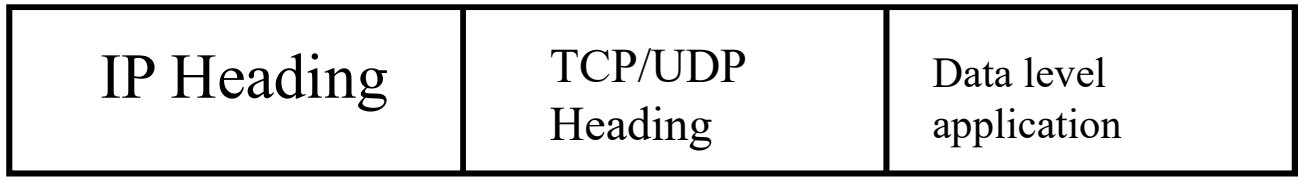
- Ensure the authentication, the integrity control and the anti-re-reading of the data encapsulated in an IP packet, as well as the IP heading itself

  ⇒ It is so a protection against attacks using IP headings (ex: IP-spoofing)

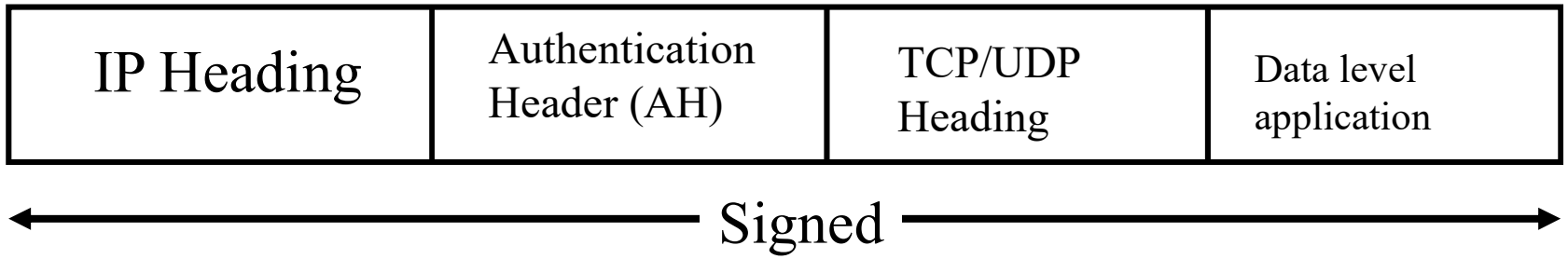
- N.B.: the integrity control does not take into account the bits of the IP heading since it is possible to modify them during their transit (ex: TTL field (lifespan) decremented when crossing a router)

4.2.7.2.1 AH 2/3

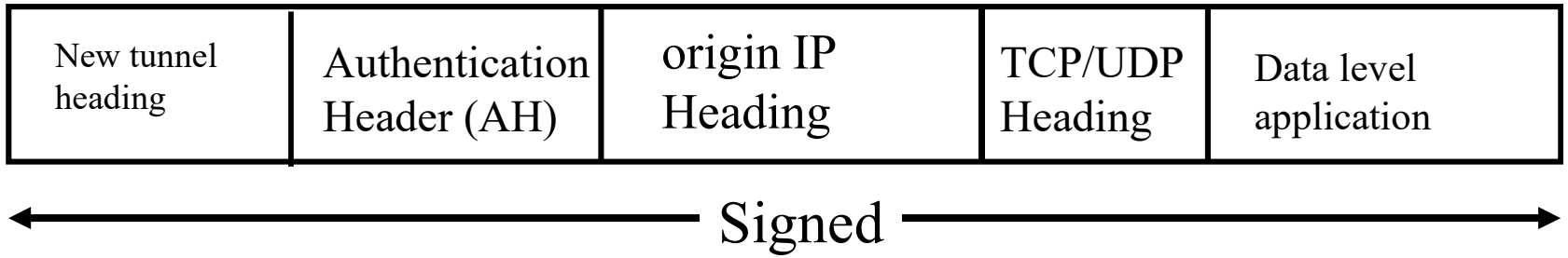
IPv4 packet



AH packet in transport mode



AH packet in tunnel mode



AH 3/3

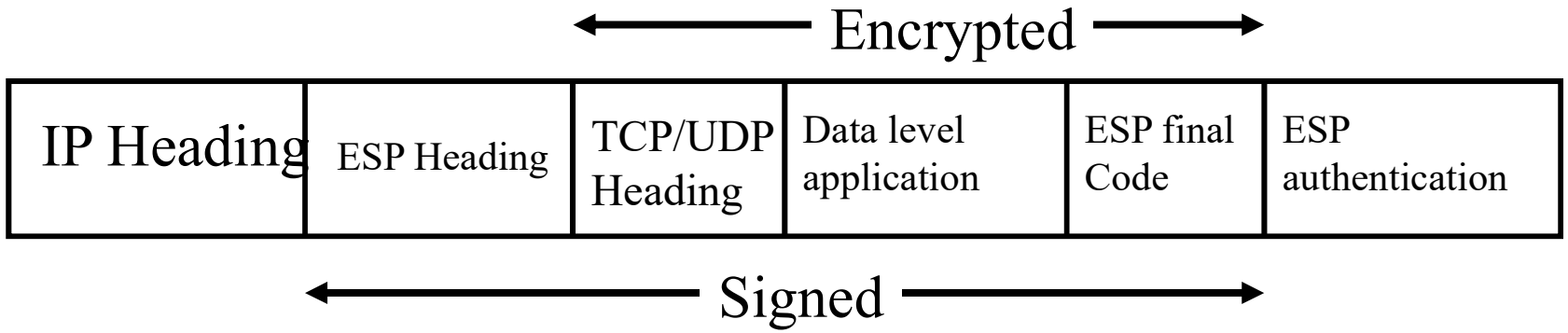
- AH uses the hashing algorithms according to
  - MD5 (Message Digest 5): 128 bits-hash
  - SHA1 (Secure Hash Algorithm): 160 bits-hash
- AH is defined in the RFC 2402

ESP (Encapsulating Security Payload)

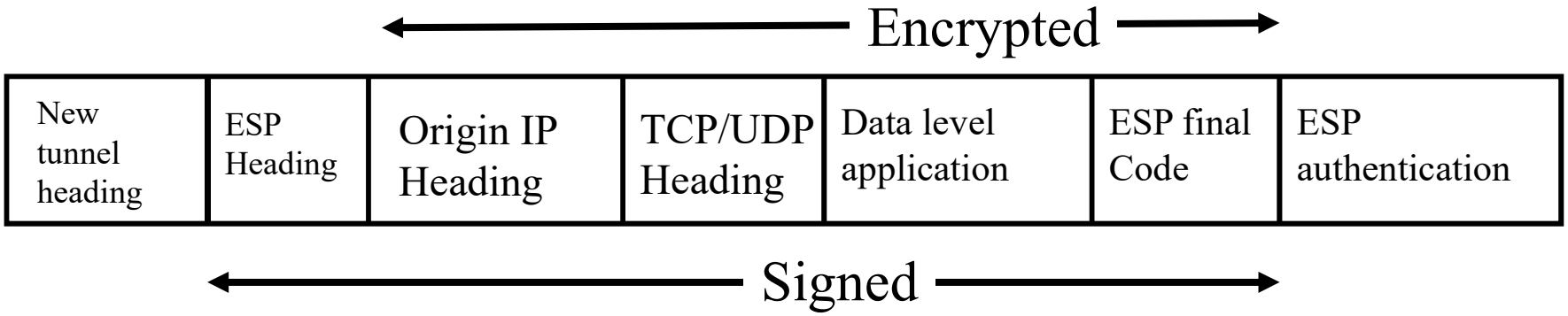
- Protocol ensuring the data confidentiality, by enciphering the contents of IP packets
  - N.B.: the headings are not encrypted, in order to be able to cross the routers!
- ESP can ensure an control integrity and an authentication, but only for the data encapsulated in IP packets
- ESP Protocol adds a heading in the IP packet

ESP

- Format of an ESP packet in transport mode



- Format of an ESP packet in tunnel mode



ESP

- Use the following hashing algorithms
  - MD5
  - SHA1
- Use the following encryption algorithms
  - DES
  - 3DES
- ESP is defined in RFC 2406

IPSec

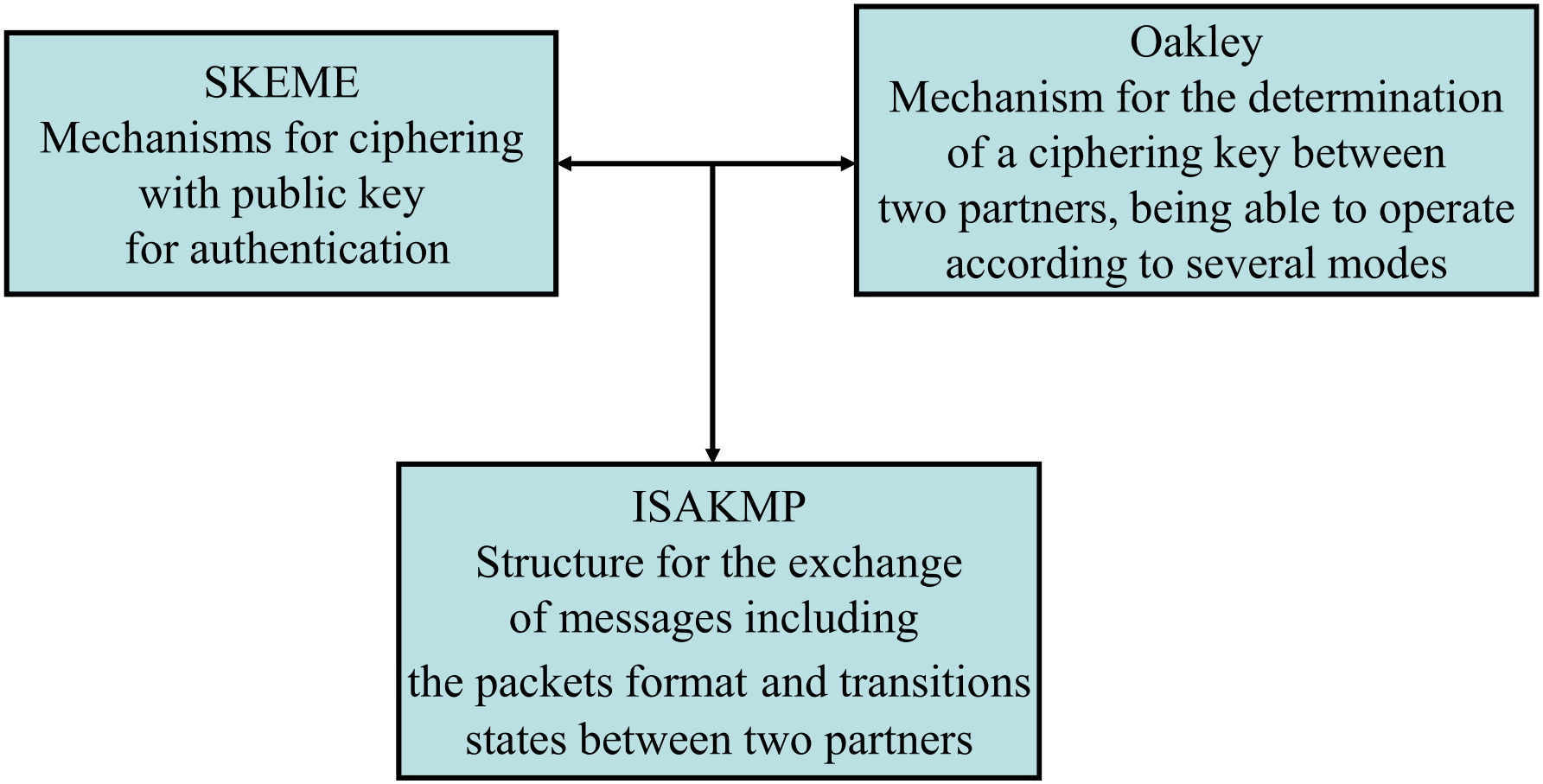
- In its most complex form (the most protected but also most consuming resources), an IPSec packet can use at the same time AH and ESP
- AH is an important consumer of CPU resources
- It is thus generally advised to use ESP alone, except if the integrity of the IP heading is a major element of the security policy
- There is hardware (accelerator cards) for IPSec implementation

Enciphering key management

- Oakley et SKEME (Secure Key Exchange Mechanism) : describes the way to exchange keys and defines services used by each exchange (based on Diffie-Hellmann exchange algorithm (RFC 2412)
- ISAKMP (Internet Security Association and Key Management Protocol) : this RFC (RFC 2408) defines procedures and packet formats to establish, negociate, modify, finish or cancel a security association. Formats are independent from key exchange protocols, from enciphering algorithms and from authentication mechanisms
- IKE (Internet Key Exchange) (RFC 2409) is an implementation of ISAKMP. IKE allows the realisation of key exchanges (authenticated keys) and the negociation of security services for security association

Protocols concerned by IKE

- IKE (Internet Key Exchange) (RFC 2409) is a hybrid protocol



IPSec strategy

- Set of parameters allowing to define how IP security must be applied to a data flow, and how the ciphering keys are generated
- One or more rules, each defining some filters, a method of authentication and filters actions

Default IPSec strategies

- Client (“simple response” strategy / *en réponse seule*)
  - Allows to forward the traffic normally, only one rule “default response rule” / “*règle de réponse par défaut*”, allowing to negotiate IPSec traffic if the distant host proposes it
- Server (ask for security / *demander la sécurité*)
  - Rule 1: negotiation for the entering and leaving IPSec traffic; if the distant computer does not use IPSec => non-protected communication
  - Rule 2: transmission of ICMP traffic without security negotiation
  - Rule 3: “default response rule” / “*règle de réponse par défaut*” (see above)

IPSec default strategies

- Server (requires security / *nécessite la sécurité*)
  - Rule 1: negotiation for the entering and leaving IPSec traffic; if the distant computer does not use IPSec => stopped communication
  - Rule 2: transmission of ICMP traffic without security negotiation
  - Rule 3: "default response rule" / "*règle de réponse par défaut*" (see above)
- "Default response rule" / "*règle de réponse par défaut*"
  - Allows to negotiate security with any host wishing to communicate in a protected way

Examination of the interaction between the rules

Direction of the traffic 	**No strategy**	**Client strategy** (simple response)	**Server strategy** (ask for security)	**Server strategy** (require security)
No strategy	Non-protected	Non-protected	Non-protected	No communication
Client strategy (simple response)	Non-protected	Non-protected	Secured	Secured
Server strategy (ask for security)	Non-protected	Secured	Secured	Secured
Server strategy (require security)	No communication	Secured	Secured	Secured

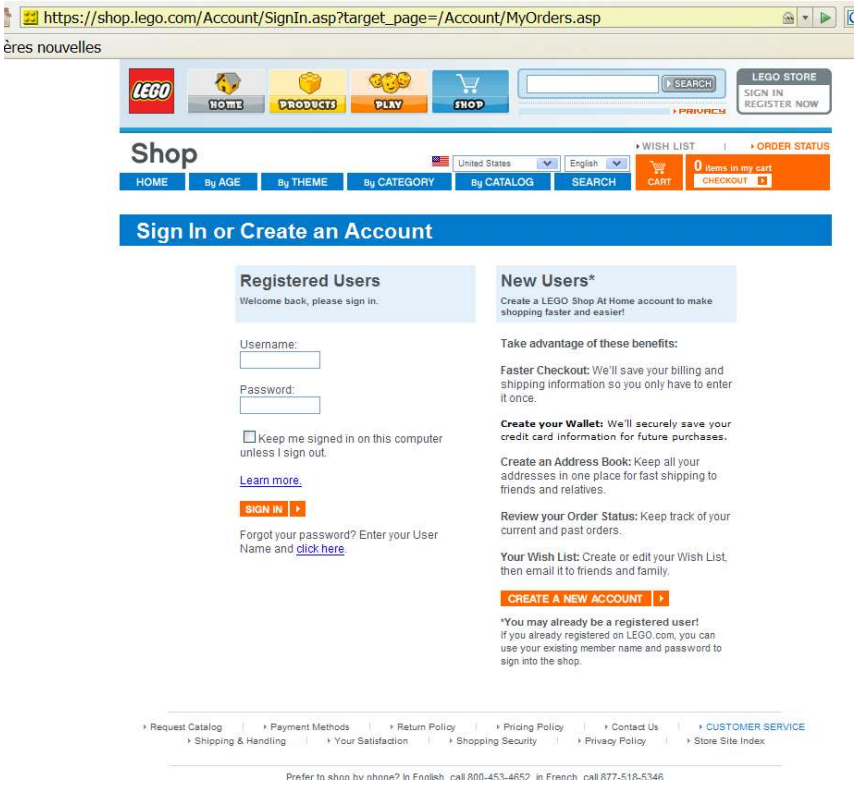
4.2.7.2.2 SSL/TLS

SSL (Secure Socket Layer)

- Data enciphering within the network protocol
- Guarantees
  - Identity of both parts
  - Data confidentiality from beginning to end
  - => data enciphering thus impossibility to read user names and passwords
  - data Integrity by the use of hash
- Generally based on TCP/IP
  - Allowing to guarantee the good arrival and the good schedule of data
- Web browsers equipped with SSL
  - Firefox, Konqueror, Internet Explorer, Safari…

SSL protocol

- SSL appeared in 1994 in Mosaic
- Web pages using SSL: HTTPS
- 1996: work for the formalization and standardization of SSL by the IETF



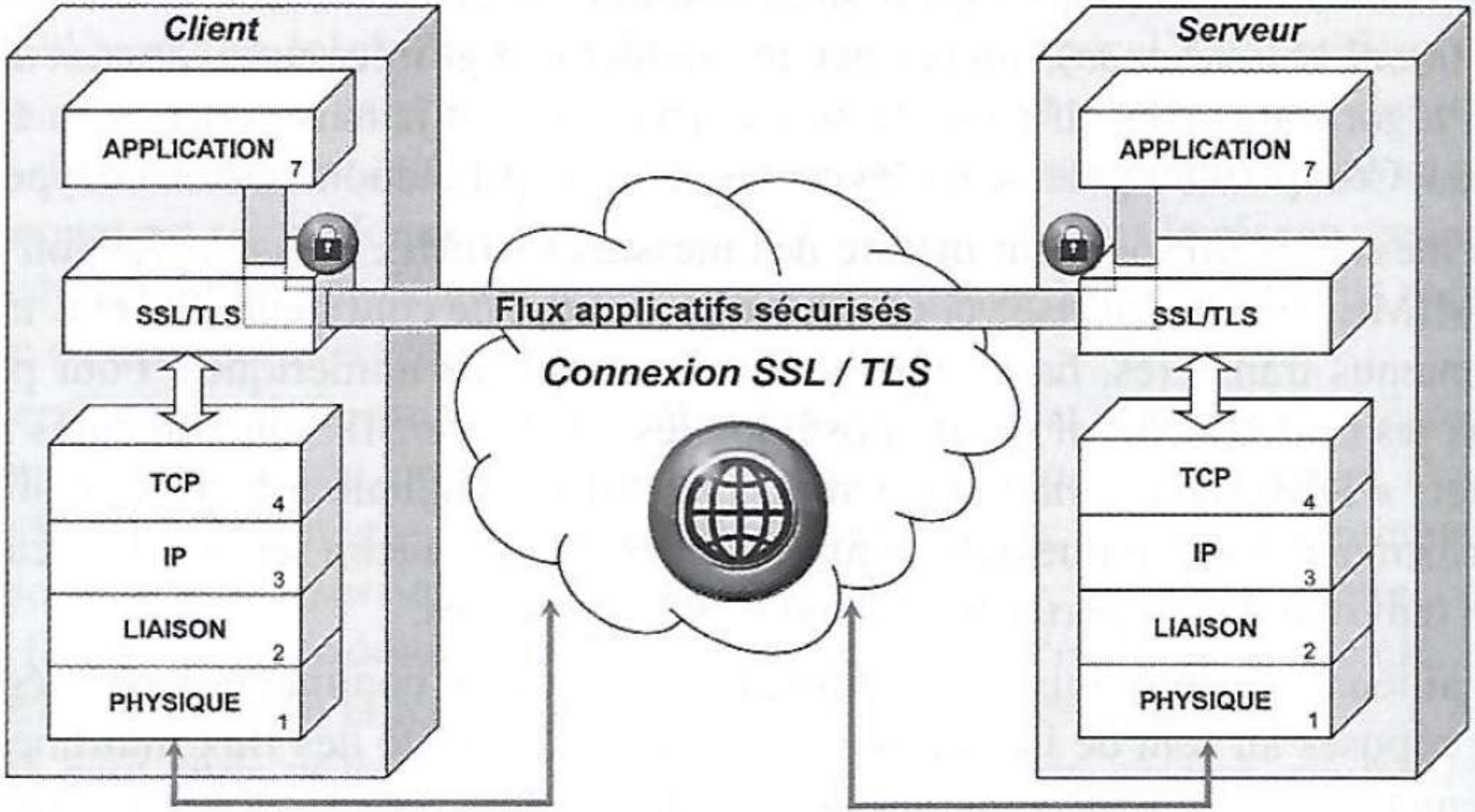


Figure 9.1 – La sécurité des flux applicatifs par SSL.

SSL: properties

- authentication
  - Proof of the client identity by certificates exchange
- Confidentiality
  - Encryption of the data by the use of a shared key and via the negotiation of the encryption algorithms
- Authentication and confidentiality phases take place during the stage of “negotiation”, also called “initialization” of SSL session
- Integrity
  - SSL checks that the data were not modified

SSL: facility of use

- Designed to be transparent for the end-user
  - The user needs only to be connected to the desired address (ex: https://...)
- RFC 1738 specifies the format
  - Web server port 80 but in SSL port 443
- A VPN based on SSL is easier to maintain than a VPN based on IPSec

Implementation of SSL

- A central server
- The client uses a communication software able to "speak" SSL
  - Browser
    - can use HTTPS
    - contains natively root SSL certificates coming from recognised certification authorities
- Possibilities to download additional software for the client computer
  - Plug-ins
  - Applets
- SSL may also be deployed on specific hardware solutions

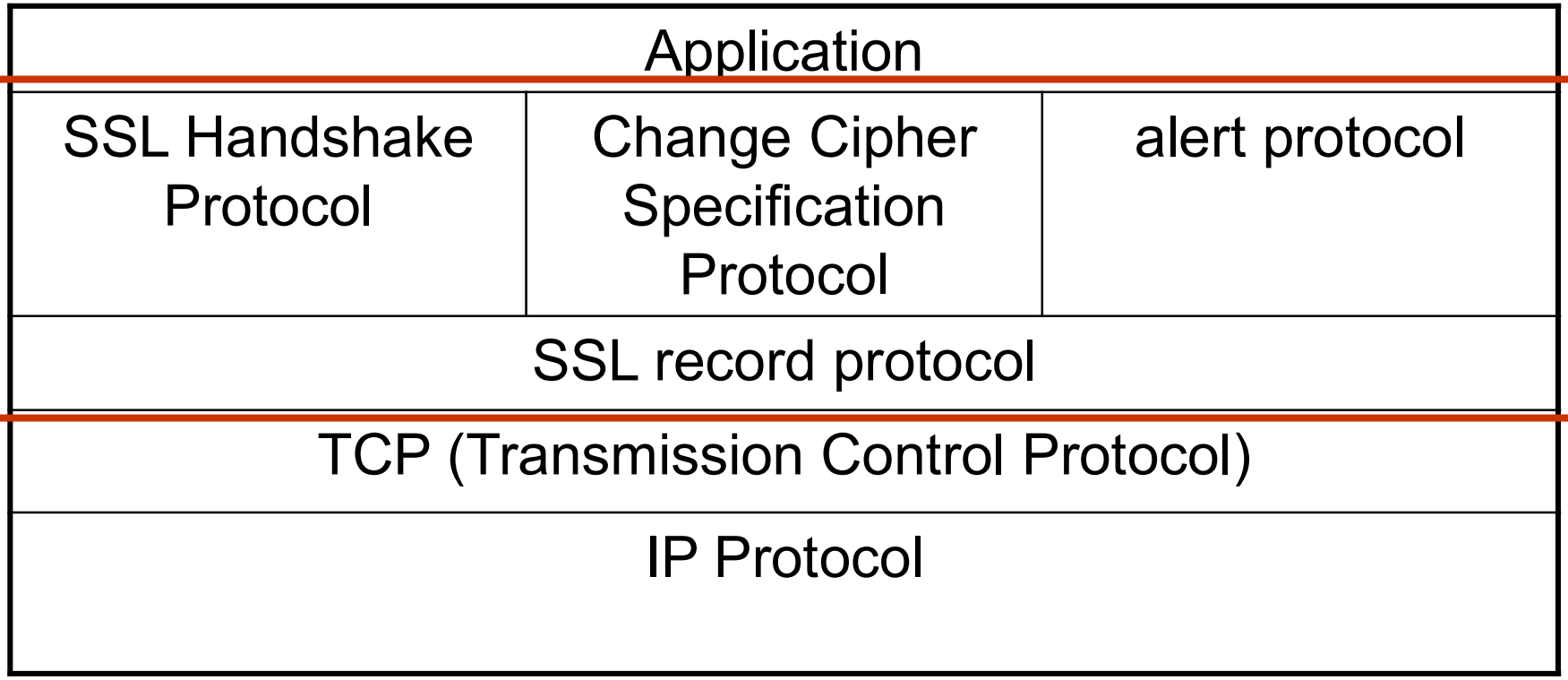
SSL: cryptography

- Symmetrical cryptography for data protection
  - Common Key (session key)
- Asymmetrical cryptography for the exchange of the session key

SSLv3

- More advanced version of SSL
  - Generator of keys
  - Hashing functions
  - Encryption algorithms
  - Management of the certificates
- Properties
  - Protocol for the change of specification: possible modification of the encryption algorithm during the communication to guarantee the confidentiality
  - Alert protocol allowing to send the alerts, accompanied by their importance (ex: unknown certificate, revoked, expired). High level alerts may cause the stop of the communication
  - Handshake protocol
    - authentication of the server by the client
    - negotiation of the protocol version
    - selection of the encryption algorithms
    - use of public-key encryption techniques for the distribution of secret keys
    - establishment of enciphered SSL connections
  - SRP recording Protocol (SSL Record Layer): encapsulation of the protocols located just above, like the Handshake protocol

Structure of the SSLv3 protocol



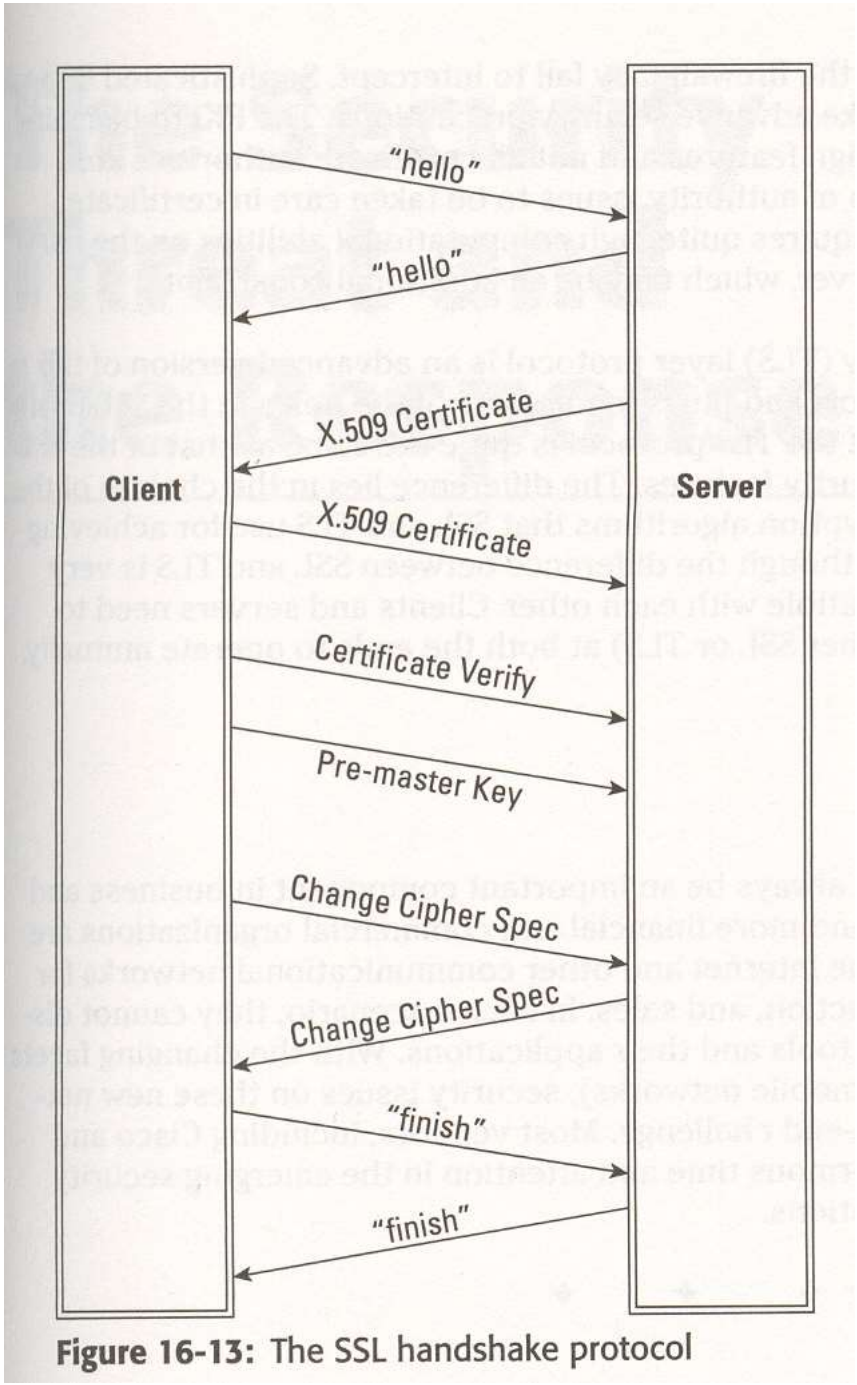


Figure 16-13: The SSL handshake protocol

SSL (Secure Socket Layer) et TLS (Transport Layer Security)

- SSL v2 and v3 obsolete since 2014 because of sevral security vulnerabilities
- TLS: new version, different algorithm, same functionalities (TLS ~ SSLv3)
- New version TLS 1.3 (June 2018) : abandonment of obsolete enciphering algorithms (MD5, SHA-224) to use Chacha20, Poly1305, Ed25519, x448 et X25519.
- BE CAREFUL about server configurations which allows retro-compatibilities with obsolete versions of cryptographic software (TLS 1.3 also forbids “downgrading”)
- TLS 1.3 is faster than previous versions
- Encryption of data within the network protocol
- Guarantees
  - Identities of both parts
  - End-to-end Confidentiality of data
  - => encryption of data so impossibility to read passwords and user names
  - Data integrity by using hash
- Is based generally on TCP/IP
  - Allows guaranteeing data order and controlling data arrival
- Web browser are equipped
  - Firefox, Konqueror, Internet Explorer, Safari, Chrome…

SSL/TLS

- Applications
  - Electronic Commerce
  - Communications Security with HTTPS
  - FTPs
  - Protected Copies
  - SSH
  - …

SSL/TLS Implementations

- OpenSSL the most widespread
- SChannel for Microsoft
- Secure Transport for Apple
- NSS for Mozilla and Chrome
- Cryptlib for banking
- GnuTLS for Open Source projects
- JSSE is an extension for Java applications to benefit SSL/TLS services
- MatrixSSL allows SSL/TLS services for embedded systems
- mbedSSL (previously PolarSSL), bought by ARM, for embedded systems

SSL/TLS Implementations

- OpenSSL the most widespread
- SChannel for Microsoft
- Secure Transport for Apple
- NSS for Mozilla and Chrome
- Cryptlib for banking
- GnuTLS for Open Source projects
- JSSE is an extension for Java applications to benefit SSL/TLS services
- MatrixSSL allows SSL/TLS services for embedded systems
- mbedSSL (previously PolarSSL), bought by ARM, for embedded systems

4.2.7.3 Applications

4.2.7.3.1 virtual private networks (VPN)

4.2.7.3.2 RADIUS servers

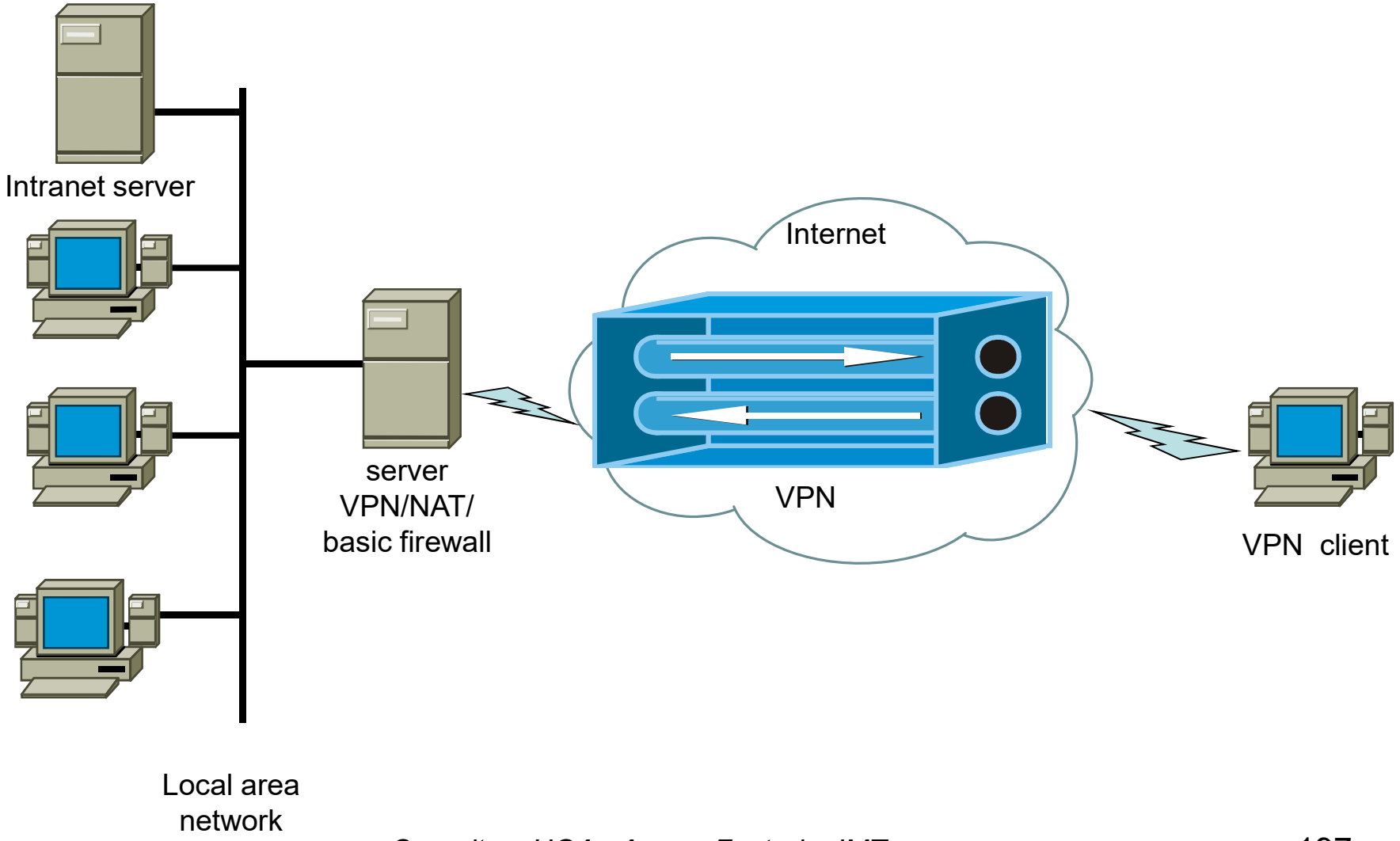
4.2.7.3.1 Some considerations about VPN

- VPN and distant access
  - To allow users located physically out of the corporate network of being able to connect itself to the corporate network
- VPN are considered as a particular class of shared networks
  - Resources of a real network shared between several sub-networks
- management Information to be taken into account
  - Topology: determination of the access points towards the sites which must be inter-connected by the VPN
  - Addressing: localization of the access points and the sites which must be inter-connected by the VPN
  - Routing: possibility of reaching the sites of the VPN
  - security Information: establishment and activation of the filters allowing or not the packets to cross them
  - quality of service Information: parameters for the control of the resources necessary for the quality of service

4.2.7.3.1 Some considerations about VPN

- level 2 VPN (frame): ex: VPN composed of Ethernet networks
- level 3 VPN (packet): ex: VPN composed of IP networks => the most widespread at the company level because integrating all IP functionalities
- level 7 VPN (application): ex: VPN set up for an application, such as HTTP
- VPN = extension of the ***private network*** of the company, ***virtually*** by the means of the public network
- ! The concept of security misses the basic definition of the VPN! => must be studied in particular (ex: use of IPSec, SSL)

4.2.7.3.1 VPN server in the periphery of the network (1/2)

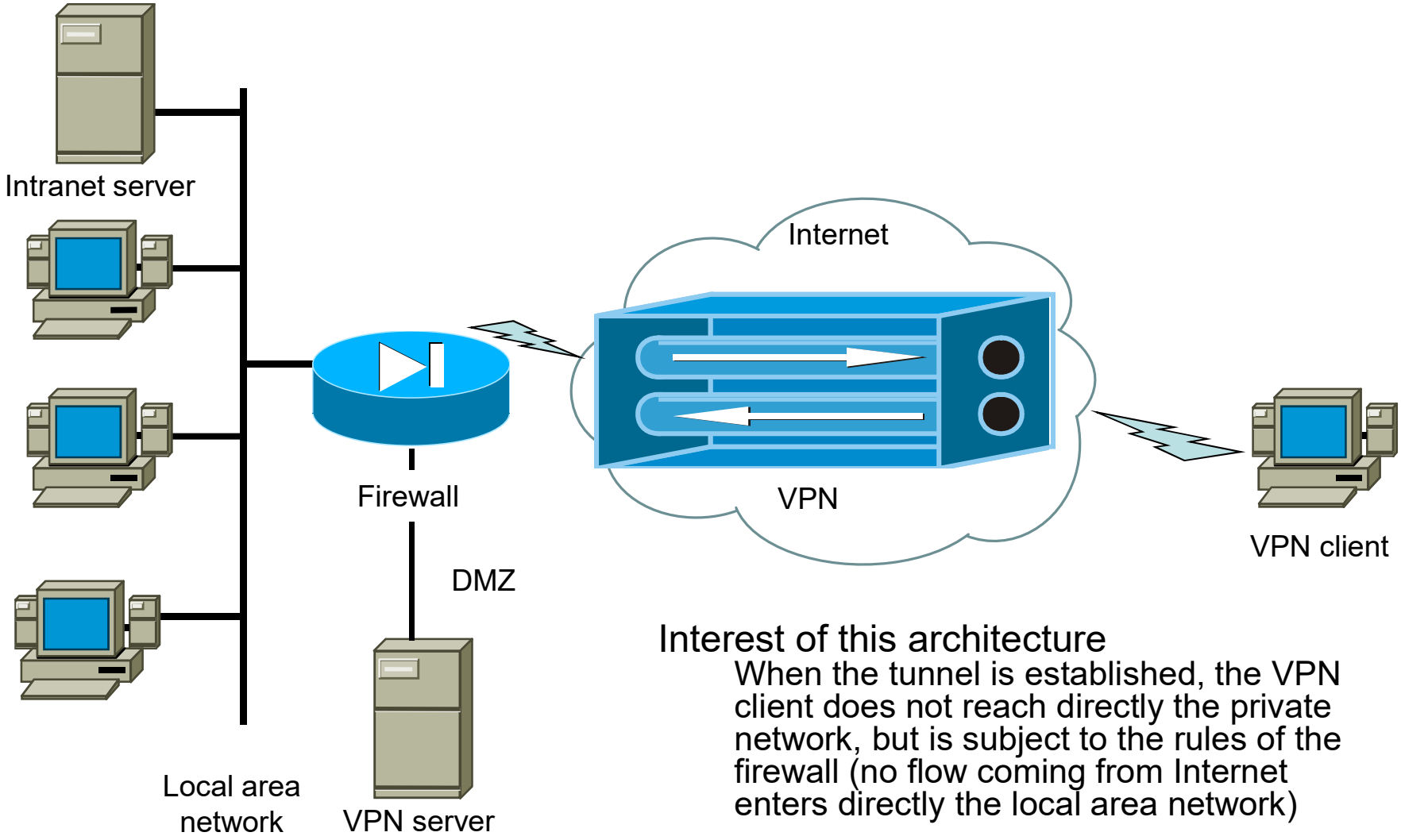


4.2.7.3.1 VPN server in the periphery of the network (2/2)

Necessary Pre-requisites for the installation of a VPN server

- VPN server must be connected to Internet (generally via the supplier of access Internet)
- VPN server must have a fixed IP public address or a corresponding DNS name
- VPN server must comply with the basic security rules
  - De-activate all the useless services on the server
  - Activate a firewall on the server
  - Use complex passwords strategies, or a “strong” authentication (smart card, biometric recognition)
- Check that the supplier of Internet access does not apply a filter to the router which connects you to Internet, and that the internet subscription allows to make flows enter

4.2.7.3.1 VPN server in a DMZ



Interest of this architecture

When the tunnel is established, the VPN client does not reach directly the private network, but is subject to the rules of the firewall (no flow coming from Internet enters directly the local area network)

4.2.7.3.1 Other aspects about VPN

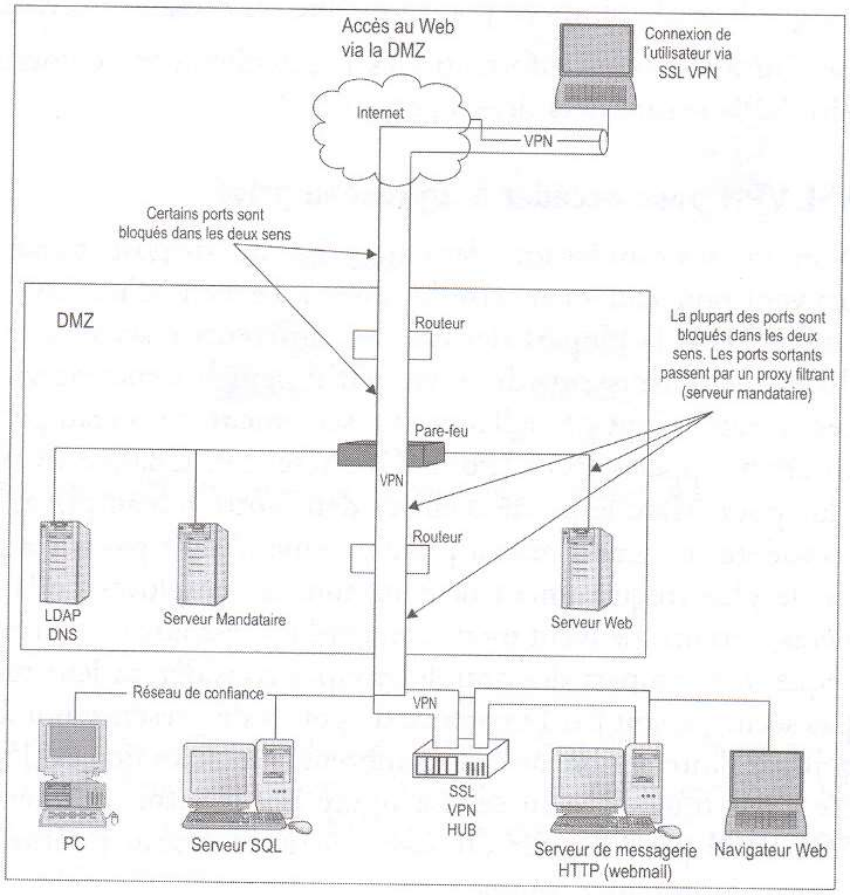
- It is possible to inter-connect two private networks through VPN
  - distant access VPN: relation between two sites of a company
  - intersite VPN : relation between a client and his supplier
- Use of other VPN-compatible devices: smart phone
- It is possible to use a secured phone (ex : VoIP SoftPhone application)

4.2.7.3.1 Advantages of VPNs

- Possibility for employees abroad to use a local connection to Internet and their VPN software client to be connected to the corporate network.
- Improvement of the productivity of the users because they connect in a protected way to the company resources independently of the geographical area where they are
- Cheapest costs thanks to the replacement of specialised WAN lines by direct broadband internet connections (distant computers can communicate through an intersite VPN)
- A large company can simplify the topology of its infrastructure by adding VPN to strategic sites

Models for the installation of SSL VPN: Access via a SSL VPN server to certain peripherals of the internal network

- Client connected via Internet (non-sure) to a SSL VPN server located in the internal network

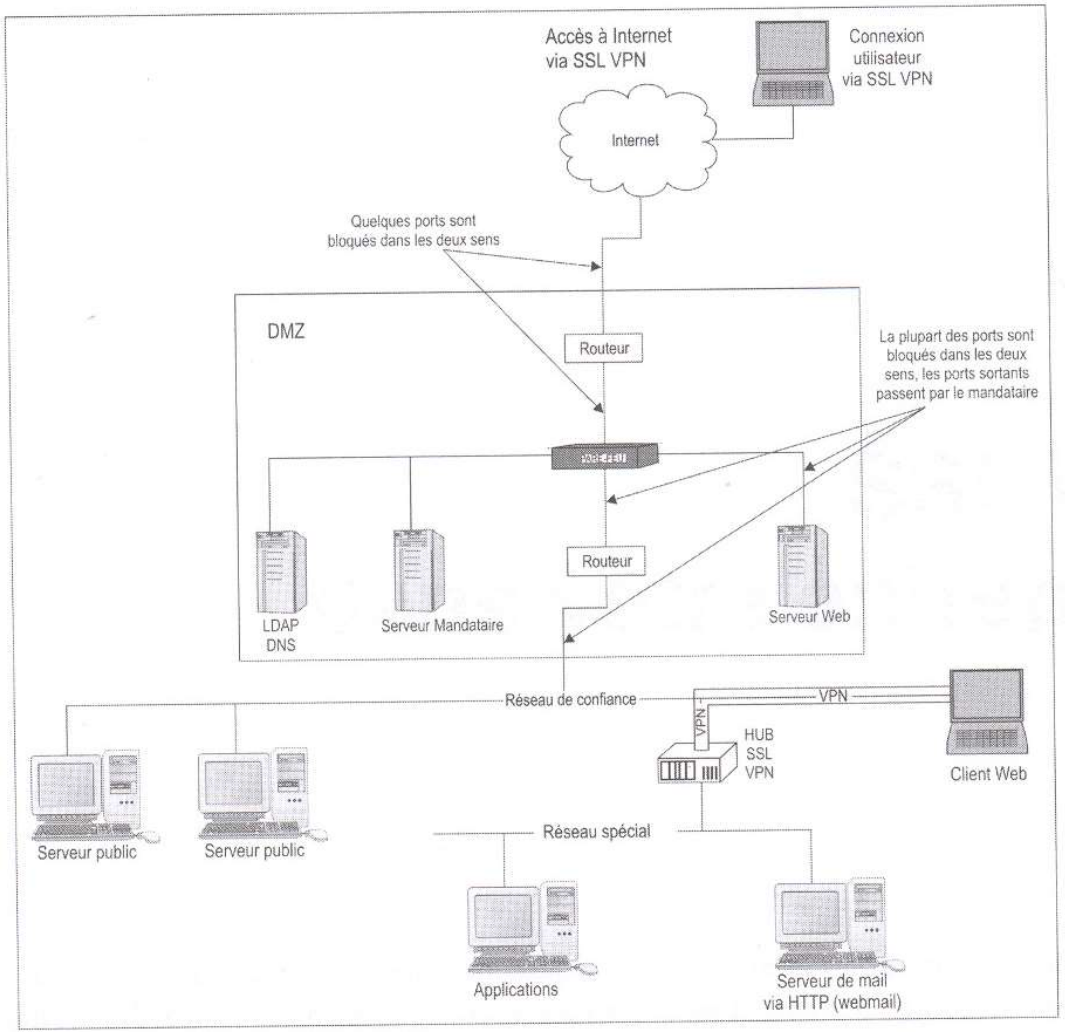


Models for the installation of SSL VPN: Access via a SSL VPN server to certain peripherals of the internal network

- Network frames routed in the DMZ
  - They arrive on the first router (border router)
  - The router will assign an IP of the DMZ to this connection
  - The DMZ checks then this connection
    - Inscription in the logs (history of connection)
    - Detection of the attacks of denial of service
    - …
- If all is correct, connection is transmitted to the router located between the DMZ and the internal network
  - The **destination** address is then modified again to use the internal SSL VPN server IP address (located within the internal network)
- The frame arrives to the internal SSL VPN server
  - SSL server will ensure additional checks, after deciphering the tunnel
    - authentication of the client
    - Negotiation of the encryption protocols
  - The communication could be transmitted to the required server (example: mail server)

Models for the installation of SSL VPN: Access via a SSL VPN server to a special network dedicated to the protected distant accesses

- SSL VPN to reach a private network



Models for the installation of SSL VPN: Access via a SSL VPN server to a special network dedicated to the protected distant accesses

- Example of the large companies
  - Several inter-connected private networks
  - Interconnection by several access suppliers (ISP: Internet Service Provider)
  - Use of POP (Point of Presence), access points to Internet => important security points
  - Protection against company employees
- Use of SSL VPN to provide a sure access to a special internal network from an internal network with limited confidence (hierarchy of internal networks)

SSL VPN at the application level

- Why?
- Many peripherals prohibit the creation of level-3 communication channels, but authorize the level-7 communications through a Web browser
- The most elementary security policy prohibit to connect to a corporate network a computer from a cybercafé or borrowed to anybody
- Disadvantage: very few widespread standards at the application level
- Communication at the layers 6 or 7 level => important impact for security

Advantages SSL VPN vs. proxy

- To reach non web applications
- Accesses to the files, printers and other resources
  - Mounting distant file
  - Web Interface for access to the files
- Access to the printers or other resources
- telnet access, access to terminals
  - Telnet, SSH, Putty…
- terminal servers
- Access to an Intranet
  - Via a non routable private address
  - Via a non published DNS domain (ex: gtr.iut)
- Coherent and ergonomic interface for distant access
- Access, if needed, with the internal network of the company (SSL VPN integrates security functionalities)

Access to the corporate internal network

- Establishment of a network connection through the SSL tunnel
  - Level 3
  - The SSL server sends a program to the client (ActiveX or Java applet) which creates a virtual network interface
  - The client receives an IP address from the internal network
  - Information frames are completely encrypted (from beginning to end)
- Two types of tunnel
  - Complete Tunnel: all the network flow passes through the tunnel => flow towards the internal network and flow towards Internet
  - Partial Tunnel: only the connections towards the internal network pass through the tunnel

Security of a SSL VPN access
1. Identification and authorizations

- Identification
  - Passwords
  - Single use passwords
    - From a passwords list
    - Hardware or software peripheral capable of generating single passwords (as a function of the time, a key…)
    - Challenge-response technique
  - Biometric information
  - Client digital certificates
    - Require a specific and “confident” peripheral
  - Smart cards or USB key
    - Contain a digital certificate which cannot be extracted. Only a digital signature is provided, proving the identity of the user

Security of a SSL VPN access
2. Client security

- Significant data in a non-secure place
  - Data coming from the corporate network on the laptop
  - Cache (browsers + files)
  - Non standard cache (used by some applications (software)…)
  - Temporary files (files attached to mail)
  - Memorization of e-mail addresses (when one fills a form by Internet) or Web addresses (in browser)
  - Cookies
  - Navigation History
  - Swap (exchange file, or auxiliary memory): can be used to store significant data (recovery is difficult, but not impossible)

Security of a SSL VPN access
2. Client security

- Possible corrections
  - Use NOCACHE command in the browser
    - NOCACHE avoids the storage of the received elements
    - Can bring dysfunctions (at the opening of a pdf or Word file for example)
  - To remove all the data at the end of the session, which is difficult:
    - Browser falls down
    - Closing the browser without disconnection
    - Bug of the computer…
  - To use an encrypted storage removed at the end of the session
    - Storage of all session information in a virtual disk
    - Removal of the virtual disk at the end of the session => if the system falls down, the disk can be re-initialised at boot; anyway, the encrypted contents cannot be read
    - Difficulty: this solution does not run with every programs

Security of a SSL VPN access
2. Client security

- Erasing of files
  - Simple “erase” on an operating system does not erase the file physically
  - Military standards: a physical erasing means that one rewrites on the file at least three times random suites of 0 and 1
- Automatic closing of session at the end of a certain time of inactivity
  - Prefer the solutions which consist in warning the user (ex: “are you still present? ” requiring to click on YES preventing connection from stopping) one or two minutes before the end of the session

Security of a SSL VPN access
2. Client security

- Virus entering the internal network via SSL VPN
- Solutions
  - Check the presence of anti-virus on the client computer (activation, last update)
    - Before giving access
  - Prohibit sending of files
  - To base on the corporate internal anti-virus
    - Files sent or attached to e-mail are scanned by the server on their arrival

Security of a SSL VPN access
2. Client security

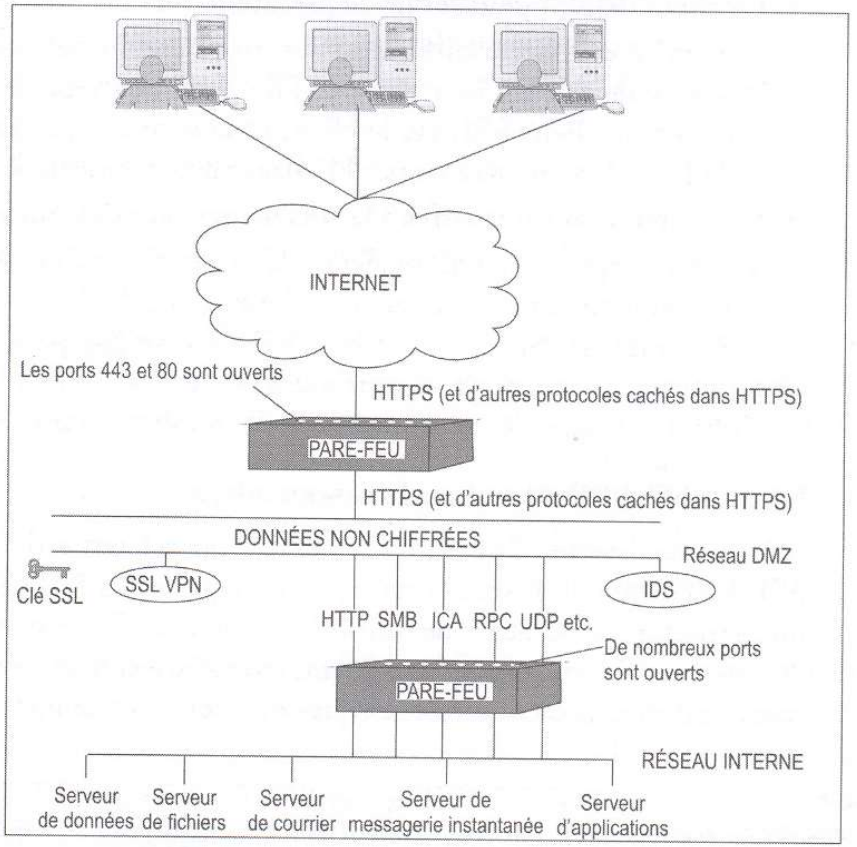
- A worm reaches the internal network via SSL VPN
- Solution: to prohibit connection to a computer which is not protected against the worms
  - Use of personal firewalls (do not let pass the non-desired network traffic, and so the worms)
  - Use of application firewall (with filtering rules)

Access rights according to the security of the distant peripheral

Security of the distant peripheral	Security of the distant peripheral	Security of the distant peripheral	Access rights on SSL VPN	Access rights on SSL VPN	Access rights on SSL VPN	Access rights on SSL VPN
Confidence computer?	Installed and updated antivirus?	Possibility of removing the temporary files?	Authorization for e-mail consultation	Authorization for e-mail sending	Authorization for opening attached files	Authorization for sending attached files
No	Yes	Yes	Yes	Yes	Yes	Yes
No	Yes	No	Yes	Yes	No	Yes
No	No	Yes	Yes	Yes	Yes	No
No	No	No	Yes	Yes	No	No
Yes, confidence level II	Yes	Yes/No	Yes	Yes	Yes	Yes
Yes, confidence level II	No	Yes/No	Yes	Yes	Yes	No
Yes, confidence level I	Yes/No	Yes/No	Yes	Yes	Yes	Yes

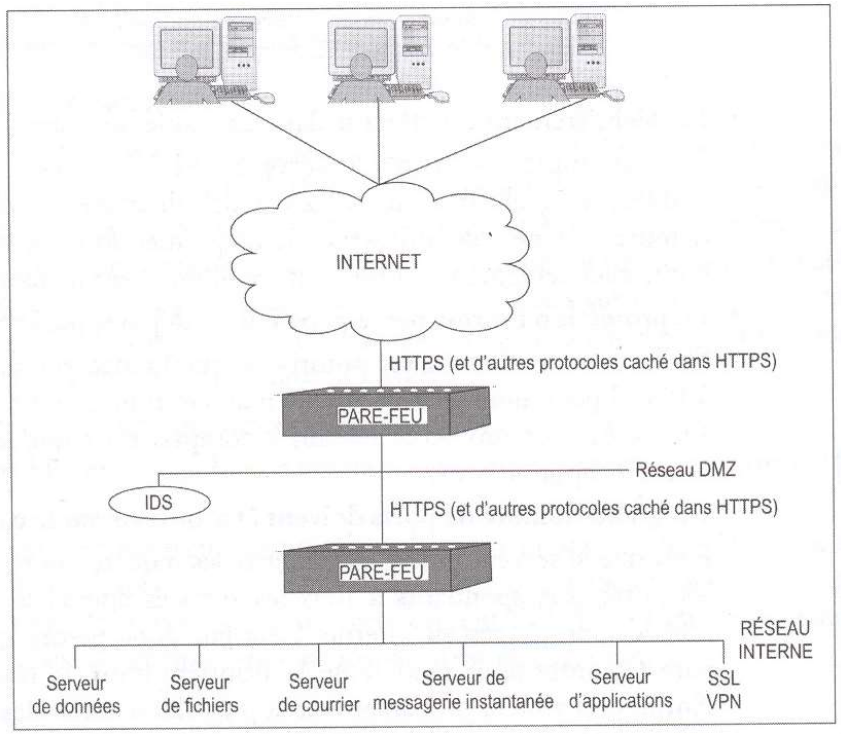
SSL VPN server security

- Firewall
  - Open communications for TCP/IP, and sometimes UDP and ICMP
- SSL VPN server located in the DMZ
  - Firewall must transmit port TCP 443 towards outside (often also port 80)
  - The SSL encrypting keys are stored in a non-secure environment (DMZ)
  - Ciphering is carried out in a non-secure zone (in particular, the communication between SSL VPN server and the internal network are not encrypted)
  - Protection provided by the firewall is thwarted by SSL VPN (it is possible to make pass through the tunnel some protocols which would have been prohibited by the firewall)
  - A large number of ports must be open on the internal firewall…
  - A distant (remote) client can be used as a gateway towards another network



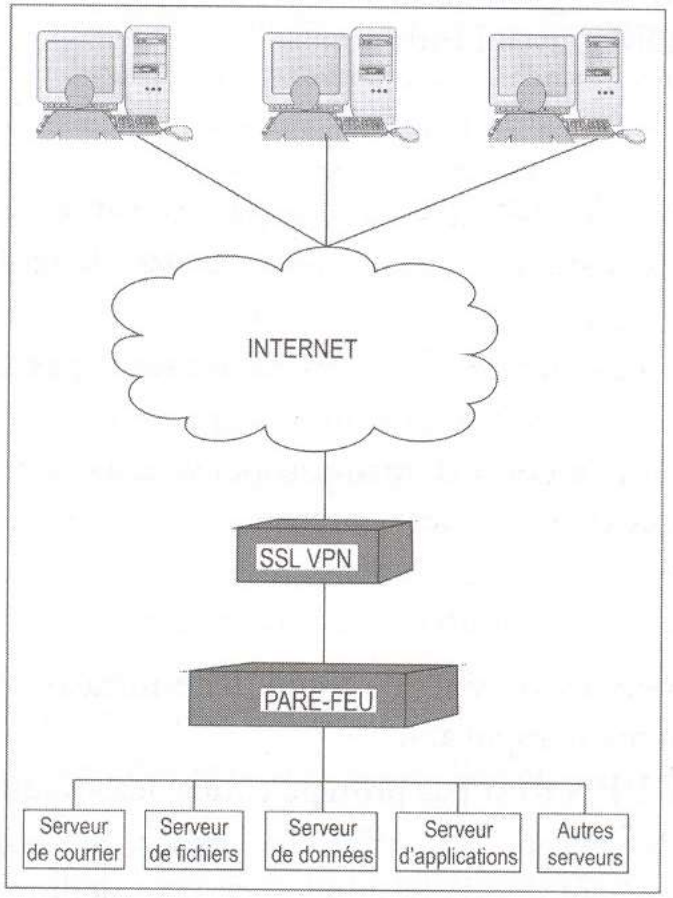
SSL VPN server security

- SSL VPN server located in the internal network
  - the firewall strategy is thwarted
  - Non identified users can send information into the internal network (ex: frames sent by users wishing to be identified)
  - The intrusion detection software (IDS) installed in the DMZ will be ineffective (because information passing through the tunnel is encrypted)



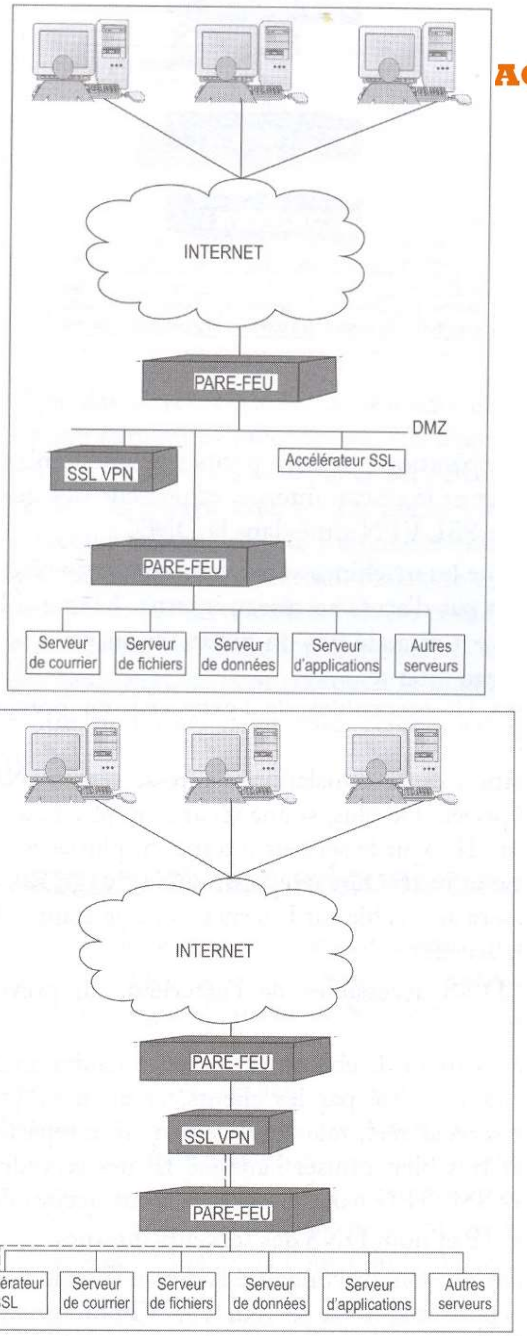
SSL VPN server security

- SSL VPN apart from the external firewall
- Advantages
  - Non - authorized protocols does enter **neither** in the internal network, **nor** in the DMZ
  - Non identified users does enter **neither** in the internal network, **nor** in the DMZ
  - IDS can detect the attacks, as well in the DMZ as in the internal network
- Disadvantages
  - SSL VPN server not protected from the attacks coming from the network
  - deciphering SSL keys are in a hostile environment
  - Need for opening many ports on the external and internal firewalls



SSL VPN server security

- Externalized SSL calculation
  - Discharge the main SSL VPN server by doing ciphering calculation to a dedicated external computer
- Caution: if one wants to install the calculation server in a network surer than the SSL VPN server itself, it is necessary to open ports on the intermediate firewalls, for ex:
  - discharge in the DMZ SSL calculation from a server located on Internet
  - discharge in the internal network SSL calculation from a server located in the DMZ
  - discharge in an internal DMZ SSL calculation from a server located on an external DMZ
- Advantages
  - Ciphering in a sure place
- Disadvantages
  - Opening of the port network

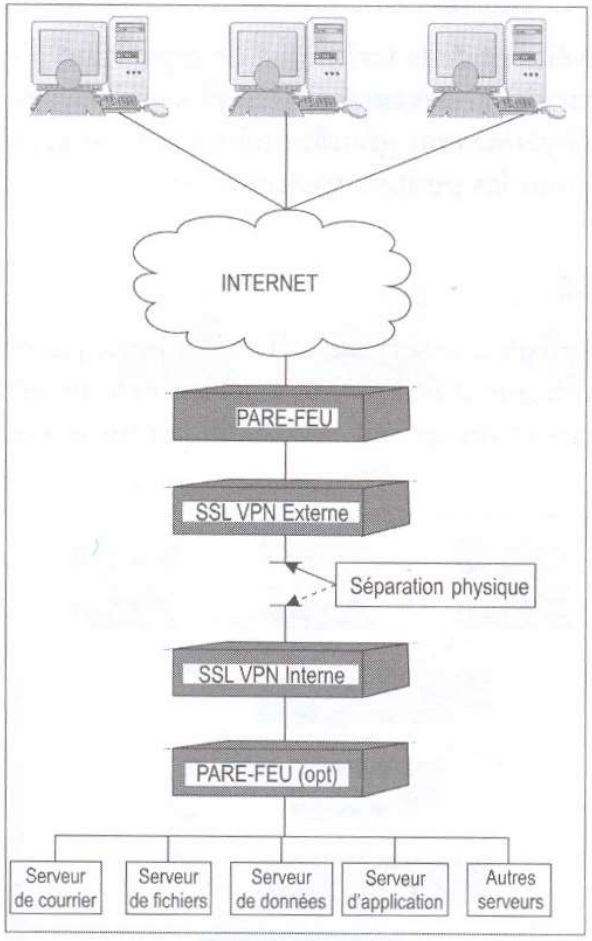


Solutions for the SSL VPN server security

- It does not exist a solution for everything
- Combine various security tools
- SSL VPN must make sure that it interacts well with the firewall and is integrated correctly in its infrastructure
- =>The possibility of network connection through a SSL VPN tunnel should be authorized only for computers which are authorized in the local area network (computers managed by the company)
- Be aware of the fact that even in this case, the firewall of the computer becomes indeed one of the firewall of the company => it is not necessarily dimensioned for that purpose…
- Prefer in general other distant (remote) accesses that the compkete access to the network (i.e. to avoid giving the possibility to a computer of going anywhere in the network while connecting itself by VPN), for example:
  - Route through the tunnel only necessary ports
  - Allow only a restricted access (whom? which application? which server? From which peripheral (a computer known with updated firewall and anti-virus…))

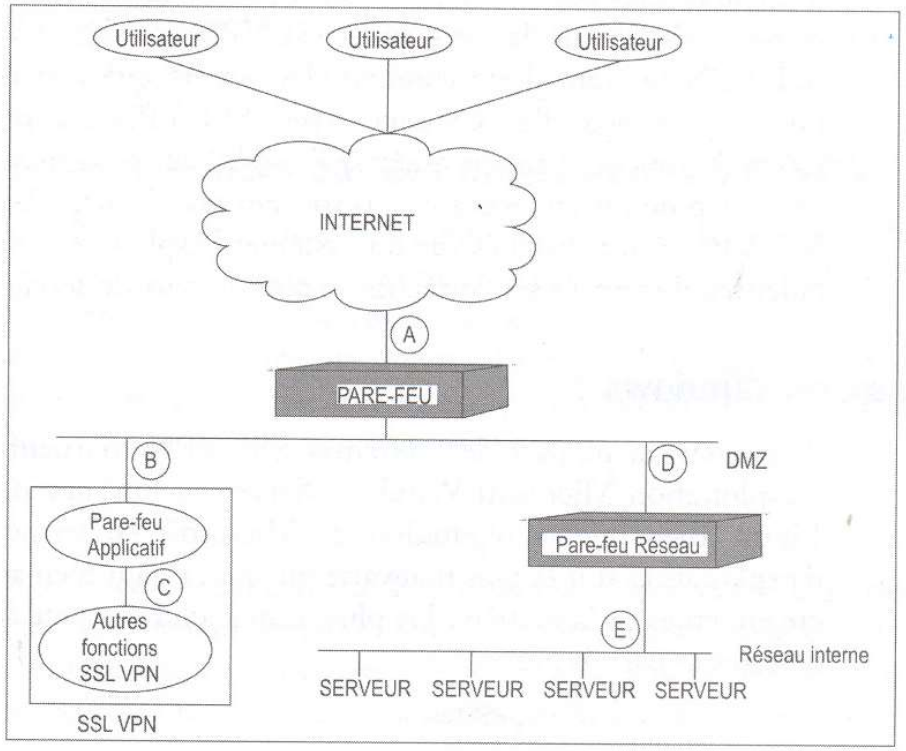
Solutions for the SSL VPN server security

- Problem of storage of SSL certificate
  - use an SSL accelerator => makes it possible to accelerate the processing times and to store the certificates in the protected environment of SSL accelerator
  - use a “physical separation”, called Air Gap Technology, which makes it possible to strore the certificate in the protected environment of the internal network. This technique consists in using two servers sharing a common memory (banks of memory allowing them to communicate)
    - a server is connected to Internet
    - a server is connected to the internal network and is running the whole SSL VPN functions
  - use both systems to protect even more effectively the certificates at the same time



Problem of the application faults

- Faults can be detected in the server application
- The operating system used by SSL VPN server can be vulnerable to security specific problems
- Protection to be considered
  - set up an application filtering to protect from the worms (on the right figure the data circulate according to ABCDE)
  - Filtering of the requests sent by a distant client to an internal server via SSL VPN



Other considerations on SSL VPN servers

- use an adequate ciphering strategy
- update the SSL VPN server software (bugs)
- Linux or Windows
  - No system is perfect
  - To be kept informed of security faults and failures and update the systems accordingly
- Consider the physical separation strategy
  - But expensive (two servers)
- Also do not to forget to make safe the SSL VPN server in the internal network
  - protect from any attack coming from the internal network
  - require a password for the access to the server administration functionalities
  - Configuration of the firewall between the internal network and SSL VPN server (to avoid the worms and spywares propagation)

Use of a SSL VPN server: determine the needs for the company

- Inter-sites communication (to give the illusion of a complete network)
  - Prefer IPSec
    - use technologies of virtual private networks between the sites, and inter-connect them
    - Equipment or dedicated software installed in the firewall in network edge
- Communication of a user towards an exploitation site
  - Distant access of the users to the resources of the internal network such as files, applications, databases, terminals services => SSL VPN is a suitable technology for this type of distant access

Use of a SSL VPN server: determine the user's needs

- E-mail distant access
  - Solutions of SSL VPN service dedicated to the e-mail
- Complete access to the network
  - prefer a solution containing IPSec + authentication of the client by digital certificate, with recognition of a specific computer => to be limited to some rare users
- Accesses for the customers and suppliers of the company
  - SSL VPN server which can deal with configurations with complexes rights of access management (accessible (reachable) applications are different according to the types of profiles and connected accounts)
- Distant access to a workstation for one or two users
  - prefer the use of a simple protected software for distant takeover such as PcAnywhere or a software from the VNC suite (UltraVNC, TightVNC, etc…) not very expensive and responding exactly to the the expected functionality

Models of SSL VPN servers

- http://www.aepnetworks.com
- www.arraynetworks.net
- fr.aventail.com
- www.checkpoint.com
- www.cisco.com
- citric.fr
- www.f5.com
- www.ipdiva.com
- juniper.net
- www.netsilica.com
- www.nokia.com
- nortel.com
- permeo.com
- portwise.com
- safenet-inc.com
- www.sonicwall.com/products/sslapp.html
- www.symantec.com/Products/enterprise?c=prodcat&refId=1006
- www.whalecommunications.com

4.2.7.3.2 RADIUS server

4.2.7.3.2 RADIUS server

- Remote Authentication Dial-In User Service: service of authentication for the users for on-the –request connections
- Allows to sub-contract the requests for session openings and the connections follow-up
- Service largely used by ISP (Internet Service Providers)
- RADIUS is not an official standard, but is maintained by a working group of the IETF

4.2.7.3.2 Introduction

- Protocol allowing to centralize the authentication and the authorization of distant users
  - Services and applications capable of taking into account RADIUS authentication are: routers, firewalls, wireless access points, etc…
  - Developed by Linvingstone Entreprise Inc
  - IETF (RFC 2865-2866 and 2869)
  - Client/server protocol functioning with UDP

4.2.7.3.2 Use of a RADIUS server (ex 1: Windows server)

- Use of Active Directory in order to authenticate the Internet accesses for the private networks users
  - The firewall which allows connection to Internet has rules forcing an authentication of the users in order to open or not the access to them
  - The firewall technology used deals with the RADIUS protocol

⇒ We can configure the firewall as a RADIUS client of a RADIUS server functioning under W. Server and member of the Active Directory domain to which the users belong

⇒ This offers an Internet accesses authentication for the users, without having to define a new accounts base for the firewall (and so having a complex management the use of the passwords)

4.2.7.3.2 Use of a RADIUS server (ex 2)

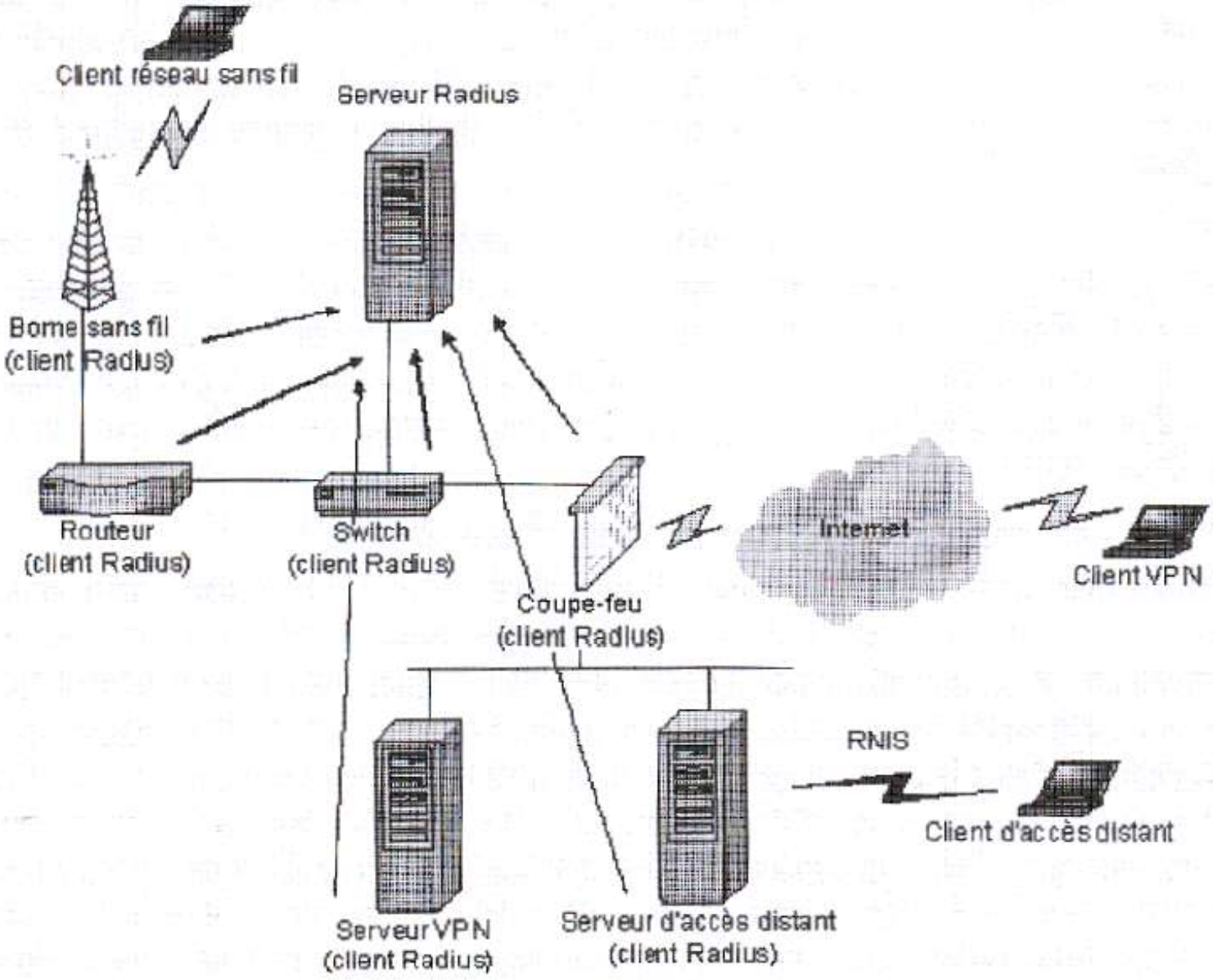
- The company owns several distant access and VPN servers
  - Creation of one or a set of strategies on the RADIUS server
  - Configuration of the distant accesses and VPN servers as RADIUS clients

Use of a RADIUS server (ex 3)

- One wishes to reinforce the access security at the borders of the wireless network. The RADIUS protocol can be used

4.2.7.3.2 Other interests to use a RADIUS server

- Centralization of the authentication
- Authentication of VPN clients in a domain which VPN server does not belong to
- Ex:
  - Installation in the DMZ of a VPN server in a working group
  - Configure in such a way that the authentication of VPN clients is done in their domains, by re-directing authentication requests towards a RADIUS server which is a member of this domain


Client réseau sans fil
Serveur Radius
Borne sans fil
(client Radius)
Routeur
(client Radius)
Switch
(client Radius)
Coupe-feu
(client Radius)
Internet
Client VPN
RNIS
Client d'accès distant
Serveur VPN
(client Radius)
Serveur d'accès distant
(client Radius)

4.2.7.3.2 Configuration of the RADIUS server

- Creation of distant access strategies
- clients authentication
- Definition of the Radius clients for whom the Radius server will operate
- Use of the MMC “authentication Internet Service” / *"Service d'authentification internet"*

4.2.7.3.2 Configuration of the RADIUS client

- Use of the “Routing and distant access” console / *"Routage et accès distant"*
- “Properties of the server” / *"Propriétés du serveur"*
- “Supplier for authentication” / *"Fournisseur d'authentification"*
- “RADIUS authentication ” / *"Authentification RADIUS"*
- “configure” / *"Configurer"*

References (1/2)

- VPN, mise en œuvre sous Windows Server 2003, P. Mathon, 2004.
- La sécurité des réseaux-First steps, Tom Thomas, Cisco Press, 2005.
- SSH, le shell sécurisé, D. J. Barrett et R.E. Silverman, O'Reilly, 2001.
- SSL VPN, Understanding, evaluating and planning secure, web-based remote access – J. Steinberg & T. Speed, 2005.
- Les réseaux, édition 2005, G. Pujolle, Eyrolles 2004
- Présentation de Eric WIESS, 2005
- S. Ghernaouti-Helie – *Sécurité informatique et réseaux, 4ème édition* – Dunod, 2013
- E. Cole, R. Krutz, JW Conley - *Network security bible* – Wiley, 2005
- L. Bloch & al. – Sécurité informatique pour les DSI, RSSI et administrateurs, Eyrolles, 2016.
- Présentation de Eric WIESS
- Cours de Jean-Luc Noizette, ESSTIN, Nancy, 2005
- G. Avoine, P. Junod, P. Oechslin – Sécurité informatique, exercices corrigés – Vuibert, Paris, 2006.
- NIST, Guidelines on firewalls and firewall policy, J. Wack & al., 2002.
- The 60 minute network security guide (first steps towards a secure network environment), 2006, SNAC

References (2/2)

- Cours réseaux et télécoms avec exercices corrigés, G. Pujolle, Eyrolles 2006
- SSL VPN, Understanding, evaluating and planning secure, web-based remote access – J. Steinberg & T. Speed, 2005.
- P. H. Oechlin, LASEC/EPFL
- http://sebsauvage.net/comprendre/encryptage/crypto_rsa.html
- S. Ghernaouti-Helie – *Sécurité informatique et réseaux, 4ème édition* – Dunod, 2013
- F. Halsall – Computer networking and the internet – Addison Welseley, 2005 + additional student support at www.pearsoned.co.uk/halsall
- D. Vergnaud – Exercices et problèmes de cryptographie, Dunod, 2015
- CEH, Certified Ethical Hacker, Matt Walker, McGrawHill, 2017
- L. Bloch & al. – Sécurité informatique pour les DSI, RSSI et administrateurs, Eyrolles, 2016.