

Sistemi operativi

Brevi note

Ruolo del S.O.

Quell'insieme di programmi, servizi, utilities che vi consentono di utilizzare al meglio le risorse e i dispositivi di un computer

Quali OS conosciamo

Windows

Windows 11, 10, 8, 7, vista, XP, 2000, NT, home, 98, 95, 3.11,

MacOs

Sierra,

Linux

Ubuntu, debian, Mint, gentoo, parrotos, kali, redhat, centos, ..., LFS (Linux From Scratch)

Elementi essenziali di un OS

**File system MANAGER
Kernel
Applicazioni di gestione**

Dei file
Della rete
CLI

...

Windows

Essenzialmente visto come un monolite

Un kernel

Quello legato alla versione del OS

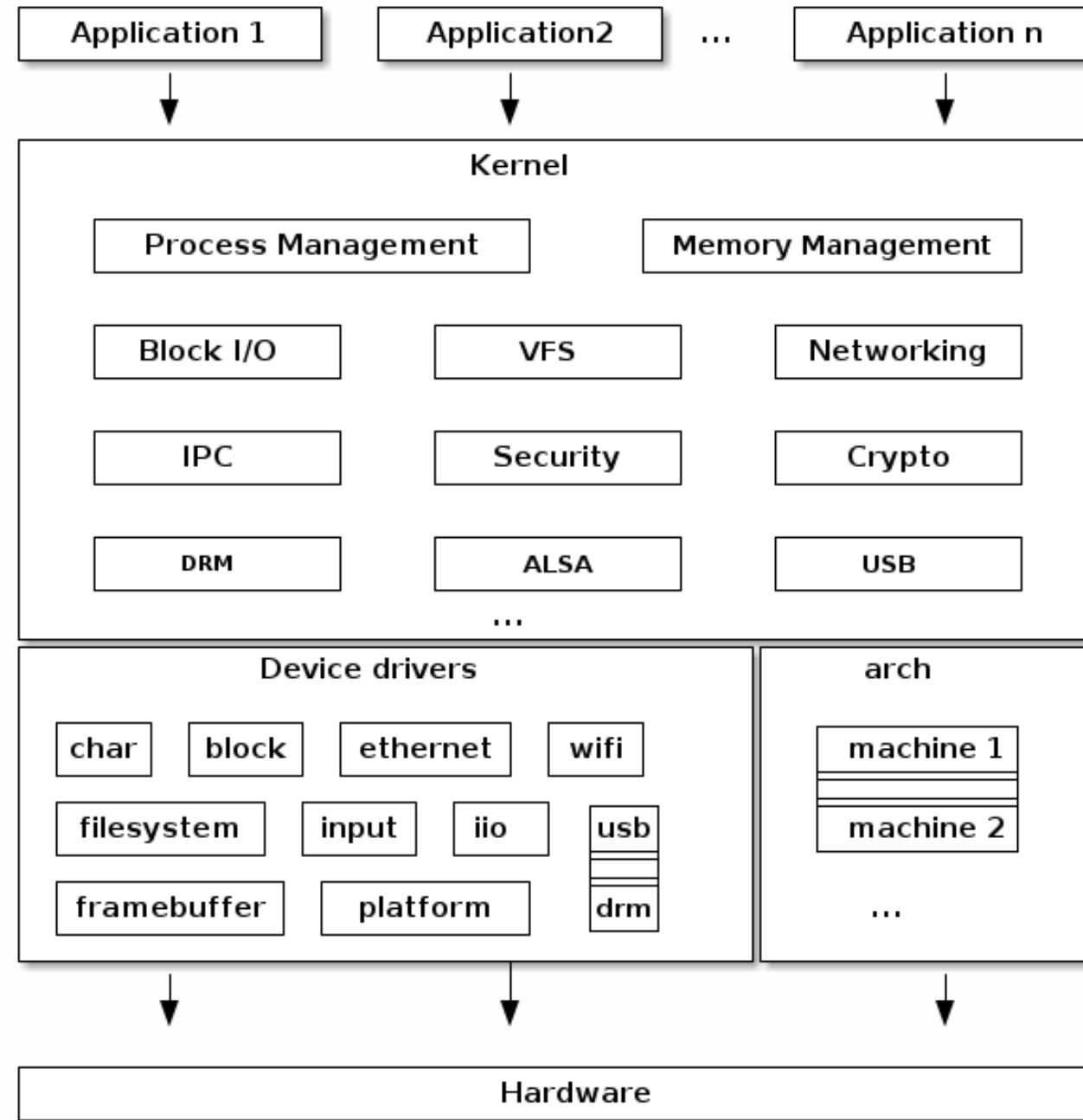
Un FSM

Essenzialmente Gestione Risorse

Strutturazione dei file/cartelle e organizzazione fisica e logica del disco: NTFS (prima era FAT, FAT32, FAT16, ...). Per i CD è lo standard ISO9660

Network manager

Un set di programmi/applicazioni/utilities



I segnali audio

Vinile

Segnali analogici: il solco del vinile fa vibrare la testina che trasforma la vibrazione in segnale elettrico che altera il campo elettromagnetico dell'altoparlante che fa a sua volta vibrare meccanicamente l'aria che porta la vibrazione meccanica ai nostri timpani che la trasformano in segnale elettrico processato dal nostro cervello

I segnali audio nel PC

**Il PC usa dati “analogici” o dati “digitali”?
Digitali e li trasforma in analogici tramite
convertitore DA (Digitale => analogico)**

MSDH-W012A
227646

6G17 0194-2349

AWM 20696 80C VW-1

Aritmetica del computer

All'interno del computer la CPU manipola esclusivamente valori binari

Bit

Byte: 8 bit

Word: 16 bit (short)

DoubleWord: 32 bit (int)

QuadWord: 64 bit (long in java e in C#, long long in C/C++/...)

CPU

Sa spostare valori (byte, word, Dword, Qword) tra zone di memoria

Sa inizializzare (byte, word, Dword, Qword) una zona di memoria

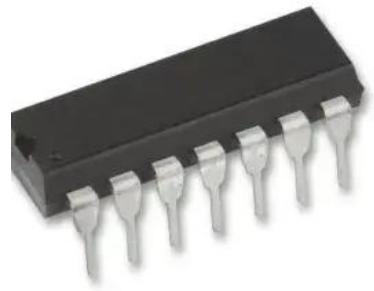
Sa effettuare operazioni logiche (byte, word, Dword, Qword) tra zone di memoria

And, Or, Xor, Not

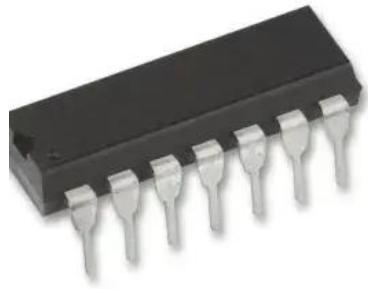
Inoltre, tramite composizione di elementi logici, sa effettuare operazioni aritmetiche elementari (+, -, *, /, inc, dec)

Sa leggere la prossima istruzione da eseguire in modo sequenziale oppure sa eseguire istruzioni di “salto” per andare a eseguire un frammento di codice in altre zone della memoria

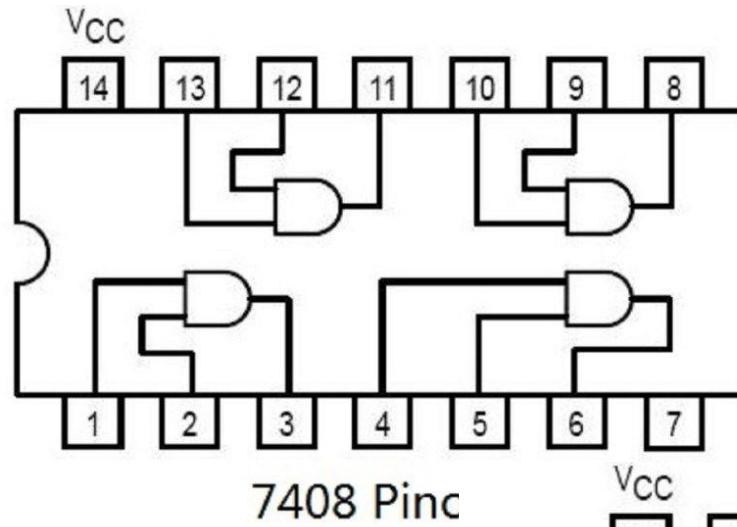
Per fare and tra due numeri a 8 bit (due byte)



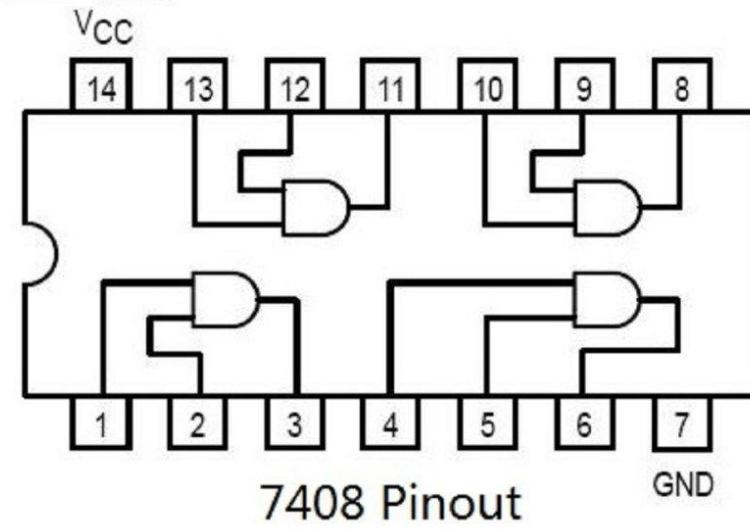
7408 AND GATE



7408 AND GATE

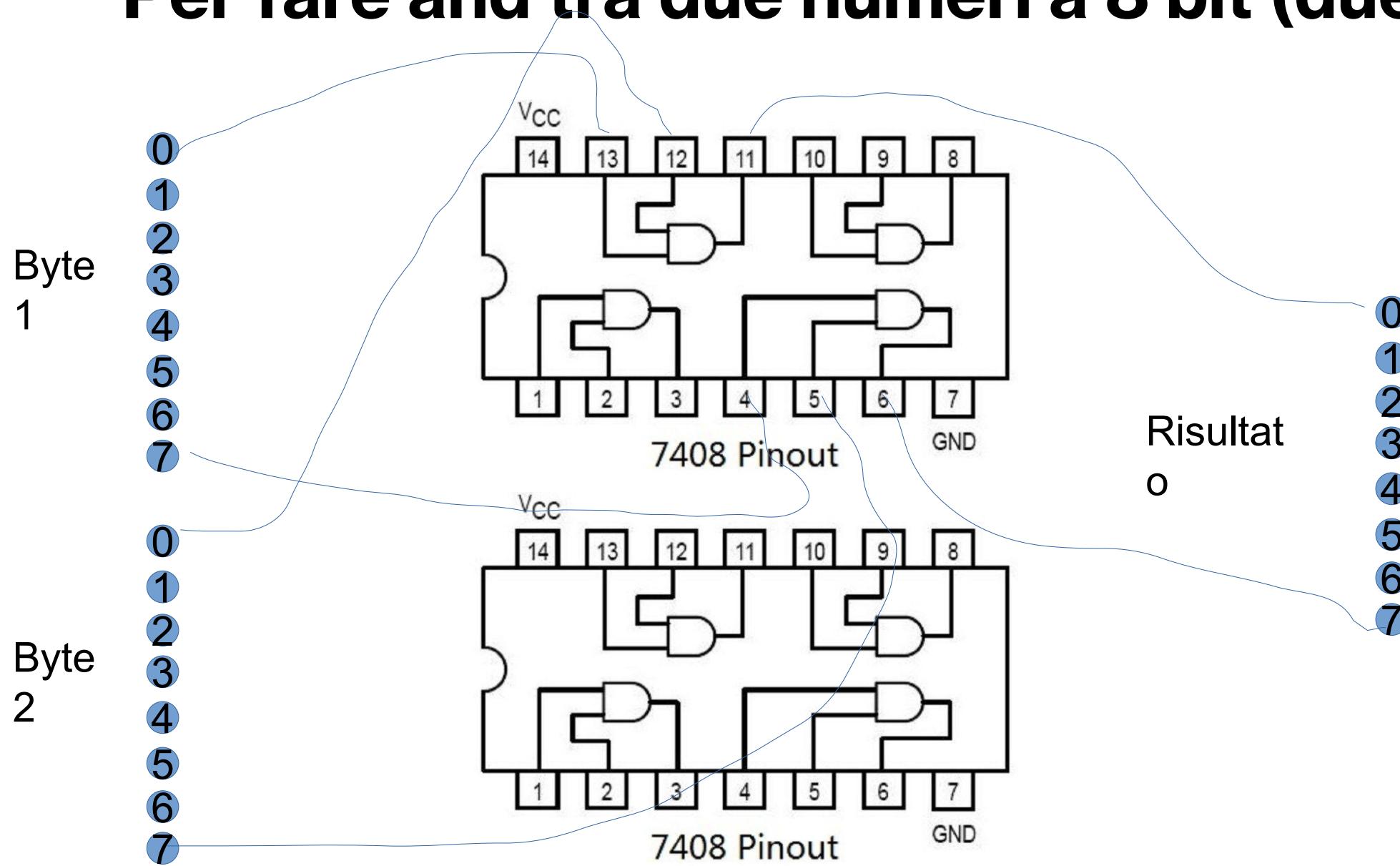


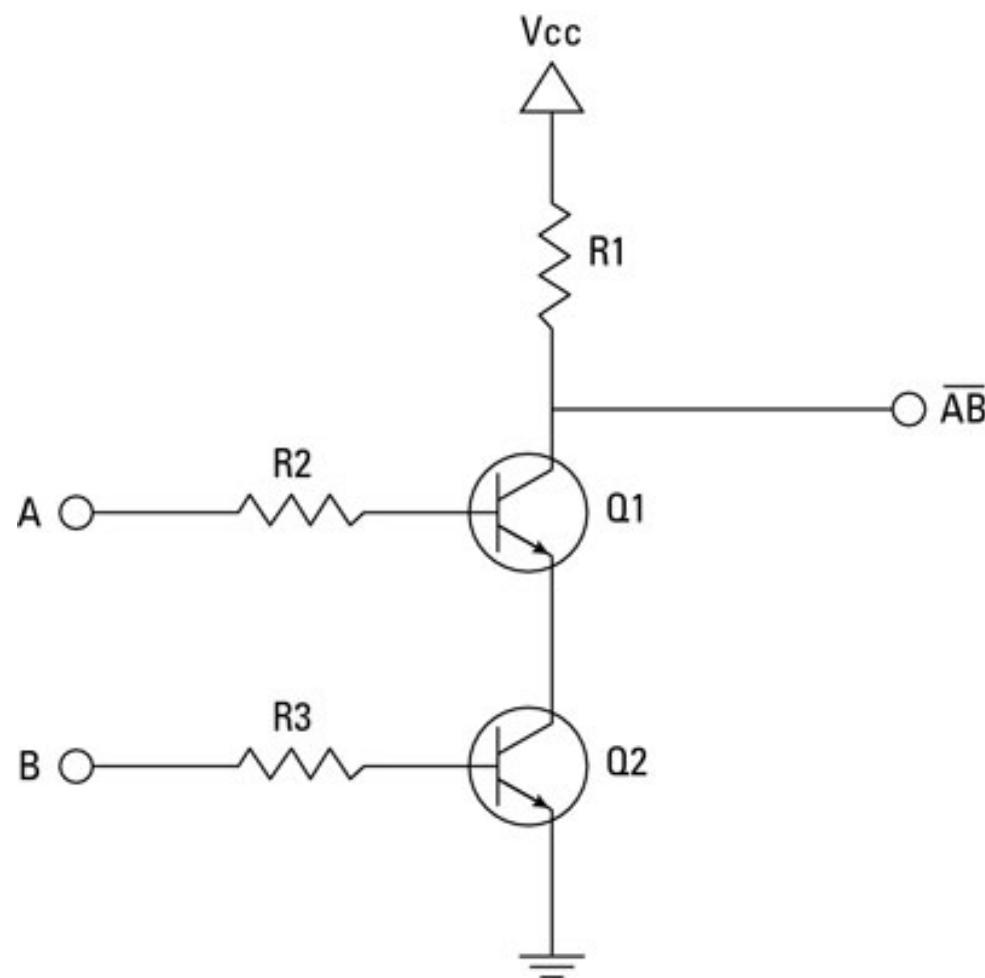
7408 Pinout



7408 Pinout

Per fare and tra due numeri a 8 bit (due byte)





Aritmetica

	SOMMA	RIPORTO
0	0	0
0	1	0
1	0	0
1	1	1

Operazioni con numeri binari

4789
1000, 100, 10, 1
4 7 8 9

4096 256 16 1
4 7 8 9

Operazioni con numeri binari

10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0
256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	0	1	0	1	0

Operazioni con numeri binari

Byte	01101011
2 nibble	0110 1011

Dato che con 4 bit posso rappresentare $2^4=16$ valori diversi, mi invento una notazione
ESADECIMALE

Basi numeriche

**Ho tre urne che contengono rispettivamente
palline rosse, gialle e verdi**

**In quanti modi posso comporre una sequenza di
tre palline?**

**In quanti modi posso comporre una sequenza di
quattro palline?**

Basi numeriche

- RGR
- RVB
- RRR
- GRG
- RGB

RRR
GGG
VVV
RRG
RRV

- NE VVR
MANCANO
ANCORA 7
POICHE
SONO RV
 $3^3=27$
• GRG

Basi numeriche

**Ho 10 urne che contengono rispettivamente
palline 0,1,2,3,4,5,6,7,8,9**

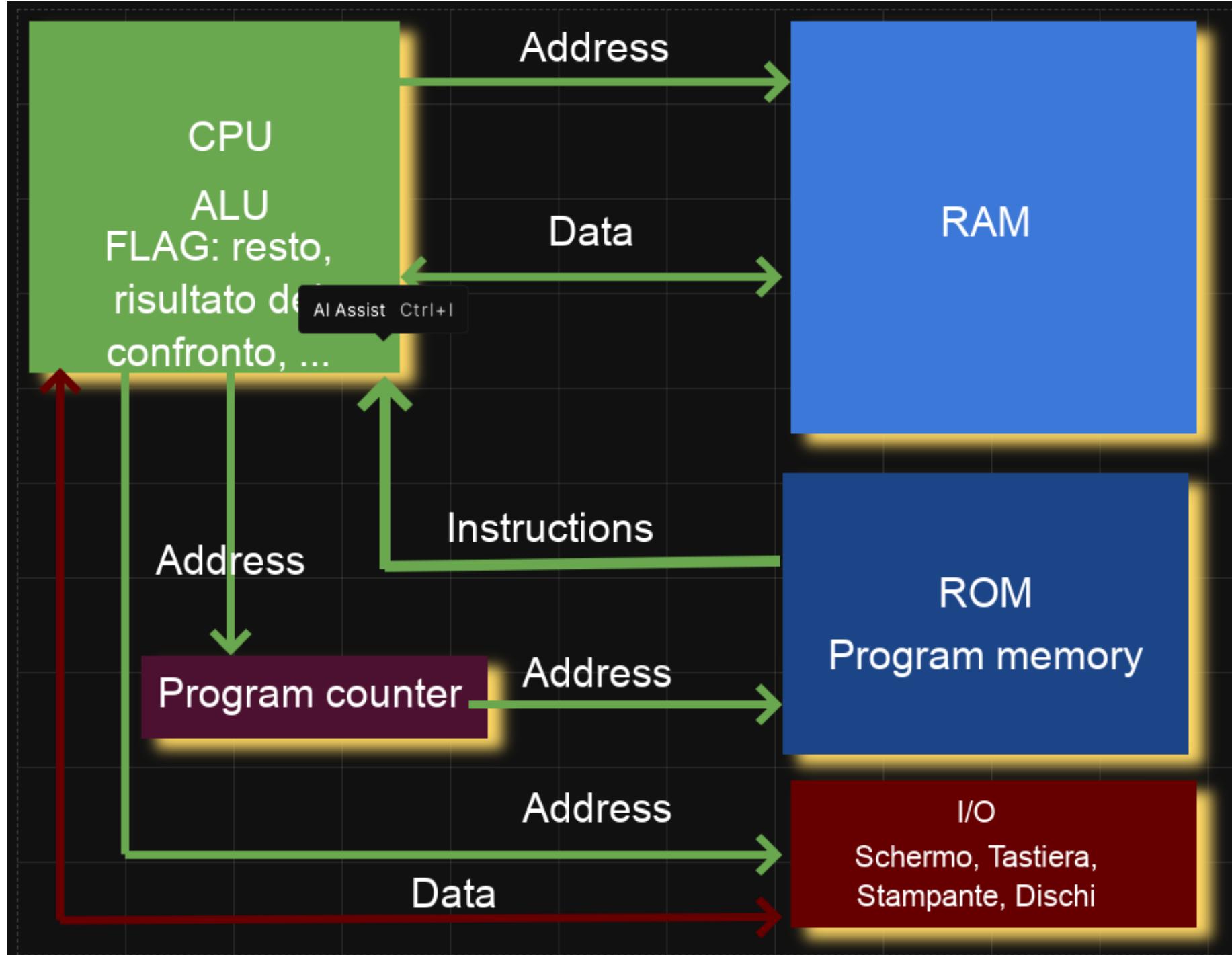
**In quanti modi posso comporre una sequenza di
tre palline?**

**000,001,002,003,004,005,006,007,008,009,01
0,011,012,013,014,015,016...**

- VRR
- VRV
- VRG
- VVR
- VVV
- VVG

RRR
RRV
RRG
RGR
RGV
RVG
RGG

- **GGR**
- **GGV**
- **GEG**
- GVR
- GVV
- GVG



DLL e SO

```
/data/..../Esercitazioni $ ldd /bin/top
linux-vdso.so.1 (0x00007ffd2c1f0000)
/usr/lib/x86_64-linux-gnu/libgtk3-nocsd.so.0 (0x00007d38bc9af000)
libproc2.so.0 => /lib/x86_64-linux-gnu/libproc2.so.0 (0x00007d38bc941000)
libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007d38bc90d000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007d38bc600000)
libsystemd.so.0 => /lib/x86_64-linux-gnu/libsystemd.so.0 (0x00007d38bc82e000)
/lib64/ld-linux-x86-64.so.2 (0x00007d38bca09000)
libcap.so.2 => /lib/x86_64-linux-gnu/libcap.so.2 (0x00007d38bc81f000)
libgcrypt.so.20 => /lib/x86_64-linux-gnu/libgcrypt.so.20 (0x00007d38bc4b8000)
liblz4.so.1 => /lib/x86_64-linux-gnu/liblz4.so.1 (0x00007d38bc495000)
liblzma.so.5 => /lib/x86_64-linux-gnu/liblzma.so.5 (0x00007d38bc463000)
libzstd.so.1 => /lib/x86_64-linux-gnu/libzstd.so.1 (0x00007d38bc3ac000)
libgpg-error.so.0 => /lib/x86_64-linux-gnu/libgpg-error.so.0 (0x00007d38bc387000)
```

Tutti questi file contengono codice eseguibile (codice macchina) che viene utilizzato dal comando “top”

LD_LIBRARY_PATH="....." top => il comando top va a cercare le librerie dinamiche prima nei path specificati da LD_LIBRARY_PATH e poi nei path di sistema

Dump delle comunicazioni uscenti da un cellulare

**Hotspot
Dump sul PC che fa hotspot**



AD conversion

Quanto spazio occupa nel disco un brano da 3 minuti?

Il campionamento è 44100 campioni / secondo

$$3*60*44100 = \text{numero di campioni acquisiti} = \\ 7938000$$

In generale noi registriamo stereo e quindi sono
 $7938000*2=16$ Mbyte * 2 poiché registro a 16 bit = 32
Mbyte/brano

Un CDROM musicale contiene max 700 MB
 $700MB/32MB = \text{circa 20 brani per CD}$

AD conversion

Quanto spazio occupa su disco un film di 2hh?

Full HD

24 pagine/sec * 1920*1080 *3 * 7200 =

1.074.954.240.000

Quanti processi ci sono in esecuzione sul vostro Laptop?

< 10?

10 < n < 100?

> 100?

ps aux
ps -ef
top

Quanti collegamenti di rete ho attivi sul mio PC in questo momento

sudo netstat -anp –tcp

Per avere tutte le comunicazioni

sudo netstat -anp

sudo apt install net-tools

Per windows

netstat -ano -p tcp

Linux

tracepath (o traceroute) -n www.microsoft.com

**Quanti file (intesi come stream) ci sono “aperti”,
cioè in uso, in questo momento?**

sudo lsof

Quante sono e quali caratteristiche hanno le CPU nel mio laptop?

cat /proc/cpuinfo

Quante porte USB ho sul mio laptop

lsusb -v

Hardware della mia macchina

sudo lshw

sudo lspci

Windows: systeminfo

E quante schede di rete (virtuali e reali) avete?

ip a

Quindi, totale dei comandi da eseguire

sudo ps aux > elencoprocessi.txt

sudo netstat -anp > elencoconnessioni.txt

sudo lsof > fileaperti.txt

cat /proc/cpuinfo >cpuinfo.txt

sudo lshw > lshw.txt

ip a > elencoschede.txt

sotto windows in command.com: systeminfo > systeminfo.txt

Network protocols

Descrizioni dei tipi Ethernet
0x0000 0x05DC IEEE 802.3
Length Fields
0x0600 0x0600 Xerox XNS IDP
0x0800 0x0800 DOD IP
0x0801 0x0801 X.75 Internet
0x0802 0x0802 NBS Internet
0x0803 0x0803 ECMA Internet
0x0804 0x0804 CHAOSnet
0x0805 0x0805 X.25 Level 3
0x0806 0x0806 ARP (for IP and
CHAOS)
0x0807 0x0807 Xerox XNS
Compatibility

0x081C 0x081C Symbolics Private
0x0888 0x088A Xplex
0x0900 0x0900 Ungermann-Bass
network debugger
0x0A00 0x0A00 Xerox 802.3 PUP
0x0A01 0x0A01 Xerox 802.3 PUP
Address Translation
0x0A02 0x0A02 Xerox PUP CAL
Protocol (unused)
0x0BAD 0x0BAD Banyan Systems,
Inc.
0x1000 0x1000 Berkeley Trailer
negotiation

0x1001 0x100F Berkeley Trailer encapsulation for IP
0x1066 0x1066 VLMIS Systems
0x1600 0x1600 VALID Systems
0x3C01 0x3C0D 3Com Corporation
0x3C10 0x3C14 3Com Corporation
0x4242 0x4242 PCS Basic Block Protocol
0x5208 0x5208 BBN Simnet Private
0x6000 0x6000 DEC Unsigned
0x6001 0x6001 DEC MOP Dump/Load Assistance
0x6002 0x6002 DEC MOP Remote Console
0x6003 0x6003 DEC DECnet Phase IV
0x6004 0x6004 DEC LAT
0x6005 0x6005 DEC DECnet Diagnostic Protocol:
DECnet Customer Use
0x6007 0x6007 DEC DECnet LAVC

0x8010 0x8010 Excelan
0x8013 0x8013 SGI graphics type (obsolete)
0x8014 0x8014 SGI network games (obsolete)
0x8015 0x8015 SGI reserved type (obsolete)
0x8016 0x8016 SGI bounce server (obsolete)
0x8019 0x8019 Apollo
0x801E 0x8025 Tymshare
0x802F 0x802F Tigan, Inc.
0x8031 0x8035 Reference ROM (obsolete)
0x8036 0x8036 Aedonic Systems
0x8038 0x8038 DEC LANBridge
0x8039 0x8039 DEC NSMV
0x803A 0x803A DEC Aragon
0x803B 0x803B DEC AXP
0x803C 0x803C DEC NSMV
0x803D 0x803D DEC Ethernet CSMA/CD
EnergyStar Protocol
0x803E 0x803E DEC DNA
0x803F 0x803F DEC Video Monitor
0x8040 0x8040 DEC NetBIOS
0x8041 0x8041 DEC MOPDS
0x8042 0x8042 DEC Unsigned
0x8044 0x8044 Planning Research Corporation
0x8045 0x8045 STAP
0x8047 0x8047 AT&T
0x8049 0x8049 ExpertData
0x805B 0x805B VMTP (Versatile Message
Transport Protocol, AF-1045,
Stanford)
0x805C 0x805C Stanford V Kernel production
0x805D 0x805D Evans & Sutherland
0x8060 0x8060 Little Machines

0x8062 0x8062 Counterpoint Computers

0x80C0 0x80C0 Digital Communication Associates

0x80C1 0x80C1 Digital Communication Associates

0x80C2 0x80C2 Digital Communication Associates

0x80C3 0x80C3 Digital Communication Associates

0x80C4 0x80C4 Digital Communication Associates

0x80C5 0x80C5 Digital Communication Associates

0x80C6 0x80C6 Digital Communication Associates

0x80C7 0x80C7 Applitek Corporation

0x80C8 0x80C8 Intergraph Corporation

0x80C9 0x80C9 Harris Corporation

0x80CE 0x80CE Harris Corporation

0x80CF 0x80D2 Taylor Inst.

0x80D0 0x80D0 Rosemount Corporation

0x80D4 0x80D4 Rosemount Corporation

0x80D5 0x80D5 IBM SNA (Multiprotocol Ethernet)

0x80DD 0x80DD Varian Associates

0x80DE 0x80DE Integrated Solutions TRFS

(Transparent Remote File System)

0x80DF 0x80DF Integrated Solutions

0x80E0 0x80E3 Allen-Bradley

0x80E1 0x8010 Database

0x80F2 0x80F2 Retix

0x80F3 0x80F3 Kinetics, AppleTalk ARP (AARP)

0x80F4 0x80F4 Kinetics

0x80F5 0x80F5 Kinetics

0x80F7 0x80F7 Apollo Computer

0x80FF 0x80F3 Wellfleet Communications

0x8107 0x8107 Symbolics Private

0x8108 0x8108 Symbolics Private

0x8109 0x8109 Symbolics Private

0x8130 0x8130 Waterloo Microsystems

0x8065 0x8065 University of Massachusetts, Amherst

0x8066 0x8066 University of Massachusetts, Amherst

0x8067 0x8067 Veeco Integrated Automation

0x8068 0x8068 General Dynamics

0x8069 0x8069 AT&T

0x806A 0x806A Autophon (Switzerland)

0x806C 0x806C ComDesign

0x806D 0x806D Compugraphic Corporation

0x806E 0x8077 Landmark Graphics Corporation

0x807A 0x807A Matra (France)

0x807B 0x807B Dansk Data Elektronic A/S (Denmark)

0x807C 0x807C Merit Intermodal

0x807D 0x807D VitaLink Communications

0x807E 0x807E VitaLink Communications

0x8131 0x8131 VG Laboratory Systems

0x8137 0x8137 Novell (old) NetWare IPX

0x8138 0x8138 Novell

0x8139 0x813D KTI

0x9000 0x9000 Loopback (Configuration Test Protocol)

0x9001 0x9001 Bridge Communications XNS Systems Management

0x9002 0x9002 Bridge Communications TCP/IP Systems Management

0x9003 0x9003 Bridge Communications

0xFF00 0xFF00 BBN VITAL LANBridge cache wakeup

Ethernet packet structure

Struttura del pacchetto ETHERNET

6 bytes Destination Ethernet Address (Tutti 1 se broadcast, ...)

6 bytes Source Ethernet Address

2 bytes Length or Type Field

per IEEE 802.3 è il numero di bytes di dati

per ethernet I o II è il "packet type", sempre 1500(05DC)

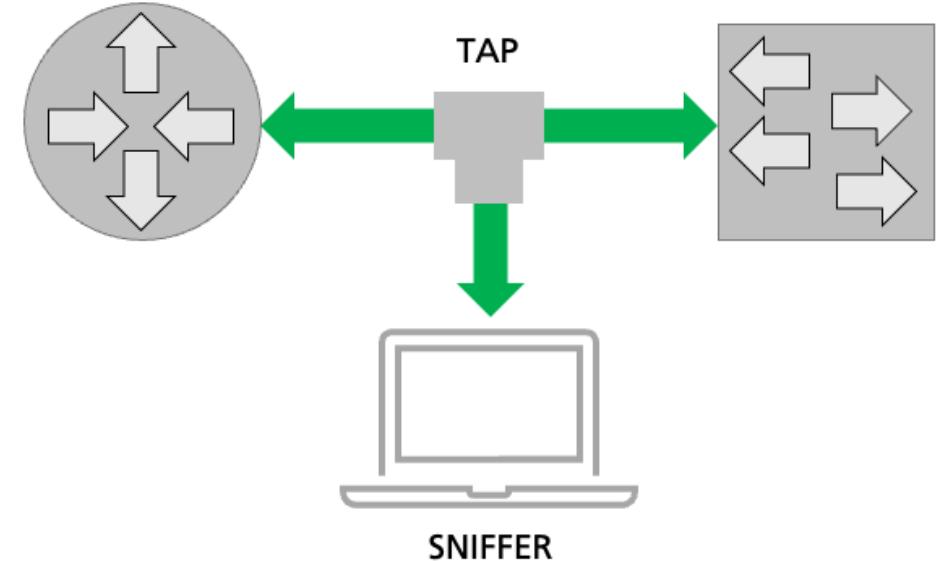
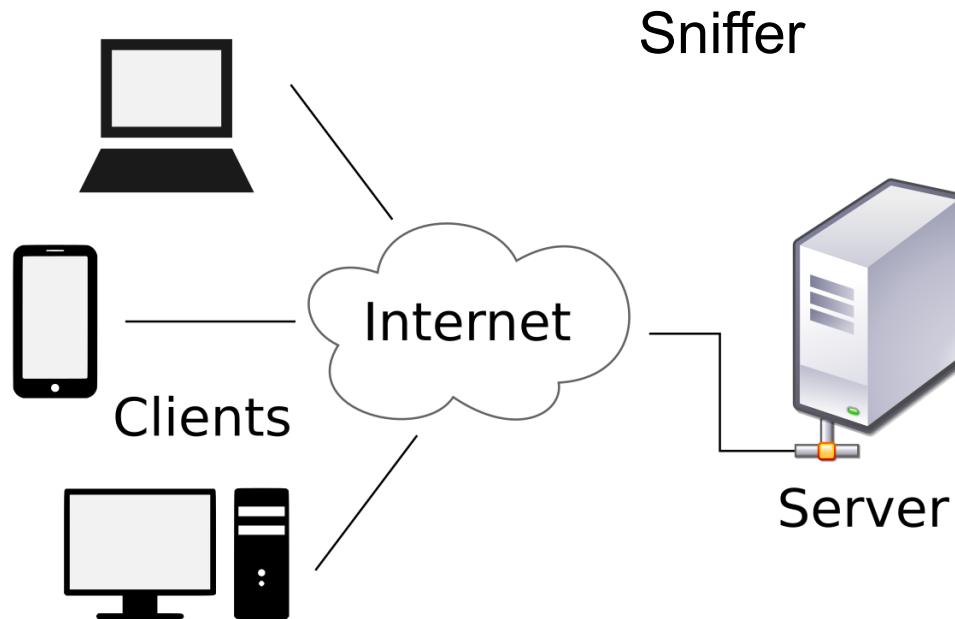
46 bytes fino a 1500 sono dati! I pacchetti troppo corti devono essere riempiti fino ad almeno 46 bytes

4 bytes (Frame Check sequence)

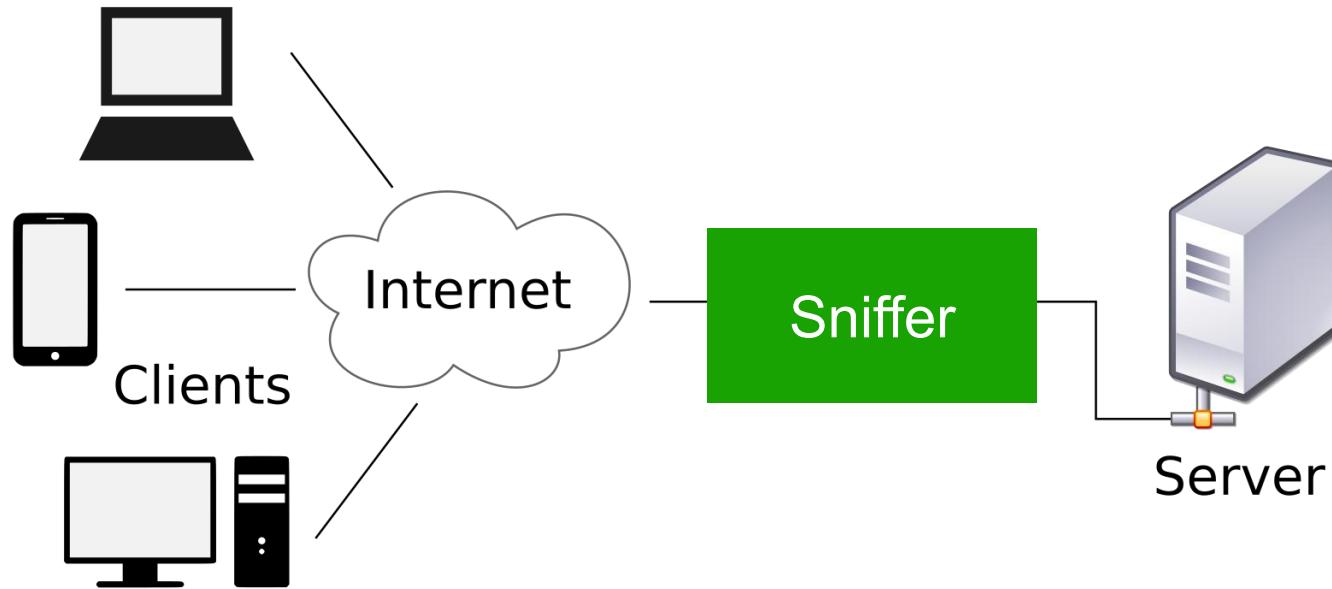
Sniffing della rete

tcpdump, tool più famoso per fare sniffing

Wireshark il più semplice



Sniffing della rete (in serie)



A invia richiesta (Q) a B e B risponde con il flusso di risposta (A)

- 1) A riceve la risposta
- 2) A non riceve la risposta
 - 1) B non ha ricevuto la richiesta
 - 2) B ha ricevuto la richiesta e ha inviato la risposta che si è fermata «sulla rete»

Logaritmo in base 2

Che cosa è il logaritmo di un numero, esempio n

Il logaritmo comporta il concetto di base del logaritmo

Esempio se

x è il logaritmo in base 10 di 1000, nel nostro caso 3

**Le usuali calcolatrici forniscono il logaritmo in
base 10**

**In informatica è usuale il logaritmo in base 2. Per
calcolarlo, nel caso fosse troppo complesso, si fa**

ARP

The Address Resolution Protocol uses a simple message format containing one address resolution request or response. The packets are carried at the data link layer of the underlying network as raw payload. In the case of Ethernet, a 0x0806 EtherType value is used to identify ARP frames.

The size of the ARP message depends on the link layer and network layer address sizes. The message header specifies the types of network in use at each layer as well as the size of addresses of each. The message header is completed with the operation code for request (1) and reply (2). The payload of the packet consists of four addresses, the hardware and protocol address of the sender and receiver hosts.

The principal packet structure of ARP packets is shown in the following table which illustrates the case of IPv4 networks running on Ethernet. In this scenario, the packet has 48-bit fields for the sender hardware address (SHA) and target hardware address (THA), and 32-bit fields for the corresponding sender and target protocol addresses (SPA and TPA). The ARP packet size in this case is 28 bytes.

Internet Protocol (IPv4) over Ethernet ARP packet

Octet offset	0	1
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

$\tau = \tau^{\mu \nu}$

$g_{\mu\nu}$

Hardware type (HTYPE)

This field specifies the network link protocol type. Example: Ethernet is 1.[2]

Protocol type (PTYPE)

This field specifies the internetwork protocol for which the ARP request is intended. For IPv4, this has the value 0x0800. The permitted PTYPE values share a numbering space with those for EtherType.[2][3]

Hardware length (HLEN)

Length (in octets) of a hardware address. Ethernet address length is 6.

Protocol length (PLEN)

Length (in octets) of internetwork addresses. The internetwork protocol is specified in PTYPE. Example: IPv4 address length is 4.

Operation

Specifies the operation that the sender is performing: 1 for request, 2 for reply.

Sender hardware address (SHA)

Media address of the sender. In an ARP request this field is used to indicate the address of the host sending the request. In an ARP reply this field is used to indicate the address of the host that the request was looking for.

Sender protocol address (SPA)

Internetwork address of the sender.

Target hardware address (THA)

Media address of the intended receiver. In an ARP request this field is ignored. In an ARP reply this field is used to indicate the address of the host that originated the ARP request.

Target protocol address (TPA)

ARP

Con il protocollo ARP posso scoprire chi ha uno specifico indirizzo IP

Ma come posso scoprire la topologia della mia rete?

In particolare quante schede di rete / device sono presenti nella mia rete locale?

Ma come faccio a andare su internet, o meglio, come posso comunicare con un dispositivo che si trova su un'altra rete locale?

Posso scoprire la topologia di una rete locale cui non appartengo?

Cerchiamo gli indirizzi MAC della mia rete locale

Conoscere il vostro MAC address

Conoscere il vostro IP address

**Inviare una richiesta ARP a tutti gli indirizzi IP definiti
tramite la maschera di sottorete (in genere
255.255.255.0 e quindi i 256 indirizzi IP che ricavate
dal vostro IP sostituendo al byte meno significativo di
valori da 0 a 255)**

**Per le richieste di cui avete risposta, allora
conoscerete l'associazione MAC address, indirizzo IP**

Client deve comunicare con un server WEB (A => B)

A conosce (nota bene che A non è l'utente ma è il browser dell'utente poiché l'utente spesso non conosce il suo IP e il suo MAC)

, Cosa fa il browser? Cosa fa lo strato di rete del vostro PC?

Client deve comunicare con un server WEB (A => B) ,

Diamo per scontato che la porta TCP del server sia la 443 (protocollo https)

Per prima cosa devo ottenere *IP_B* da *Nome di Dominio_B* Invio una richiesta DNS a chi?

Omettiamo /etc/hosts e cache intermedi: devo inviare all'IP dei DNS che sono stati configurati sulla mia macchina una richiesta DNS

Noto l'IP del dns (IP_dns), cosa fa il browser/rete?

Invia una richiesta ARP in broadcast per ottenere il MAC address del gateway che lo porterà all'IP del dns

Ottenuto il MAC Address invia al MAC address una richiesta su protocollo DNS che contiene il nome di dominio di B

Riceve tramite gateway la risposta del server DNS che gli indica l'IP corrente di google

Noto l'IP di B

Richiesta ARP

Preparazione del pacchetto di richiesta (HTTP)

Invio del pacchetto di richiesta a B

Ricezione della risposta

Decodifica della risposta (HTTP)

Visualizzazione della risposta (interpretazione HTML)

Protocollo DNS??

See dns_packet_structure.pdf

Il protocollo HTTP

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

RFC dell'HTTP/1.1

Request = **Request-Line** ; Section 5.1
*((general-header ; Section 4.5
| request-header ; Section 5.3
| entity-header) CRLF) ; Section 7.1
CRLF
[message-body] ; Section 4.3

```
| POST ; Section 9.5
| "PUT"
| "DELETE"
| "TRACE"
| "CONNECT" ; Section 9.9
| extension-method ; Section 9.7
extension-method = token ; Section 9.6 ; Section 9.8
```

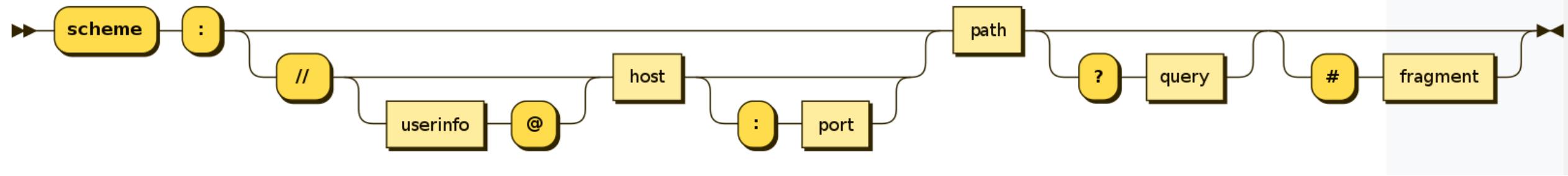
HTTP - request

Method URI Version (es: GET /pag1/pag2.html

HTTP/1.1)

CRLF

body



HTTP - answer

Version error code (HTTP/1.1 200 OK)

Headers.... (CRLF)

CRLF

body

I layer di rete che portano all'HTTP

Application protocols (SMTP, POP3, IMAP, DNS, FTP, SSH, HTTP, ...)

TCP / Transport

IP / ~~internet~~
<https://datatracker.ietf.org/doc/html/rfc7231>

Data Link Layer / ethernet

I layer di rete che portano al TRIS

Protocollo applicativo dello sviluppatore del TRIS (es: A click nella casella 1,2, B click nella casella 0,0, ecc)

Application protocols (SMTP, POP3, IMAP, DNS, FTP, SSH, HTTP, ...)

TCP / Transport

IP / internet

Data Link Layer / ethernet

Method	Description	Sec.
GET	Transfer a current representation of the target resource.	4.3.1
HEAD	Same as GET, but only transfer the status line and header section.	4.3.2
POST	Perform resource-specific processing on the request payload.	4.3.3
PUT	Replace all current representations of the target resource with the request payload.	4.3.4
DELETE	Remove all current representations of the target resource.	4.3.5
CONNECT	Establish a tunnel to the server identified by the target resource.	4.3.6
OPTIONS	Describe the communication options for the target resource.	4.3.7
TRACE	Perform a message loop-back test along the path to the target resource.	4.3.8

HTTP

GET /pag1/index.html HTTP/1.1
Host: www.mioweb.it

Questo deriva dalla richiesta scritta nella barra di indirizzo del browser

<http://www.mioweb.it/pag1/index.html>

Vediamo lo sniffing di una comunicazione WEB

Con docker metto in esecuzione un server web

```
docker run --rm -it -p 8888:80 -v ./usr/local/apache2/htdocs/ httpd:latest
```

<http://10.7.0.26:8888/>

**sudo tcpdump -i any -s 2000 -X host 10.7.0.26 and
port 8888**

**sudo tcpdump -i any -s 2000 -A host 10.7.0.26 and
port 8888**

Esempio di richiesta di un browser a un sito web

GET / HTTP/1.1 <metodo> <risorsa(url)> <versione di http>

Host: 10.7.0.26:8888 <indirizzo host del server>

**User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:123.0)
Gecko/20100101 Firefox/123.0**

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

If-Modified-Since: Wed, 20 Mar 2024 08:58:33 GMT

If-None-Match: "181-61413cba33cca"

La risposta del server web

**HTTP/1.1 304 Not Modified <protocollo http
utilizzato> <codice di errore>**

Date: Wed, 20 Mar 2024 09:14:15 GMT

Server: Apache/2.4.58 (Unix)

Last-Modified: Wed, 20 Mar 2024 08:58:33 GMT

ETag: "181-61413cba33cca"

Accept-Ranges: bytes

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

HTTP/1.1 200 OK

Date: Wed, 20 Mar 2024 09:22:16 GMT

Server: Apache/2.4.58 (Unix)

Last-Modified: Wed, 20 Mar 2024 08:58:33 GMT

ETag: "181-61413cba33cca"

Accept-Ranges: bytes

Content-Length: 385

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Content-Type: text/html

<!DOCTYPE html>

<html>

I codici di risposta

- **1xx: Informational** - Request received, continuing process
- **2xx: Success** - The action was successfully received, understood, and accepted
- **3xx: Redirection** - Further action must be taken in order to complete the request
- **4xx: Client Error** - The request contains bad syntax or cannot be fulfilled
- **5xx: Server Error** - The server failed to fulfill an apparently valid request

Il protocollo HTTP, i problemi derivanti dalla condizione STATELESS

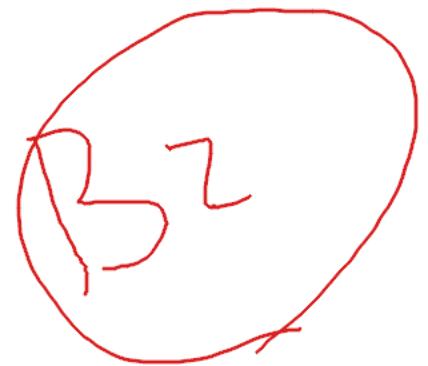
Nel protocollo basic http, l'autenticazione viaggia sempre con le richieste

`http://username:password@www.server.it/risorsa.html`

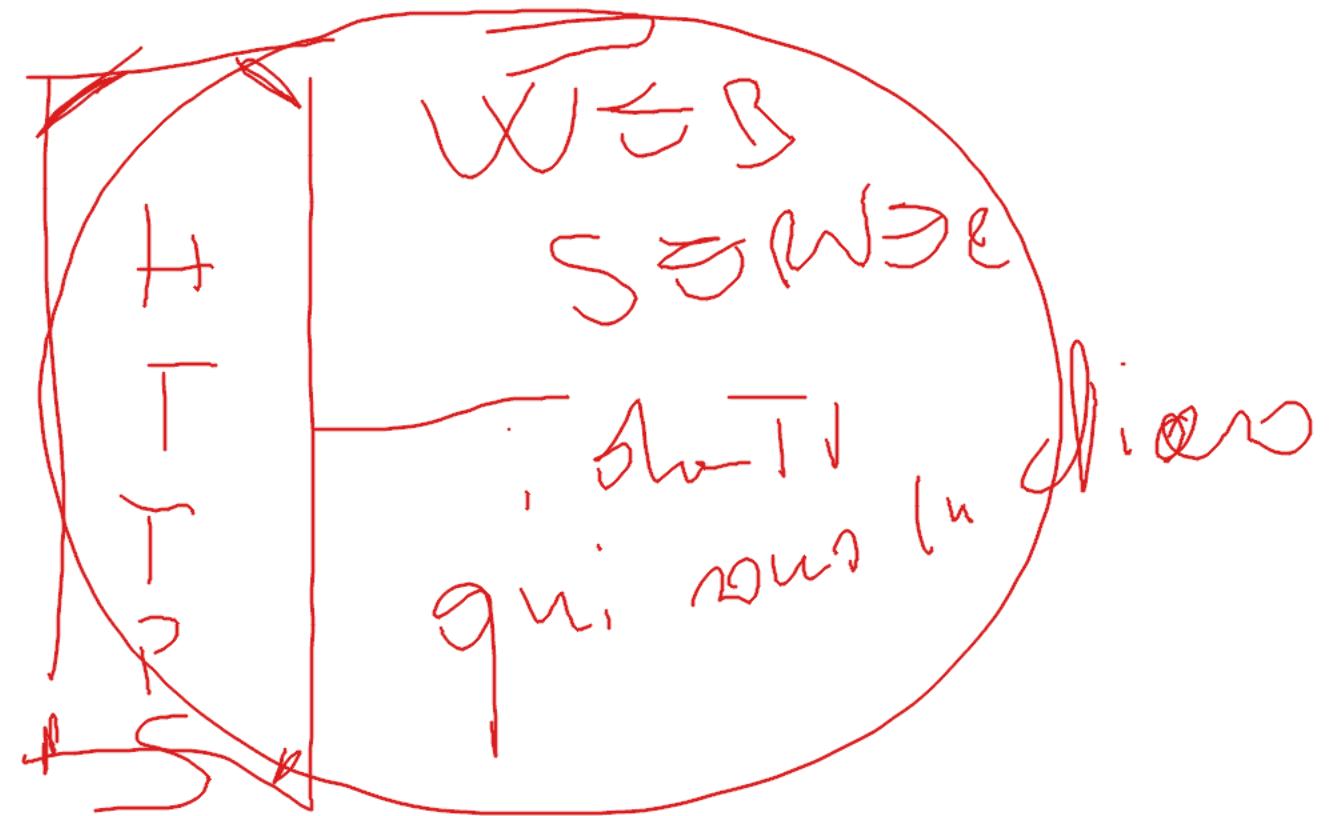
In ogni richiesta il web server verifica username e password, ogni volta!!!

C'è un modo per evitarlo? Come viaggiano usualmente username e password?

Usiamo google chrome / firefox e attiviamo l'analisi della rete



HITIPS



BR

HTTP(S)

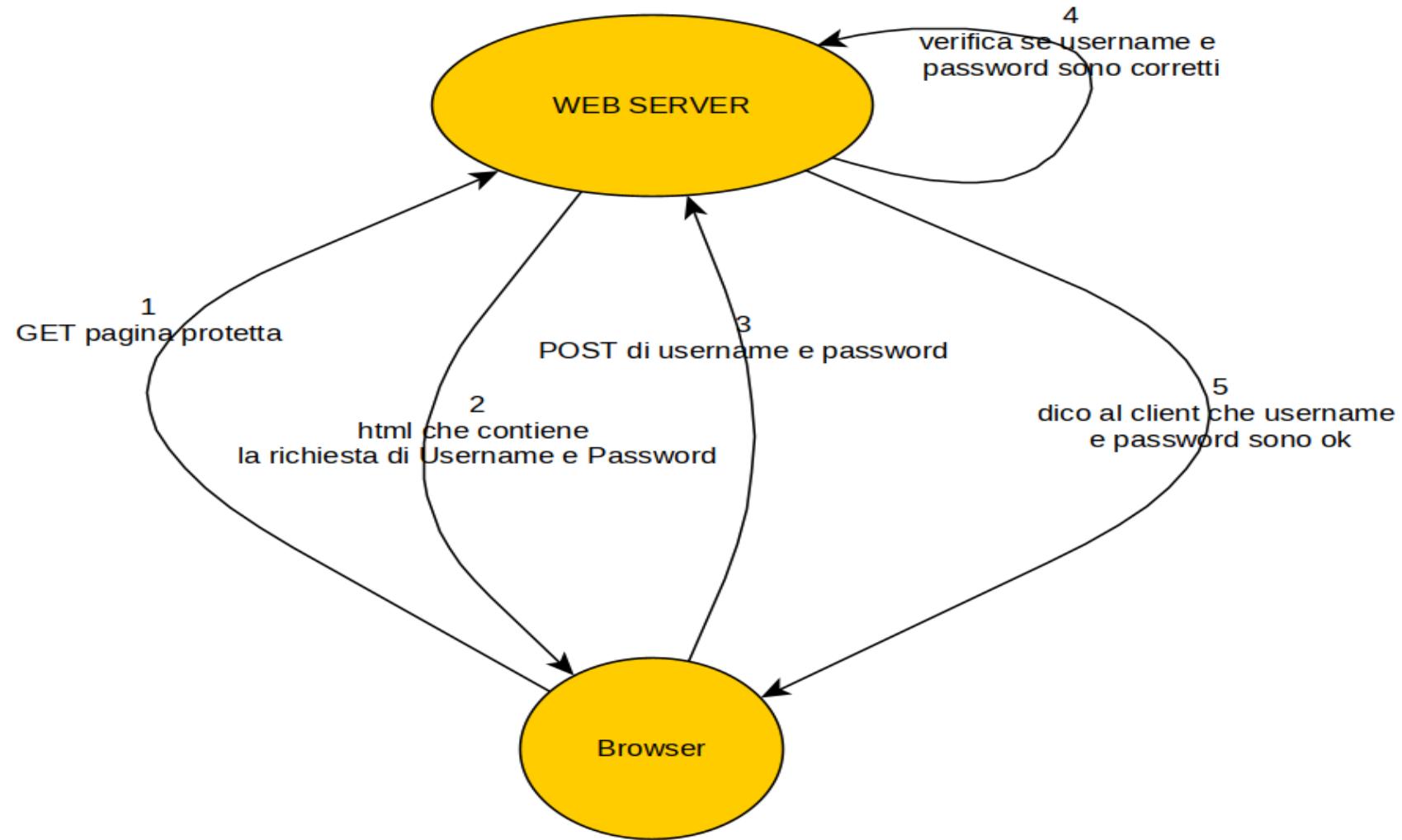
HTTP(S)
+
NON
ENCRYPTED

HTTP

Chaos!

XSS
SQL INJECTION

Gestione dell'autenticazione



autenticazione

Il client chiede pagina protetta

Il server propone form con username e password

**Il client inserisce username e password e invia al
server**

Il server verifica username e password

**Il server dice al client che è stato autenticato
Che il client?**

Va al punto 1 MA, dato che il protocollo HTTP è stateless, il server non ricorda che l'utente è già stato autenticato!! Come risolvere questo problema?

autenticazione

Il client chiede pagina protetta

Il server propone form con username e password

Il client inserisce username e password e invia al server

Il server verifica username e password

**Il server dice al client che è stato autenticato e gli dà un
“token(testimone)”**

Che fa il client?

Va al punto 1 e fornisce al server, nella richiesta, anche il token

**Il server verifica che il token sia “corretto” e fornisce la
risorsa protetta al client**

Il token

Come è fatto il token?

Innanzitutto deve essere non facilmente copiabile
Crittografia, sì ma non è evidentemente sufficiente

Passibile di attacchi di “retransmit” (dove il token viene copiato e inviato da un altro PC)

Potremmo mettere username o altro codice identificativo dell'utente poiché il server deve sempre gestire Autenticazione e Autorizzazione

Autentico le credenziali

In base alle credenziali autenticate fornisco o no un particolare servizio

Nel token aggiungo una data di scadenza

Il vantaggio del token cifrato è che solo il server che conosce la chiave di cifra e decifra può decifrarlo

Altre informazioni che potrebbero essere di uso per il server

E il logout???

Non esiste!!!
Il server dice al client

Per favore butta via il token

I meccanismi di gestione dei token

In che modo un browser può inviare il proprio token al server?

Ricorda che A termine dell'autenticazione il server fornisce un token al client(browser)

NB: il token null'altro è che una stringa

Ora il browser invia una richiesta al server

In che parte della richiesta mette il token?

Struttura della richiesta web

Struttura della richiesta web

http://mioserver.it/pagina.html?token=<il token>

Metto il token come parte del campo QUERY dell'HTTP

http://mioserver.it/<token>/pagina.html

Il web server sa che il primo elemento del path della richiesta non è file system ma è il token. Lo toglie, lo verifica e agisce di conseguenza

Ricordate che il protocollo http prevede una riga di intestazione, seguita dagli header http, un insieme di righe che contengono coppie chiave/valore

GET /pagina.html HTTP/1.1

Host: mioserver.it

Accept-encoding: utf-8

Token: <token> //aggiungo un header all'http e l'header si chiama "Token"

I cookie

Usualmente il server web e i browser utilizzano un header particolare, Cookie, per scambiarsi dati nel formato chiave/valore

Il token, spesso, è inviato dal server al client nel seguente formato (header http)

Set-Cookie: <nome del cookie>=<valore del cookie>, <durata del cookie>, <url al quale il cookie può essere inviato>, <flag che indica che il cookie è usabile solo dentro https>

Esempio da google

Set-Cookie:

Nome del cookie

_Secure3PSIDCC=

Valore del cookie

AKEyXzVVq5QtJb6i9_wnx1xQx6ZYrJUMmwyHMFdFb9e9A4wQDTRd1aoiW7Lge_xEsn9_NKWbGA;

Data di scadenza

expires=Thu, 20-Mar-2025 11:24:45 GMT;

La data di scadenza può anche essere del tipo: non appena chiudi il browser, perdi questo cookie

path=/;

Su quali pagine puoi rimandarmi questo cookie

Nome di dominio

domain=.google.com;

Su quali nomi di dominio puoi inviarmi il cookie

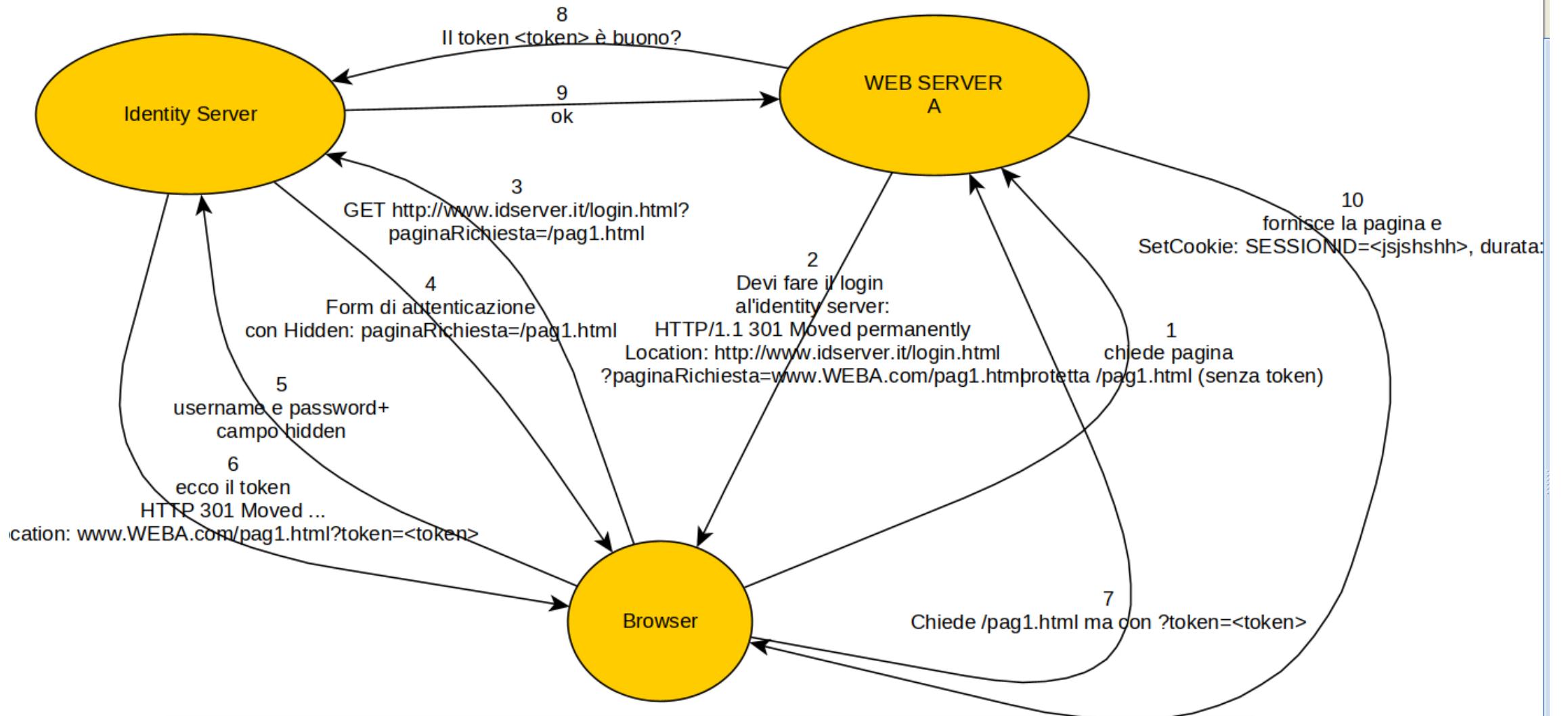
Secure; HttpOnly;

priority=high; SameSite=none

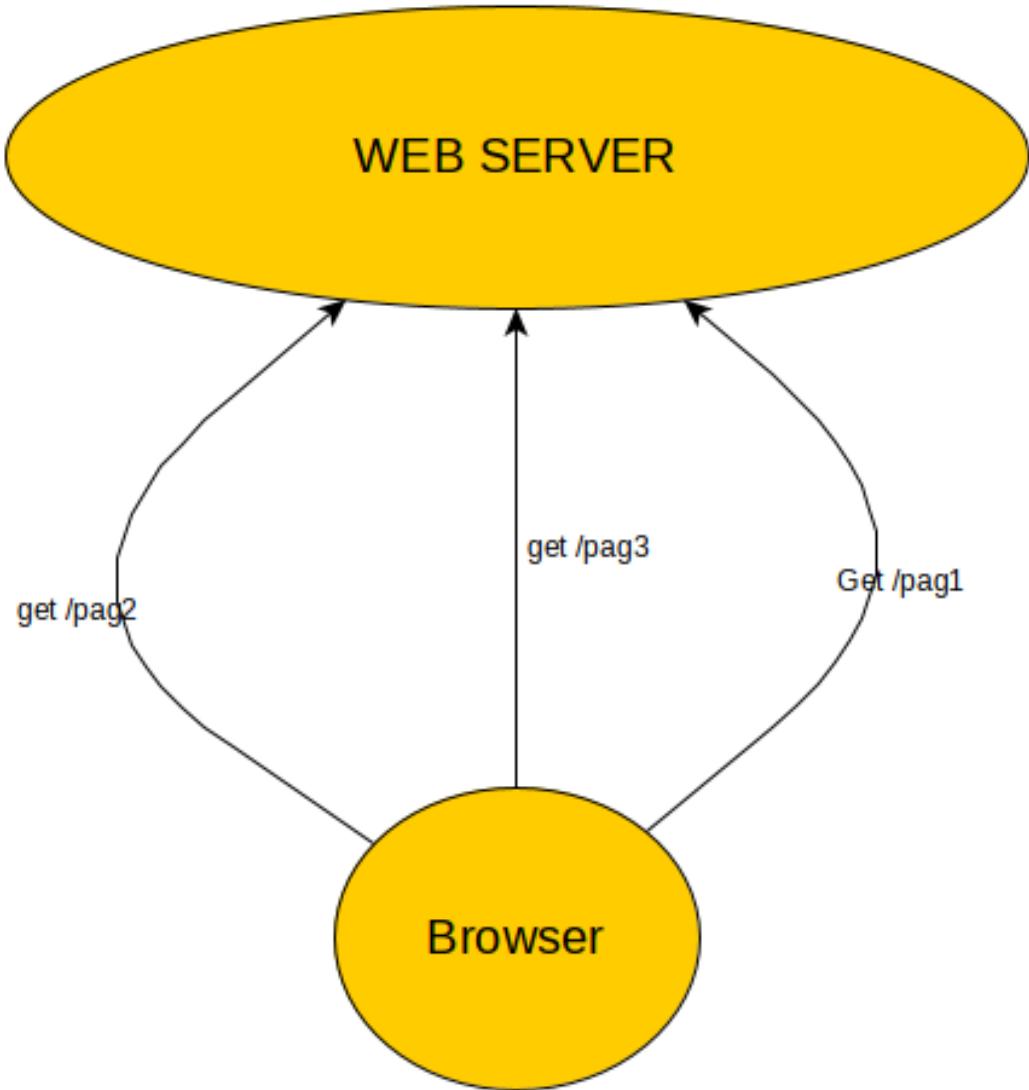
**quanto un browser riceve la risposta 301, prende
la riga di header successiva (Location) e
automaticamente va verso ls url indicata in**

HTTP/1.1 301 Moved Permanently
Location: https://www.example.org/
Content-Type: text/html
Content-Length: 174

```
<html>
<head>
<title>Moved</title>
</head>
<body>
=Moved=
<p>This page has moved to <a
href="https://www.example.org/">https://www.example.org/</a>.</p>
</body>
</html>
```



Nel caso di logout, il server invia una SetCookie: SESSIONID=""



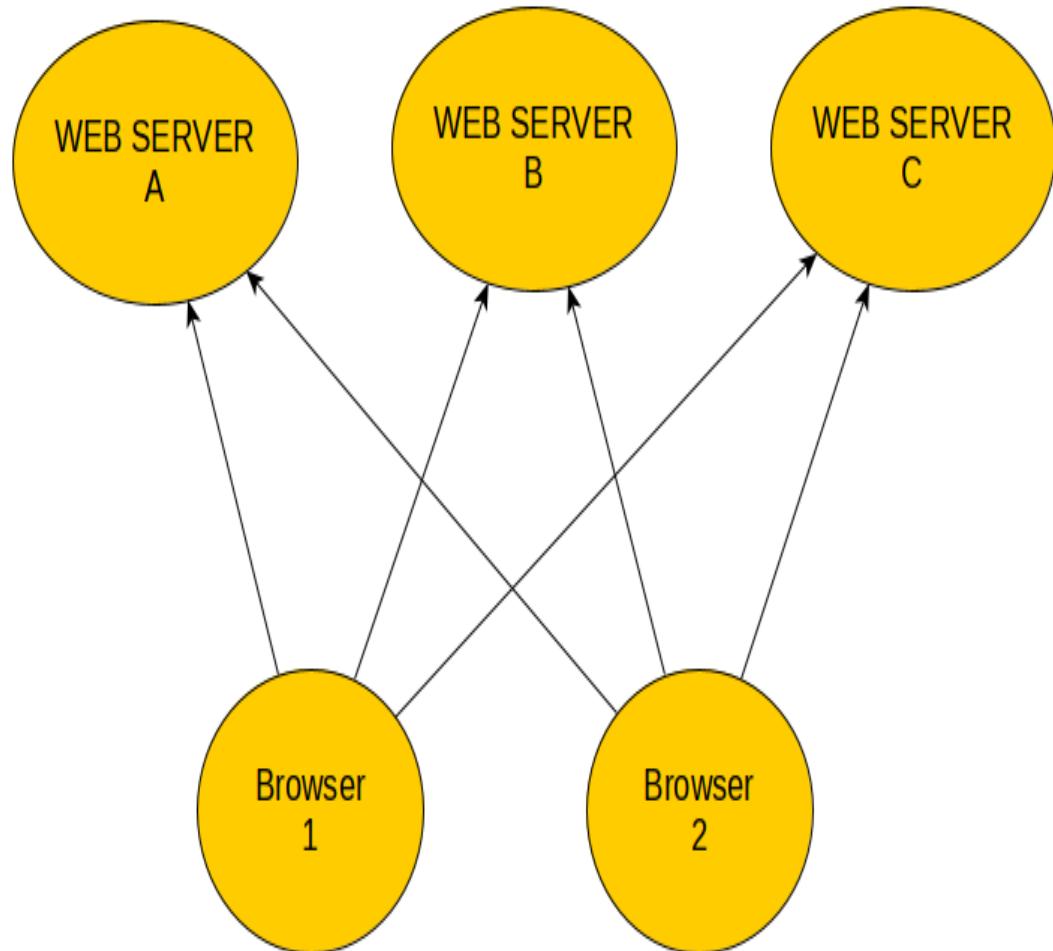
› è di utilizzare un cookie, che al quale appendiamo tutte le visite alle pagine visitate una pagina (es “/pag1”) che il browser abbia inviato un nome “tracker”

er” la stringa “;”pagina visitata (“;/pag1”) e nella risposta al browser i la nuova pagina>1/1/2100

erver riesce a “tracciare” tutte le visite da un browser il browser ma sa che lo stesso browser ha navigato sull’insieme di pagine tracciate

Assegnare un'identità unica a un browser

Qui la richiesta è



- È possibile riuscire a tracciare che il browser 1 ha navigato
 - Su pag1 di web A
 - Su pag2 di web B
 - Su pag1 di web C
 - Su pag2 di web A
 - E così via?
- Cioè, è possibile tracciare il comportamento di un browser (non identificabile in rete) in relazione a un sistema di servizi composto da iù siti WEB?

Uso mitm

Quindi metto un web server davanti a WebA, WebB e WebC

Il Mitm non è identificabile sulla rete. Lui intercetta tutte le comunicazioni e le inoltra al server di riferimento
Un browser va da WebA e riceve il cookie WA-01

Il MITM non è un web server quindi se il MITM aggiungesse alle Set-Cookie inviate da WebA anche un altro cookie, es: TRK-01 questo TRK-01 per il browser sarebbe legato esclusivamente a WebA

Cioè se lo stesso browser andasse verso WebB non presenterebbe TRK-01 poiché per lui TRK-01 appartiene a WEBA

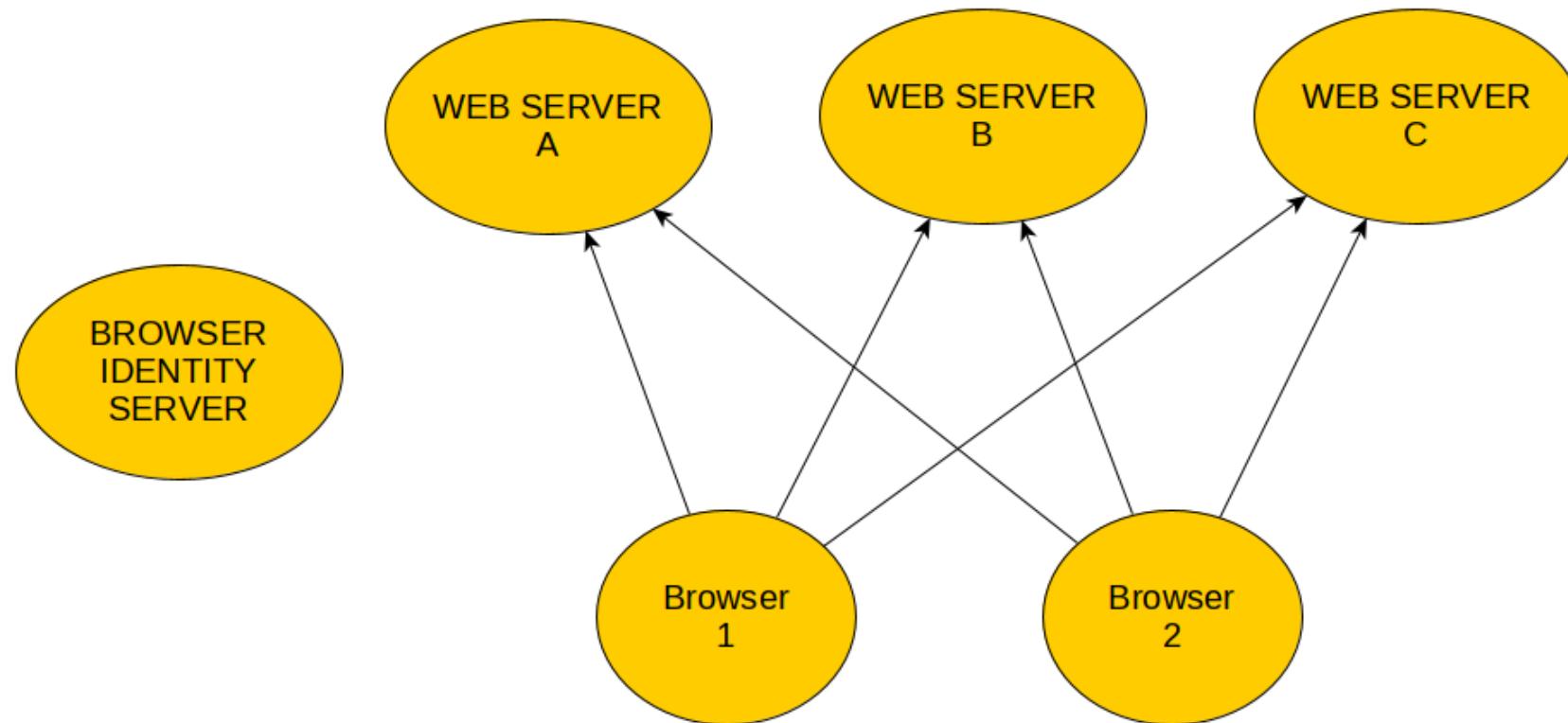
Uso mitm

Quindi metto un web server davanti a WebA, WebB e WebC

Se il Mitm fosse un web server?

Come potrebbe un browser indicare che vuole andare sulla pagina P1 del WebA e poi sulla pagina P1 del web B?
Dovrei rimappare tutte le pagine, in qualche modo, e quindi torno al caso del Web server singolo.

Aggiungiamo un nuovo web serve che ha il compito di assegnare un ID unico ai browser



Come assegnare id unico ai browser in un sistema multiservizi?

B è un browser
WA, WB, WC sono tre web server
BID è un ulteriore web server

Il cookie ID sia il cookie che ha lo stesso nome nei confronti dei tre server e il suo valore sarà lo stesso per i tre server

Cioè: una volta stabilità l'identità, il browser si presenterà verso WA, Wb e WC con un cookie che si chiama ID e ha valore, ad esempio, 81327A212 (numero unico)

Oppure potrei associare al cookie ID l'elenco delle pagine visitate, web per web, del tipo: ID=WA-P1,WA-P8,WB-P2,WC-P9

Vediamo prima i caso di ID unico e poi vediamo come l'ID può diventare l'elenco delle pagine

Assegnazione di un cookie ID con stesso valore per i tre siti

B => WA

Ha un cookie che si chiama ID?
Se sì allora emetti log(ID, pagina visitata)

Lo stesso per WB e WC

Se B ha un cookie che si chiama ID sia nei confronti di WA, di WC di WB, allora i tre log conterranno informazioni relative allo stesso browser

Il problema che non abbiamo ancora risolto è come posso assegnare a ID del browser lo stesso valore per i tre siti web

Assegnazione di un cookie ID con stesso valore per i tre siti

B => WA

Ha un cookie che si chiama ID?

No, non lo ha!!!

Faccio una redirect al sito BID, indicando anche che sono il WA

Dato che è una redirect a un sito web, il browser, se il sito BID gli ha assegnato un cookie, lo presenta!!!

Supponiamo che il cookie si chiami UID

Se BID riceve UID, allora significa che il browser ha già un ID unico assegnato (supponiamo 12345) e quindi invia una redirect di ritorno al sito originale (nel nostro caso WA) passando il valore dello UID come parametro della GET (campo query aggiungo: ?UID=12345)

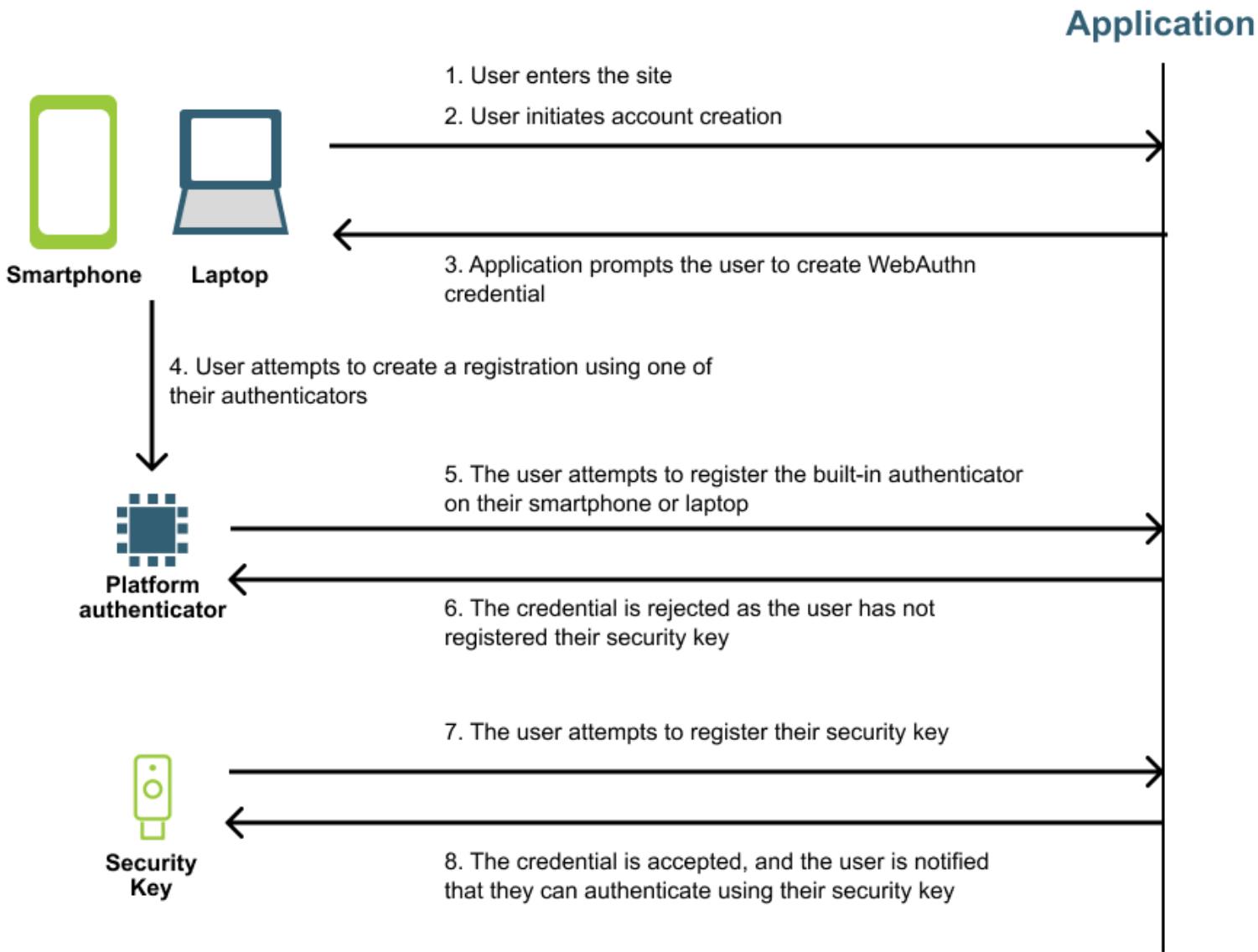
Altrimenti

Crea un valore random (unico, ad esempio, 12345) e lo assegna a UID facendo Set-CookieUID=12345. Lo stesso valore lo associa alla redirect di ritorno scrivendo in fondo alla url la parte quesy ?UID=12345

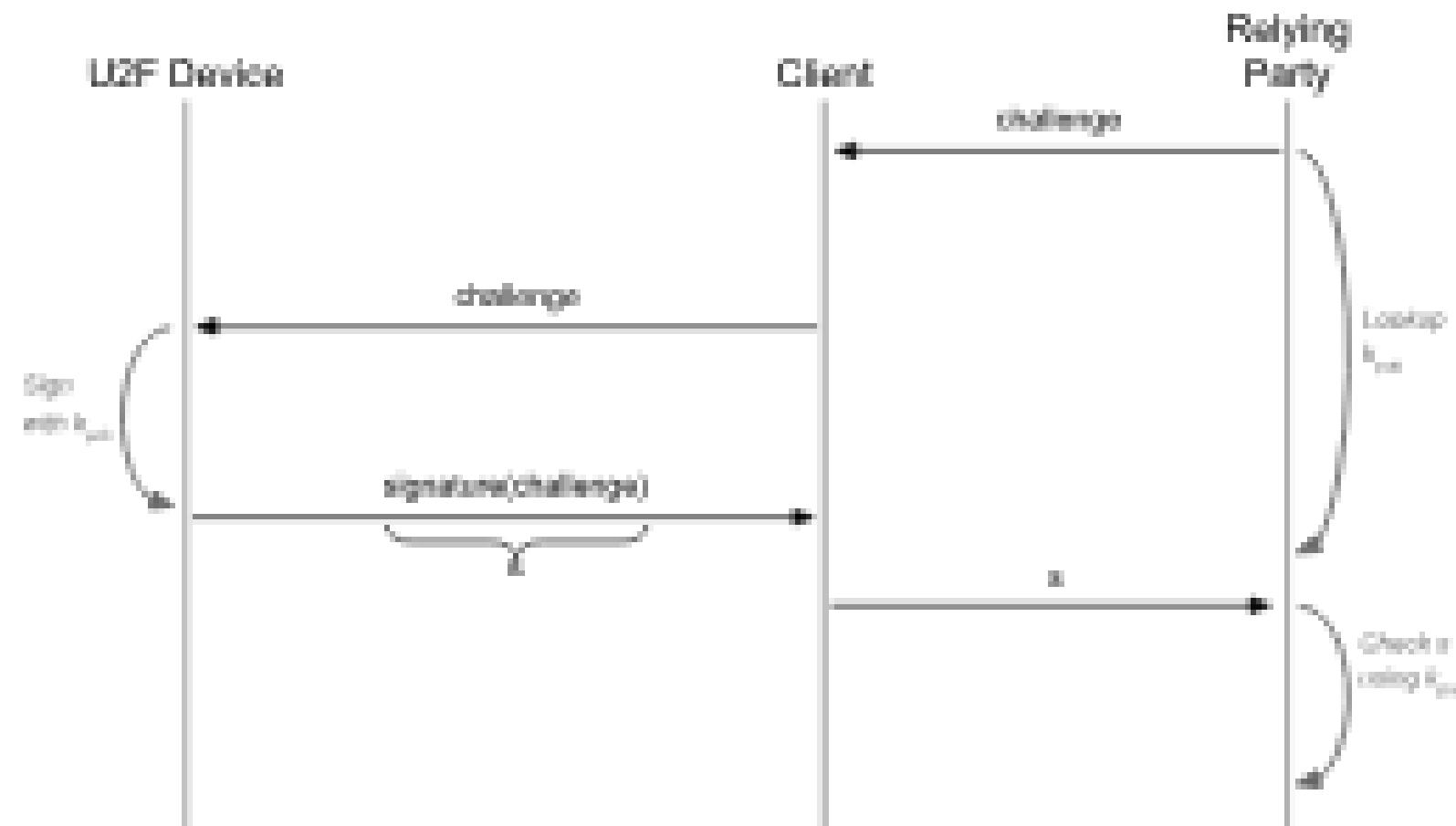
Clone di un sito

**CURL, ve lo dice il vostro collega
WGET, guardate il manuale**

Initial account registration with security key



https://developers.yubico.com/U2F/Protocol_details/Overview.html



Cifrario di Cesare

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

X Y Z A B C D E F G H I J K L M N O P Q R S T U V W

MESSAGGIO = CIAO

CIFRATO = ZFXL

I DUE PEER condividono lo stesso segreto

Nella crittografia simmetrica tutti i partecipanti condividono lo stesso segreto (la chiave di cifra e decifra)

Il segreto è utilizzato per cifrare e per decifrare il messaggio

A	B	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

X AND 0 => 0

X OR 0 => X

X XOR 0 => X

X XOR 1 => NOT X

X XOR X => 0

Una nota

In assembler INTEL per assegnare il valore 0 a un registro (AX) si usa

MOV AX, 0 => 2 cicli macchina

XOR AX, AX => 1 ciclo macchina

Se la cpu va a 4GHz, con la mov ne posso fare 2miliardi, con la XOR ne faccio 4 miliardi di assegnazione a 0 di un registro

$X \text{ xor } X \Rightarrow 0$

$Y \text{ xor } 0 \Rightarrow Y$

$R = Y \text{ xor } X \Rightarrow \text{cifra}$

$R \text{ xor } X = Y \text{ xor } X \text{ xor } X = Y \text{ xor } 0 = Y$

$R \text{ xor } X \Rightarrow Y \Rightarrow \text{decifra}$

$X \text{ xor } Y \text{ xor } X = X \text{ xor } X \text{ xor } Y = 0 \text{ xor } Y \Rightarrow Y$

Scrivere un programma PYTHON che a partire da una stringa la cifra con la tecnica XOR
Successivamente mostrare che la stringa cifrata, riapplicando lo stesso XOR, torna la
stringa originale

Per fare lo XOR utilizzate un solo valore: 57

Quindi data la stringa di esempio “Nel mezzo del cammin di nostra vita”, dovete fare per
ogni carattere della stringa lo xor con il valore 57

“N” xor 57, “e” xor 57, ...

Ottenendo una lista di numeri es: 78 (che è il codice ascii della lettera N) xor (si indica
con il simbolo ^) => $78 \wedge 57 = 119$

E così via per tutta la stringa.

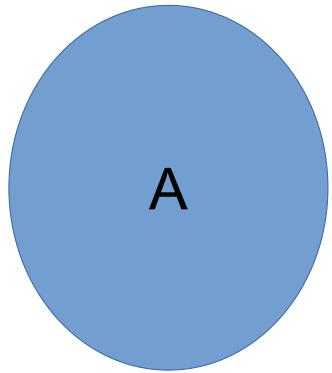
Al termine stampare la lista di numeri ottenuti

In fondo a partire dalla lista di numeri, riapplicare lo xor sempre con 57 e quindi ottenere
(ricostruendola) la stringa originale

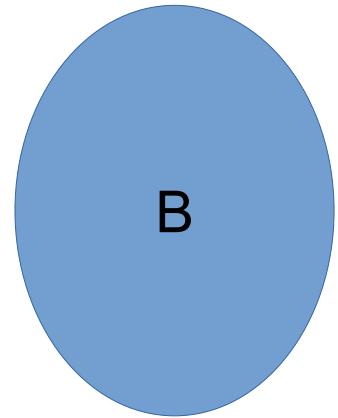
NB: potreste utilizzare `input("...")` in modo da leggere sia la stringa da cifrare, sia il valore
segreto da applicare come xor

```
str = input("Inserisci stringa: ")
key = int(input("Chiave: "))
# esempio "abc Ciao"
# per ogni carattere della stringa devo fare
xor
```

Crittografia riservatezza vs integrità



A vuole inviare un messaggio M a B ed essere sicuro che solo B lo possa leggere
Crittografia Simmetrica



Crittografia Asimmetrica

Crittografia Simmetrica

**Una sola chiave (K) per cifrare e decifrare
La chiave è condivisa tra tutti i partecipanti, 2 o**

più

$$C = \text{Cifra}(K, M)$$

Pro

$$M_{dec} = \text{Decifra}(K, C)$$

Veloce

Predisposta a cifrare messaggi di qualsiasi lunghezza (a blocchi)

Cons

Se un attaccante entra in possesso della chiave, allora tutte le comunicazioni e i messaggi sono compromessi

Crittografia simmetrica

Cifra a blocchi?

Per poter cifrare un messaggio di dimensioni D, devo decomporlo in tanti blocchi ognuno di essi lungo quanto la dimensione utilizzata dall'algoritmo di cifra (dalla chiave, ma se la chiave è troppo corta, molte implementazioni la duplicano al fine di ottenere la lunghezza desiderata)

ES

AES-128

Significa che sia chiave, sia il blocco minimo che può essere cifrato/decifrato è lungo 128 bit

Riservatezza vs Integrità

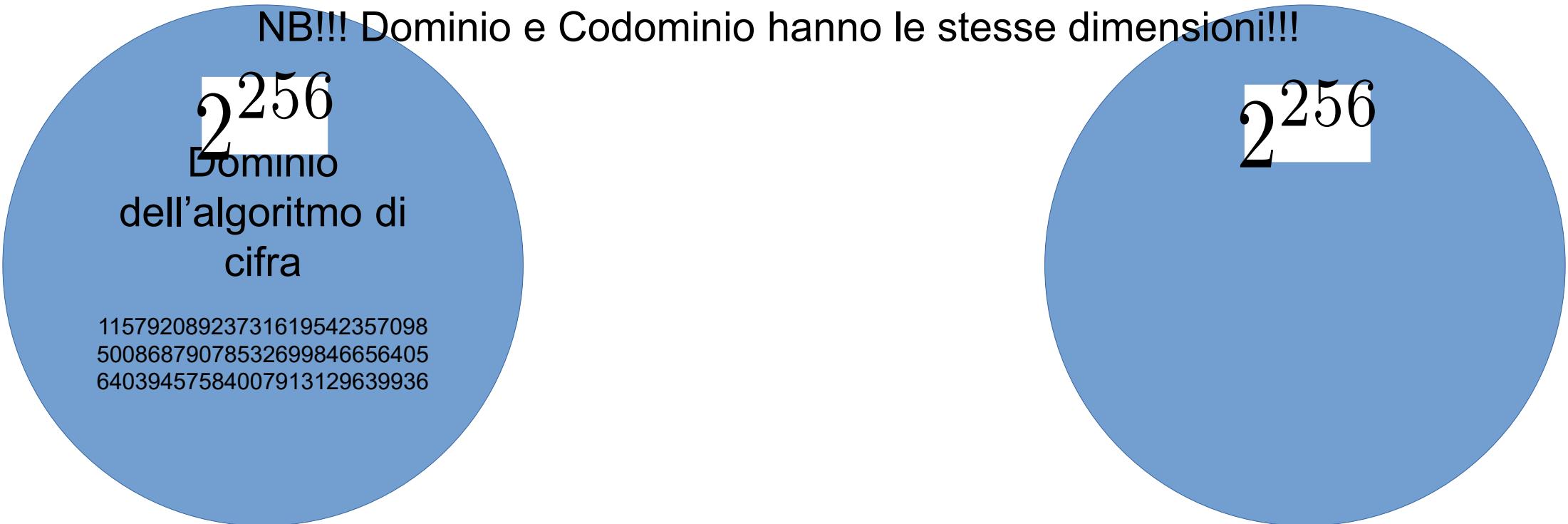
Es: AES-256

Spazio dell'algoritmo di cifra: 256 bit, $256/8 \Rightarrow 32$

In input all'algoritmo di cifra posso avere un messaggio lungo 32 caratteri

Per cifrarne uno più lungo lo dovrò frammentare in blocchi di 32 caratteri

NB!!! Dominio e Codominio hanno le stesse dimensioni!!!


$$2^{256}$$

Dominio
dell'algoritmo di
cifra

11579208923731619542357098
50086879078532699846656405
64039457584007913129639936

$$2^{256}$$

Cifra simmetrica

$$C = Cifra(K^a, M)$$

$$C \rightarrow C'$$

Un attaccante modifica un bit
del messaggio cifrato
ottenendo un nuovo messaggio
C'

Decifra(K^a, C') →?

Esempio di crittografia

La nostra chiave condivisa

ElChiringuito

Openssl è la nostra soluzione

Per cifrare un messaggio mess1.txt

```
openssl enc -e -in mess1.txt -out mess1.cfr -aes-256-cbc
```

Per decifrare il messaggio cifrato

```
openssl enc -d -in mess1.cfr -out mess1.dec -aes-256-cbc
```

Aggiungere AAAAAAAA al posto 7000 di alice.txt

```
head -c 7000 alice.txt >alice1.txt  
echo -n AAAAAAAA >>alice1.txt  
tail -c +7000 alice.txt >> alice1.txt
```

I comandi effettivi per cifrare in modalità RAW (come da modello matematico della crittografia simmetrica)

Genera il messaggio (16 byte)

```
echo -n Ciao come va,ok? >ciframi.txt
```

Cifra con chiave da 32 byte

```
openssl enc -e -aes-256-cbc -in ciframi.txt -out ciframi.enc -K  
00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff -iv  
0102030405060708090a0b0c0d0e0f10 -nopad -nosalt
```

Per Decifrare

```
openssl enc -d -aes-256-cbc -in ciframi.enc -out ciframi.dec -K  
00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff -iv  
0102030405060708090a0b0c0d0e0f10 -nopad -nosalt
```

Il messaggio decifrato è identico al messaggio cifrato

Messaggio cifrato: ciframi.enc

Scrivere un programma python che modifica un solo bit (casuale) del file che deve essere passato come parametro

Esempio di applicazione

`python3 randombit.py ciframi.enc`

Un codice per modificare un solo bit di un messaggio cifrato

```
import sys
import random

if len(sys.argv)<2:
    print("Usage: python randombit.py <file name>")
    sys.exit(1)

nomeFile = sys.argv[1]
data=None
with open(nomeFile, 'rb') as f:
    data = f.read()

    # devo modificare un solo bit!!
    # 1) scelgo il byte da modificare
    pos = random.randint(0, len(data) - 1)
    byte = data[pos]
    # 2) scelgo il bit da modificare
    bit = random.randint(0, 7)
    # Supponiamo di aver scelto il bit 3, come
    # faccio a modificare il bit 3 di byte?
    byte ^= (1 << bit)
    # 3) ricostruisco il byte modificato
data = data[:pos] + bytes([byte]) + data[pos + 1:]

with open(nomeFile, 'wb') as f:
    f.write(data)
print(f"Modified byte at position {pos}, bit {bit} in file {nomeFile}.")
sys.exit(0)
```

Verifica

**Eseguendo il codice python su file ciframi.enc
(file cifrato) e quindi modificando un singolo bit,
la decifra non dà errori (ovviamente trattandosi di
una codifica RAW) e fornisce un risultato
completamente differente dal messaggio
originale**

Crittografia simmetrica

**Supponiamo che il messaggio sia molto breve,
esempio**

A

Allungo il messaggio con un “padding”

Riempio fino a 32 (nel caso di 256 bit) con valori casuali

Note sulla crittografia simmetrica

padding?

Senza padding

Impossibile conoscere la dimensione del messaggio originale

Passbile di attacchi che modificano il pacchetto cifrato poiché dominio e codominio sono delle stesse dimensioni e quindi a qualsiasi punto del codominio corrisponde comunque un messaggio

Con padding

Dato il messaggio 011101100111010101011001, come posso cifrarlo e poi decifrarlo ottenendo il messaggio originale e evitando attacchi che modifichino il messaggio cifrato?

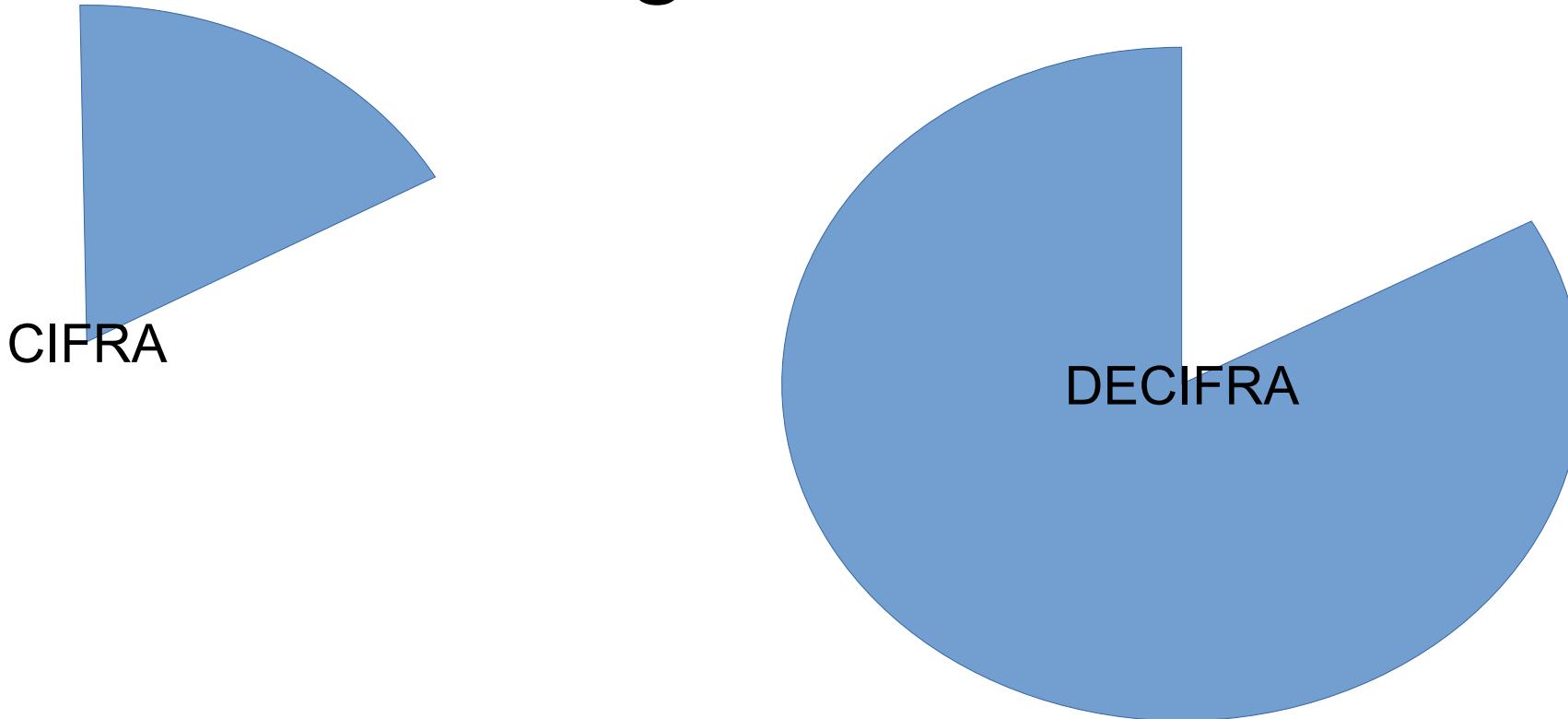
$M=011101100111010101011001 + 29 \text{ byte non utilizzati}$

Nel pacchetto ci sarà

Dimensione di M (in testa)

M (a seguire)

Crittografia asimmetrica



Numero di ore dell'orologio

32317006071311007300714876688669951960444102669715484032130345427524655138867890893197201411522913463688717960921898019494119559150490920
95088152386448283120630877367300996091750197750389652106796057638384067568276792218642619756161838094338476170470581645852036305042887575
8915410658086075523991239303855219143338966834242068497478656456949485617603532632205807780565933102619270846031415025859286417711672594
36037184618573575983511523016459044036976132332872312271256847108202097251571017269313234696785425806566979350459972683529986382155251663
89437335543602135433229604645318478604952148193555853611059596230656

Crittografia asimmetrica

Ogni partecipante ha una chiave pubblica e una chiave privata

Esempio, il peer A avrà:

Ovviamente, K_A^{pub} deve privare di EVE restare nella sola disponibilità di A, mentre la chiave pubblica è nella disponibilità di tutti

Crittografia asimmetrica

A e B si scambiano le chiavi pubbliche a tempo 0

Cioè in un momento in cui la sicurezza è massima

Esempio:

Siamo al PUB e A dà la sua chiave pubblica a B e viceversa

Crittografia asimmetrica

A, per inviare il messaggio M a B

B, per decifrare il messaggio cifrato inviato da A a

$$C = \text{Cifra}(K_{pub}^B, M)$$

$$M_{dec} = \text{Decifra}(K_{pri}^B, C)$$

Esempio RSA

- Attualmente le chiavi ammesse devono essere almeno 2048 bit, nel futuro cresceranno ancora
- Forziamo 3 come esponente per la chiave pubblica
 - Generazione della coppia di chiavi
 - openssl genrsa -out FAprivkey.pem -3 2048
 - Questo genera la chiave privata!!!
 - Per stampare i contenuti
 - openssl rsa -in FAprivkey.pem -text -noout
 - La chiave pubblica è un “di cui” della chiave privata e la si estrae dalla privata
 - openssl rsa -in FAprivkey.pem -out FApubkey.pem -pubout -RSAPublicKey_out

Algoritmo RSA

Genera due numeri primi p e q. Se RSA 2048, p e q devono essere numeri a 1024 bit

Poni $n=p^*q$

Scegli un esponente pubblico (e). La chiave pubblica sarà la coppia e, n

Calcola d (esponente chiave privata) a partire da p e q

La chiave privata sarà la coppia d, n

Pubblica e privata

openssl genrsa -out FAprivkey.pem -3 2048

Genera una chiave privata utilizzando 3 come esponente della chiave pubblica

**openssl rsa -in FAprivkey.pem -out
FApubkey.pem -pubout -RSAPublicKey_out**

Estrae la chiave pubblica (esponente e modulo) dalla chiave privata

**NB: chiave pubblica: esponente (e) e n (base del
modulo)** **NB: chiave privata: esponente (d) e n
(base del modulo)**

Per cifrare e poi decifrare, il comando si basa su

`openssl pkeyutl -h`

`openssl pkeyutl -encrypt -inkey FApubkey.pem -pubin -in messRSA.txt -out MessaForFA.dat`

`openssl pkeyutl -decrypt -inkey FAprivkey.pem -in MessaForFA.dat -out MessaForFA.dec`

Si basa sulla f $(m^e)^d = m \pmod{n}$

La difficoltà è trovare e e d che null'altro sono che la chiave pubblica e la chiave privata

È come se nell'aritmetica tradizionale d fosse il reciproco di ee quindi $m^{(e*1/e)}=m^1=m$

$$(m^e)^d = k * n + m$$

E cosa posso fare per verificare che il mio messaggio non sia stato alterato?

Uso della crittografia asimmetrica (non è una reale verifica di non alterazione, ma costituisce la sua base)

A vuole inviare un messaggio a B e vuole essere certo
Che solo B possa leggerlo
Che B sappia che il messaggio è stato inviato da A

Mutua autenticazione

A cifra prima con la sua privata e poi con la sua

$$Cifra(K_{pub}^B(Cifra(K_{pri}^A, M)))$$

B prima decifra con la sua privata e quindi verifica che il messaggio sia diretto a lui e poi decifra con la pubblica di A per verificare che il messaggio sia stato inviato da A

E per quale motivo questo secondo metodo non funziona?

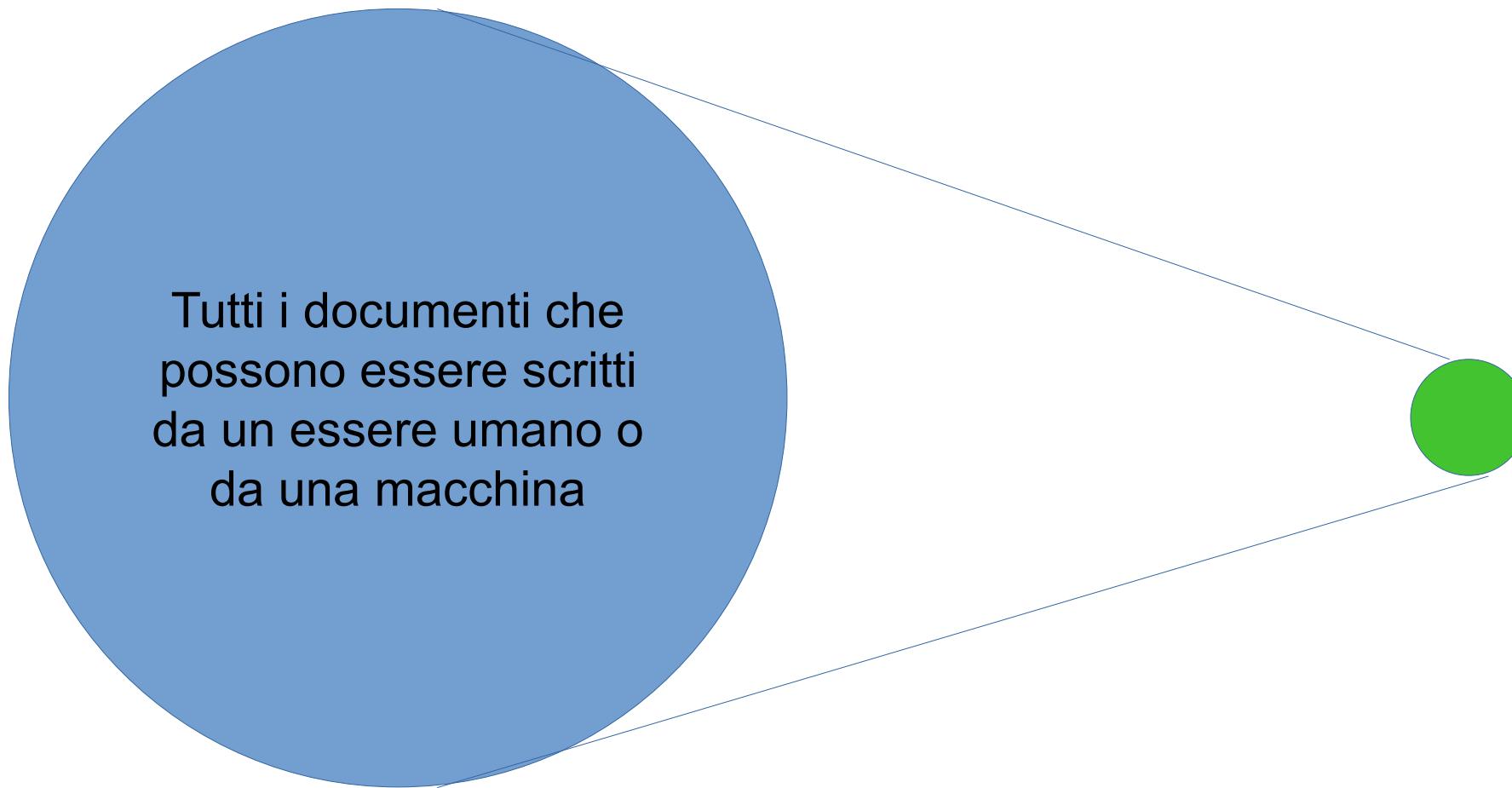
$$Cifra(K_{pri}^a, Cifra(K_{pub}^B, M))$$

Hash

L'HASH è una tecnica che consente di ricavare un numero (grande) da un documento.
L'hash si usa per il verificare che il documento sia inalterato

Caratteristica dell'hash

trasforma un elemento del suo dominio in un elemento di un codominio più piccolo del dominio originale ma pur sempre impraticabile da gestire con i sistemi di calcolo attuali



Esempio di funzione hash

**Sia M il documento visto come una sequenza di byte,
questo significa che è equivalente a un numero di
dimensioni molto elevate**

$$\text{hash} = f(M) \bmod n$$

Applico una funzione di trasformazione al numero M e poi ne calcolo il modulo (il resto della divisione) rispetto al numero n
In tal modo il valore risultante dall'hash è un numero compreso tra 0 e n

Il modulo

$a \pmod n \equiv b$ questo della divisione di a con n

Quindi posso esprimere a come
 $a = k * n + b$, k numero intero

Verifichiamo

$$7 \pmod 5 = 2$$

$$A = k * 5 + 2$$

$$K=1, 1 * 5 + 2 = 7$$

$$K=2, 2 * 5 + 2 = 12$$

$$K=3, 3 * 5 + 2 = 17$$

Il modulo

$a \pmod n \equiv b$ al'è quel numero x tale che

$$X \bmod 7 = 1$$

$$x = k^* 7 + 1$$

Il fatto che aritmeticamente esistano più numeri che, modulo “qualcosa” danno lo stesso risultato, significa che se nell’hash c’è l’operazione modulo, allora più documenti possono collidere nello stesso hash

Più prosaicamente parlando, è un documento è lungo 1000000 di byte e lo “accorcio” a 32 byte, è inevitabile che con 32 byte io non possa rappresentare in modo biunivoco tutti i documenti lunghi 1000000 di byte

In termini numerici, 1000000 byte è il numero

Mentre 32 byte danno il numero

$$2^{(32 * 8)} = 2^{256} \quad 2^{1000000 * 8} = 2^{8000000}$$

hash

**L'hash soffre del birthday paradox
L'hash deve garantire la “sparsificazione” dei bit
Il numero generato dall'hash deve essere
“crittograficamente” forte**

I testi die hard nel NIST consentono di verificare la qualità di un algoritmo hash

**In termini generali una sequenza che deve garantire
caratteristiche di casualità (e quindi anche il risultato
di un hash) deve soddisfare la seguente
considerazione**

Leggendo i primi n bit dell'hash non devo essere in grado di calcolare se il bit n+1 sarà 1 o 0 con una probabilità diversa dal 50%

Tipi di hash

Md5
Sha1
Sha256
Sha384
Sha512
keccak

Usiamo openssl per calcolare hash

openssl dgst -hex -md5 mess1.txt

ea59dcf7967eb25aef21c4ee01f0064f

openssl dgst -hex -sha1 mess1.txt

221c688ec22e0f93e4c12073571c08ae30f37564

Oggi si usa almeno sha256

openssl dgst -hex -sha256 mess1.txt

d9fe0a25017909743fbe161449f43fd66982b4591eadea4939a9bd1ab2576c57

Sparsificare?

Ciao ciao ciao ciao ciao (1000000 di volte)

Sha256 = 8e3d785280006319eace5f73dc28622c4f5c1ec47365d162ff461097b1dd8210

**Modifico un bit del documento (da i a k nella prima
riga)**

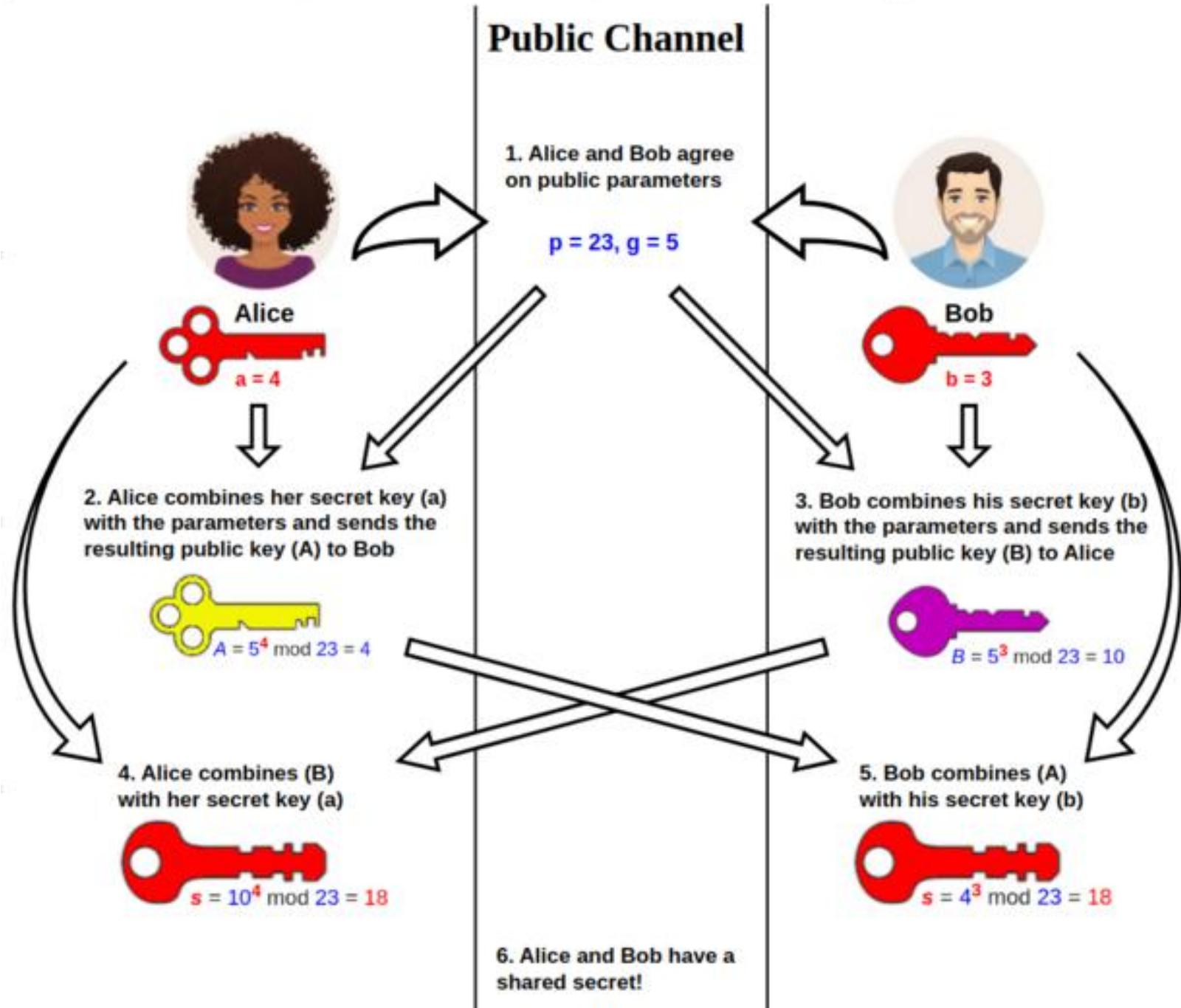
Sha256=166be6075142d63be348b08681babb3f943b19e64e92831e839816064384e6bb

Verifica dell'integrità

Cifra(K_{pub}^B, M) **meCifra($K_{pri}^A, Hash(M)$)** gatorio che il documento sia cfrato) e l'hash cfrato con la chiave privata del mittente

Chi lo riceve, se è cfrato, lo decifra con la sua chiave pubblica

Dopodiché decifra l'hash con la chiave pubblica di A, calcola lo stesso hash del documento ricevuto e questi due DEVONO essere uguali



Generazione di chiave come scambio dati

A e B condividono

p=23
g=5

a=11 (segreto di alice)

b=8 (segreto di bob)

Alice fa: $g^{**}a \text{ mod } p \Rightarrow 22$

Bob fa: $g^{**}b \text{ mod } p \Rightarrow 16$

Alice dà il risultato a Bob

Bob dà il risultato a Alice

Alice fa: $16^{**}a \text{ mod } p \Rightarrow 1$

Bob fa: $22^{**}b \text{ mod } p \Rightarrow 1$

Nota su python

Create un ambiente virtuale in modo da poter installare con pip senza problemi di conflitto con altre librerie

```
python -mvenv IlMioPython  
./IlMioPython/bin/activate
```

**Qui usate pip
Per terminare
deactivate**