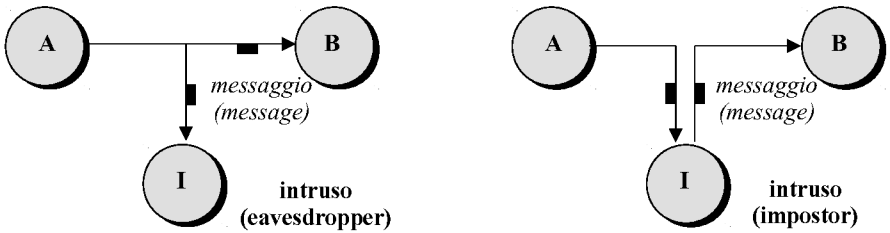
Introduzione alla crittografia

Un modello di comunicazione

anale di comunicazione sicuro

Un modello di comunicazione



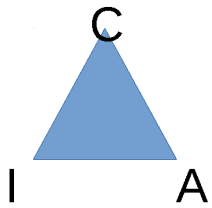
Attacco	Azione svolta dall'intruso	Minaccia sulla comunicazione
passivo	intercettazione	violazione della riservatezza
attivo	eliminazione, alterazione, falsificazione	violazione dell'integrità

I problemi della sicurezza

P1: la riservatezza (***Confidentiality***)- *E' possibile garantire al mittente ed al destinatario che un opponente, pur intercettando i messaggi in transito sul canale, non riesca ad entrare in possesso dell'informazione da essi trasportata?*

P2: l'integrità (***Integrity***)- *E' possibile garantire al destinatario che i bit del messaggio ricevuto non abbiano subito modifiche lungo il canale?*

P3: l'autenticità (***Authenticity***)- *E' possibile garantire, non solo al destinatario ma anche a terze parti, l'individuazione univoca dell'autore di un documento ricevuto tramite un messaggio e ad un mittente che nessuno gli potrà attribuire la paternità di un documento che non ha mai inviato?*



(P4: l'identità - E' possibile garantire a ciascuno dei due partecipanti ad una comunicazione che l'altro è proprio quello che dichiara di essere?)
(P5: la disponibilità: è possibile garantire l'accesso alle informazioni quando il legittimo utente ne ha necessità?)

Soluzione: crittografia

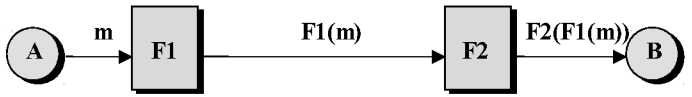
- Un'elaborazione matematica e algoritmica della codifica delle informazioni

finalizzata a

- Prevenire la violazione della riservatezza (una rilevazione a posteriori sarebbe inefficace!) → alterare il codice in modo da renderlo incomprensibile a chi non ha diritto di apprendere le informazioni
- Rilevare la violazione dell'integrità e autenticità (non può essere prevenuta!) → aggiungere al codice elementi che permettano la verifica delle informazioni ricevute

Operazioni primitive per la sicurezza

Comunicazione non interattiva (unidirezionale)



R1: prerequisiti per la riservatezza - *La trasformazione svolta in ricezione deve essere l'inversa di quella svolta in trasmissione ed il legittimo destinatario del messaggio deve essere l'unico a saperla fare.*

R2: prerequisiti per l'autenticazione - *La trasformazione in ricezione deve consentire al destinatario di verificare se il messaggio ricevuto è o meno il risultato di quella parti-colare trasformazione che il mittente deve essere l'unico a saper fare.*

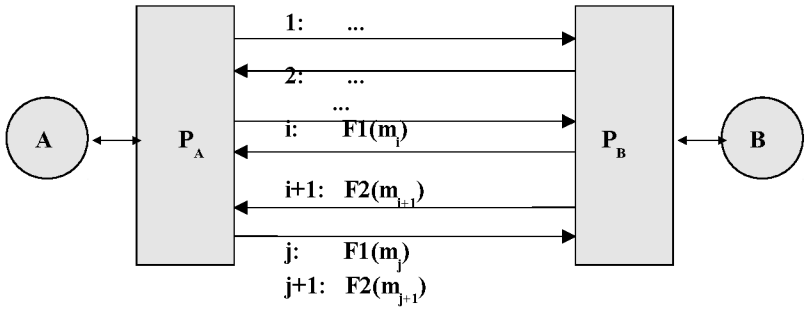
Operazioni primitive per la sicurezza

Caratteristiche (prestazioni) delle primitive:

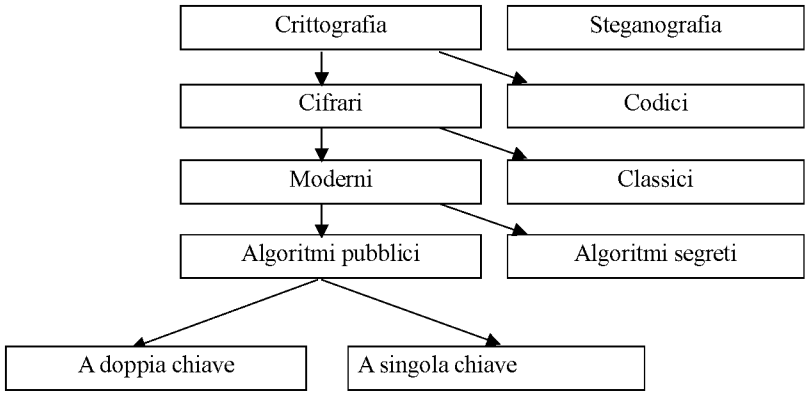
- **robustezza** (resistenza alla "rottura")
- **velocità** (n° di bit al secondo che i blocchi sono in grado di elaborare)
- **efficienza** (rapporto tra il n° di bit del messaggio a valle e a monte di
- **onerosità di gestione** (complessità delle azioni per mantenere "allineati" i due blocchi).

Protocollo di sicurezza

Comunicazione interattiva (bidirezionale) tra due o più partecipanti



Tecniche per la realizzazione delle primitive



Una lunga storia

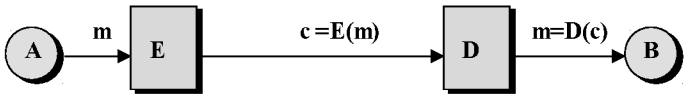
- VI sec. a.C. - Il cifrario Atbash degli Ebrei
  - Sostituzione monoalfabetica
- V sec. a.C. - La tavoletta di Demarato
  - Steganografia
- IV sec. a.C. - La scitala degli Spartani
  - Trasposizione
- IV sec. a.C. - Lo schiavo rapato di Istieo
  - Steganografia
- I sec. a.C. - Il cifrari di Cesare
  - Sostituzione monoalfabetica
- VIII sec. d.C. - Il trattato di Al-Kindi
  - Studio sistematico della crittoanalisi

Steganografia

- L'arte e la scienza del comunicare senza che altri se ne accorgano
- Esempi storici:
  - La tavoletta cerata di Demerato
  - Lo schiavo "rapato" di Istieo
  - Gli inchiostri invisibili
- Tecniche moderne
  - Modifica dei bit meno significativi di dati multimediali

Cifrari per la riservatezza

- Un *messaggio* è composto di simboli di un *alfabeto*
- Un cifrario opera sui simboli del messaggio
- Le operazioni primitive definite in precedenza prendono il nome di
  - *cifratura* (converte il *testo in chiaro* in *testo cifrato*)
  - *decifrazione* (converte il *testo cifrato* in *testo in chiaro*)



Crittoanalisi

- Un cifrario è detto sicuro se, dato un messaggio cifrato **c** e non conoscendo **D**, è impossibile trovare un testo **p** tale che **E(p) = c**.
- La sicurezza presuppone dunque due cose:
  - mantenere il segreto su almeno un particolare di **D**
  - usare una **E** non invertibile se non si conosce quel particolare.

Crittoanalisi

A seconda del materiale a disposizione del crittanalista si possono avere:

Tipo di attacco	Descrizione
FORZA BRUTA	Si tira ad indovinare D (o il suo particolare segreto) e si decifra il testo intercettato: se non ha alcun senso si ripete il procedimento
SOLO TESTO CIFRATO	Si eseguono analisi statistiche su una grande quantità di materiale cifrato e se ne usano le indicazioni per individuare quale p probabilmente corrisponde ad un dato c
TESTO IN CHIARO NOTO	Ci si procura in qualche modo sia dei testi cifrati, sia i corrispondenti testi in chiaro e si cerca di dedurre D analizzando le varie coppie
TESTO SCELTO	Si può scegliere testo da cifrare o da far decifrare per ottimizzare il procedimento di deduzione della chiave
RUBBER-HOSE	Si minaccia, ricatta o tortura qualcuno finché non cede la chiave.

Sicurezza assoluta e computazionale

sicurezza incondizionata - Un Cifrario è detto "assolutamente sicuro" se l'incertezza a priori sul testo in chiaro è uguale all'incertezza a posteriori, cioè dopo che si è analizzato in ogni possibile modo il corrispondente testo cifrato.

sicurezza computazionale - Un Cifrario è detto "computazionalmente sicuro" se il calcolare un **m** da un **c**, senza conoscere **D**, richiede una potenza di elaborazione superiore a quella che si può ipotizzare essere a disposizione dell'attaccante.

Algoritmi segreti

- Rappresentano una possibilità di rendere irrealizzabile **D**
- Benefici apparenti: difficoltà di studiare come invertire la cifratura
- Problemi:
  - mancanza di revisione della qualità
  - difficoltà di diffusione delle procedure
  - difficoltà di sostituzione delle procedure

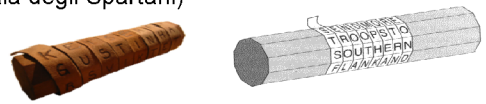
Algoritmi pubblici con chiave

- L'algoritmo è noto a tutti
- L'algoritmo impiega un parametro, detto *chiave*, noto solo ad A e B
- La forza dell'algoritmo risiede nell'oscurare il legame tra chiave e testo prodotto

- Vantaggi:
  - peer reviewing
  - maggiore facilità nello scambio di chiavi
  - minori problemi in caso di compromissione del segreto

Crittografia classica - trasposizione

Trasposizione (es. la scitala degli Spartani)



ALLE PROSSIME ELEZIONI MI PRESENTO ANCH'IO

A	P	I	L	N		E	A	I						
L	R	M	E	I	P	N	N	O						
L	O	E	Z		R	T	C							
E	S		I	M	E	O	H							
	S	E	O	I	S		'							

APILN EAILRMEIPNNOLOEZ RTC ES IMEOH SEOIS '

La chiave consiste nella dimensione della tabella e nell'ordine di lettura delle righe

Crittografia classica - trasposizione

Cosa succede applicando una trasposizione?

Si aggiunge *diffusione*

→

si "spargono" le proprietà statistiche di **m** in zone diverse di **c**

- Solitamente facile da attaccare con le statistiche dei di/trigrammi
- Per nulla banale se applicata ripetutamente

Crittografia classica - sostituzione

Sostituzione (es. il cifrario di Cesare)

Ogni lettera viene sostituita con quella che la segue di tre posizioni nell'alfabeto

Messaggio in chiaro: DE BELLO GALLICO
Messaggio cifrato: GH EHOOR LDOONFR

Più in generale: ogni lettera viene sostituita con un'altra dell'alfabeto

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

La chiave consiste nella sequenza alfabetica di destinazione (seconda riga)
In questo esempio lo spazio di chiavi ha dimensione $26! \approx 4*10^{26} \approx 2^{88}$

Crittografia classica - sostituzione

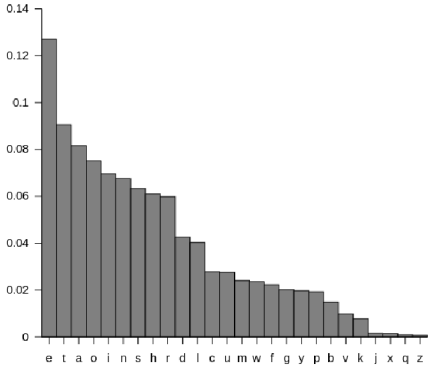
Cosa succede applicando una sostituzione?

Si aggiunge *confusione*

=

si oscura il legame tra **m** e **c**

- Nel linguaggio naturale, estremamente facile da attaccare con le statistiche di frequenza dei caratteri (in figura il grafico per la lingua inglese)
- Nel mondo binario, blocchi grandi hanno frequenze basse e uniformi (compressione)



Crittografia classica - sostituzione polialfabetica

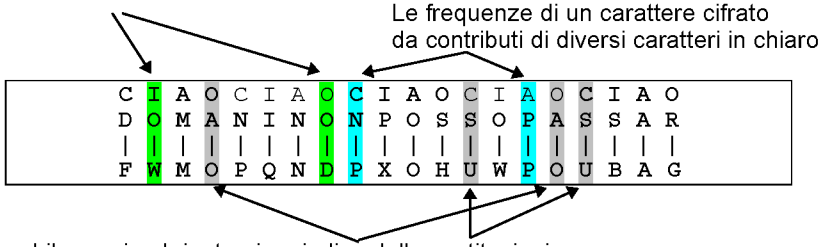
- Leon Battista Alberti (1466)
  Forma generale e implementazione meccanica
- Bellaso/Vigenère (1553)
  Forma semplificata
  - usata per 4 secoli
  - (es. la macchina Enigma - WWII)

Crittografia classica - sostituzione polialfabetica

Ogni carattere viene sostituito da un altro dipendente dalla chiave e dalla sua posizione (nell'esempio si consideri A=0, B=1, … Z=25 e si sommi modulo 26 la chiave al testo)

Le frequenze di un carattere in chiaro vengono sparse su più caratteri cifrati

Le frequenze di un carattere cifrato da contributi di diversi caratteri in chiaro



Attaccabile grazie al ripetersi periodico delle sostituzioni
Attaccabile facendo ipotesi sul contenuto del messaggio (*cribs*)

One-time pad

- Vernam/Mauborgne 1917
- E' l'unico cifrario a *sicurezza perfetta*
- Principi di funzionamento:
  - sostituzione polialfabetica
  - la chiave è *lunga come il messaggio*
  - la chiave è *perfettamente casuale*
  - la chiave non viene mai riutilizzata

Es: testo cifrato = WPE

Chiavi possibili: AAA … EVT … DYE… **RYE** … FHQ … (Tutte equiprobabili)

Testi in chiaro: WPE … SUL … TRA … **FRA** … RIO …

Ipotesi di decifrazione valide: ***tutte*** quelle della lingua considerata
→ Quella giusta è indistinguibile

Cifrari simmetrici moderni

- Applicano gli stessi principi di confusione e diffusione
- Operano sull'alfabeto binario invece che naturale
- Sono studiati per essere *computazionalmente sicuri*
- La sicurezza risiede nella lunghezza della chiave

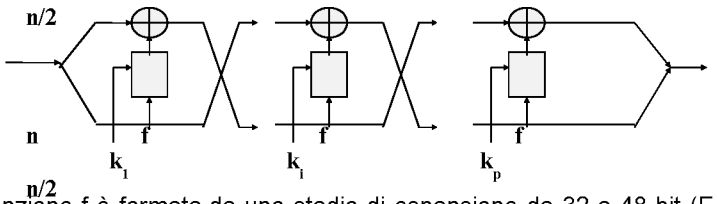
Standard storico: DES (National Bureau of Standards degli U.S.A in collaborazione con IBM, pubblicato nel 1977, chiave di 56 bit)

Standard attuale: AES/Rijndael (chiave variabile di oltre 64 bit)

http://csrc.nist.gov/encryption/aes/rijndael/

DES

DES è un *Cifrario a blocchi di Feistel*; opera su blocchi di 64 bit.



La funzione f è formata da uno stadio di espansione da 32 a 48 bit (E-box), seguito da uno stadio di EX-OR, da uno stadio di sostituzione formato da 8 S-boxes (che comprimono di nuovo il dato a 32 bit) e da uno stadio di permutazione (P-box).

Modi di operazione – ECB

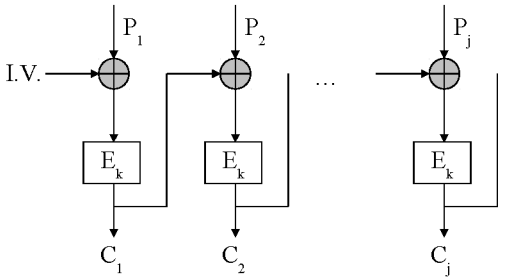
ECB – Electronic codebook, il modo più ovvio → usare un algoritmo per codificare efficientemente 2^n sostituzioni

Problemi:

- Non nasconde le ripetizioni del testo in chiaro
- Non ostacola la manipolazione del testo
- Lo stesso testo in chiaro e la stessa chiave producono sempre lo stesso testo cifrato → replay attacks

Modi di operazione – CBC

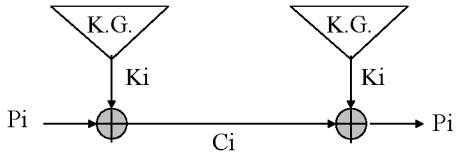
CBC – Cipher Block Chaining, aggiunge *feedback* per collegare tra loro i blocchi di un messaggio:



- Il cambiamento di un bit in C_i provoca un'analoga modifica di P_{i+1} e distrugge P_i
- L'aggiunta di blocchi in coda ad un messaggio non è rilevabile → usare MAC!

Stream ciphers

- Producono un flusso di bit (*keystream*) che vengono sommati ai bit del testo in chiaro
- La sicurezza risiede nella casualità dei bit del flusso
- Mai usare lo stesso keystream per cifrare più messaggi
- Problema: mantenere sincronizzati i keystream di cifratura e decifrazione



Altri modi di operazione

CFB – Cipher Feedback Mode

OFB – Output Feedback Mode

- Sono basati sull'utilizzo di retroazione di parti del testo mediante registri a scorrimento
- Possono offrire lievi vantaggi in termini di sicurezza e/o prestazioni in casi specifici

I problemi "difficili"

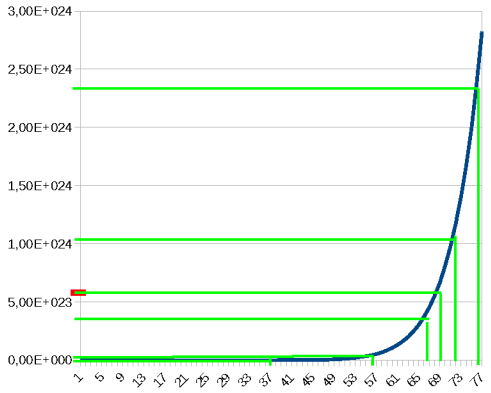
- Sono la base dei sistemi di cifratura asimmetrici
- Si *suppone* che siano computazionalmente non affrontabili

Esempi:

- diverse operazioni in *aritmetica modulare*

Utilità: costruire funzioni che siano facili da calcolare in un verso ma non nell'altro (unidirezionali) o lo diventino conoscendo un segreto (pseudo-unidirezionali)

Un'illustrazione intuitiva della difficoltà di invertire l'esponenziale modulare



y=x^13

Su R, se non conosco l'inversa di una funzione “regolare”, mi avvicino per approssimazioni successive (es. bisezione)

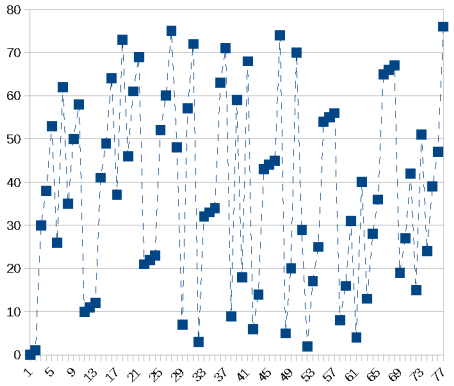
Per una funzione monotona, si parte dagli estremi del dominio, e si valuta la funzione nel punto medio del dominio.
In questo esempio, valutiamo la funzione per {0,76} → {38,76} → {57,76} → {66,5,76} → {66,5,71,25}

Per x=68.875 otteniamo il risultato

Un'illustrazione intuitiva della difficoltà di invertire l'esponenziale modulare

y=x^13 mod 77

Su Z_{77}, (il campo di Galois con 77 numeri, in cui le operazioni si effettuano modulo 77) l'effetto di riduzione modulare rende estremamente irregolare la funzione



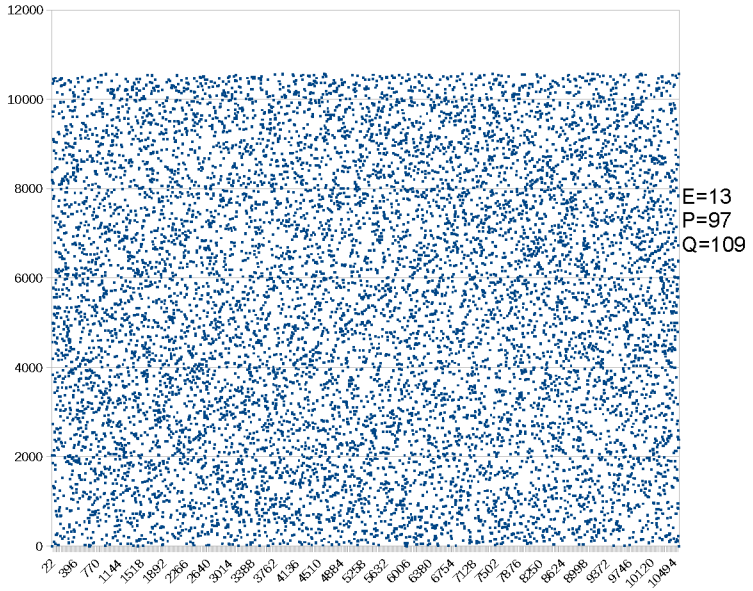
RSA

Generazione delle chiavi:
1. si scelgono due numeri primi *p* e *q*
2. il *modulo* viene calcolato come $n = pq$
3. si sceglie a caso un numero *d* e si calcola un numero *e* tale che $ed \bmod (p-1)(q-1) = 1$
4. si "gettano via" *p* e *q*
5. la chiave pubblica è (*e*, *n*), la chiave privata (*d*, *n*)

Cifratura: $\mathbf{c = m^e} \bmod n$
Decifrazione $\mathbf{m = c^d} \bmod n$

- Basato sulla difficoltà di fattorizzare grandi numeri

L'effetto di RSA



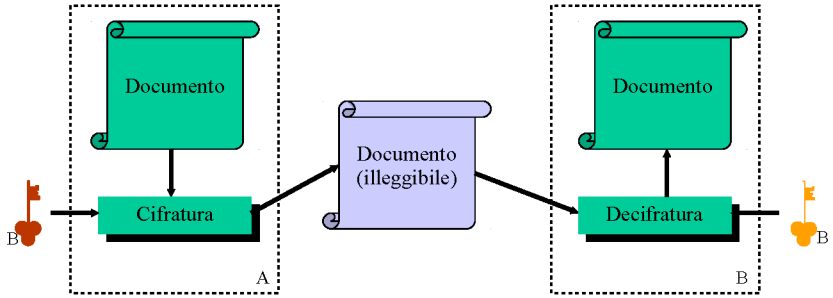
La crittografia asimmetrica

Nella crittografia asimmetrica le chiavi usate per cifrare e decifrare <u>sono diverse</u>

- Si basa sul concetto di funzione unidirezionale:
  - Si deve poter scegliere a caso una chiave di decifrazione
  - Si deve poter determinare con facilità la chiave di cifratura corrispondente
  - Non si deve poter determinare la prima dalla seconda

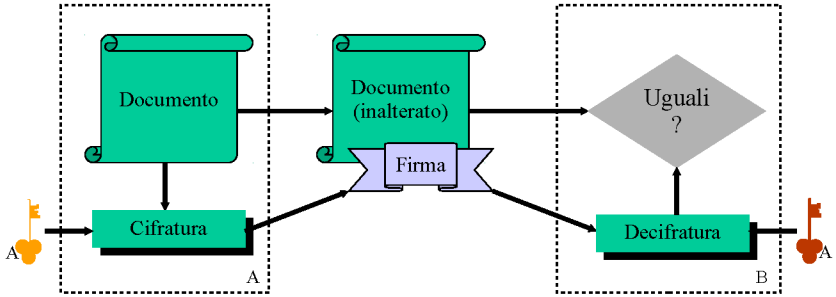
Si può rendere *pubblica* la chiave di cifratura
→ chiunque può cifrare un messaggio per **B**
Si tiene *privata* la chiave di decifrazione
→ solo **B** potrà rimettere in chiaro il messaggio

Cifratura a chiave pubblica

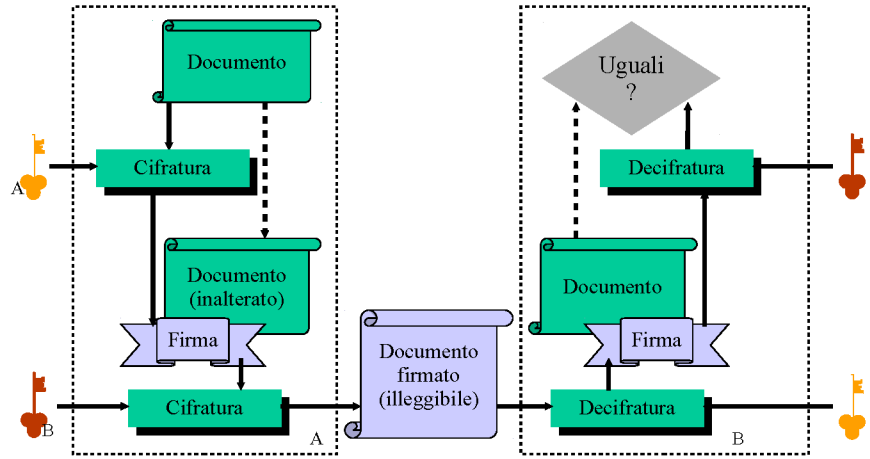


Problemi: attacchi con testo in chiaro noto → randomizzazione

Cifrari asimmetrici reversibili - firma

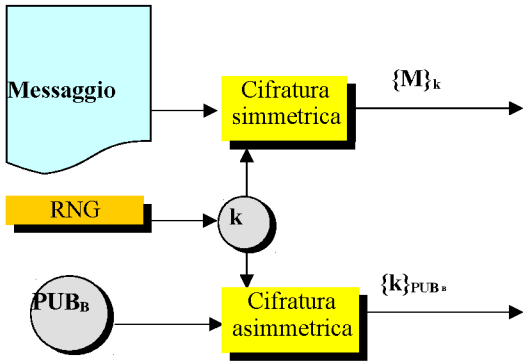


Firma e cifratura



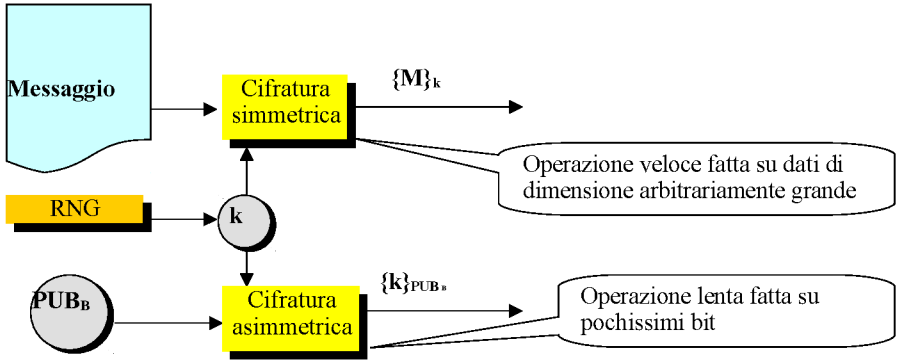
Cifrari ibridi

- I cifrari simmetrici sono molti ordini di grandezza più veloci di quelli asimmetrici.
- Per cifrare grandi quantità di dati è conveniente l'approccio *ibrido*:



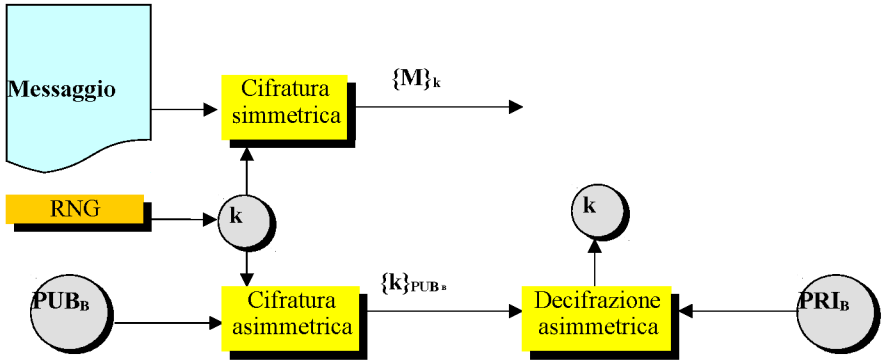
Cifrari ibridi

- I cifrari simmetrici sono molti ordini di grandezza più veloci di quelli asimmetrici.
- Per cifrare grandi quantità di dati è conveniente l'approccio *ibrido*:



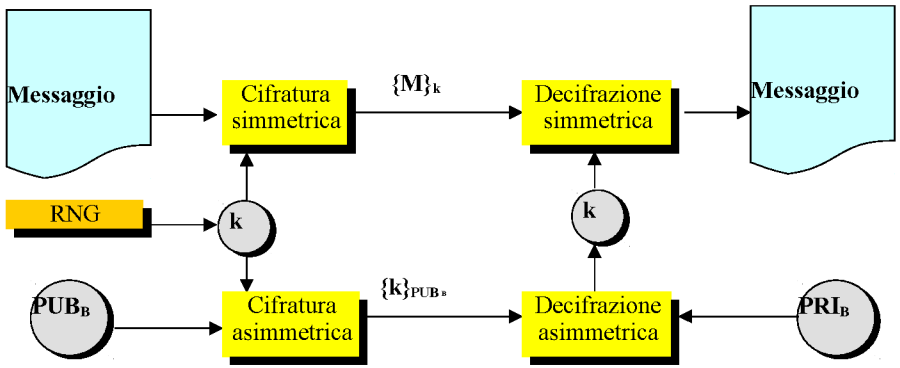
Cifrari ibridi

- Il destinatario è l'unico a poter decifrare *k*, disponendo della chiave privata corrispondente a quella pubblica utilizzata dal mittente



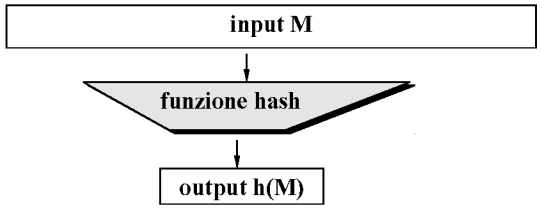
Cifrari ibridi

- L'utilizzo di chiavi simmetriche ogni volta differenti riduce al minimo la quantità di materiale cifrato con la stessa chiave ed ostacola ulteriormente la crittanalisi



Funzioni hash

- Usate per calcolare il "riassunto" o la "impronta" (*fingerprint*) di un messaggio
- Trasformazione *computazionalmente facile*
- Comprime ogni suo *input* **M,** di lunghezza arbitraria, in un *output* **h(M)** di lunghezza fissa e di piccole dimensioni



Funzioni hash

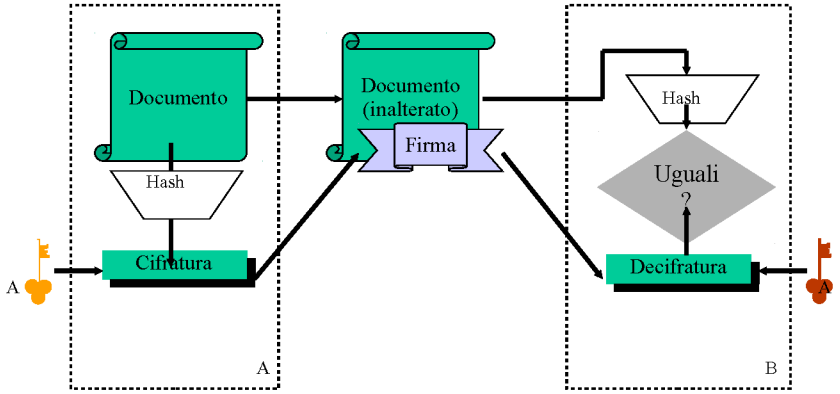
- Deve essere *difficile* trovare un **M** che fornisca un dato **h(M)** .
- Deve essere *difficile* trovare due messaggi $\mathbf{M_1}$, $\mathbf{M_2}$ tali che $\mathbf{h(M_1) = h(M_2)}$.

→ birthday attack

Due varianti:

- Hash con chiave → per verificare l'integrità di un messaggio
- Hash senza chiave → per migliorare le prestazioni di un sistema di firma digitale

Funzioni hash per l'efficienza della firma



Certificazione della chiave pubblica

La verifica di una firma garantisce solo la correttezza della coppia di chiavi usate

↓

Serve un meccanismo per collegare chiave pubblica e *identità del titolare*

Due possibili soluzioni (*modelli di fiducia*):

- Web of trust
- Infrastruttura di certificazione

Modelli di fiducia - web of trust

- L'autenticità di una chiave pubblica è *testimoniata* da altri utenti
- L'utente che riceve una chiave da uno sconosciuto può decidere di accettarla per autentica se è firmata da qualcuno fidato
- Il sistema gratuito PGP adotta questo modello di fiducia

Vantaggio: nessuna entità "super partes" di cui doversi fidare

Svantaggio: pessima scalabilità

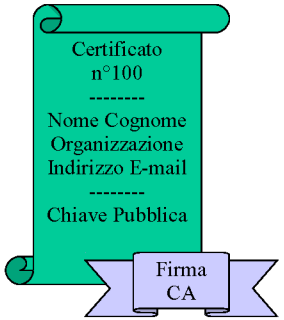
Modelli di fiducia - infrastruttura di certificazione

Al crescere della comunità di utenti si rende necessaria una architettura più
- Garanzia di affidabilità uniforme delle chiavi
- Garanzia di riuscire a reperire informazioni sulla chiave di un utente sconosciuto

Per questo sono state standardizzate le *infrastrutture per la certificazione della chiave pubblica* (public-key infrastructure - PKI). Tali infrastrutture si occupano di gestire i *certificati* che associano la chiave pubblica all'identità del titolare della coppia.

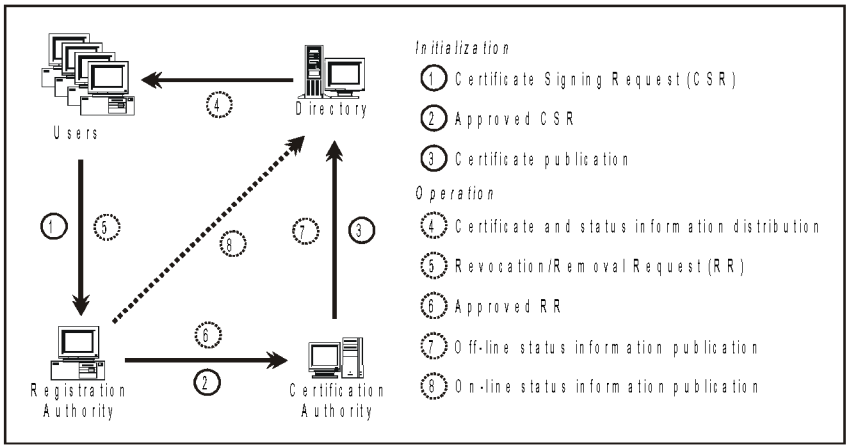
Componenti:
- Registration Authority (RA)
- Certification Authority (CA)
- Directory

PKI - certificati X.509v3



Composti da:
- Parti fisse
  - Dati identificativi
  - Chiave pubblica
  - Numero di serie
  - Date di emissione e scadenza
  - Firma della Certification Authority
- Estensioni

PKI – ciclo di vita dei certificati



Generazione delle chiavi crittografiche

Può coinvolgere
- *entrambi* i partecipanti
- *uno solo dei due*
- *una terza parte* di cui entrambi si fidano.

E' opportuno:
- che la scelta della chiave coinvolga un “generatore di numeri casuali”
  - per chiavi simmetriche qualsiasi valore è ammissibile
  - per chiavi asimmetriche è necessario verificare l'adeguatezza ai requisiti matematici
- che le chiavi asimmetriche vengano generate dal titolare della chiave privata

Generazione di numeri casuali

Se questa operazione non viene svolta correttamente, le relazioni statistiche residue possono ridurre notevolmente lo spazio delle chiavi da cercare (es. Netscape 1.1: generatore di chiavi a 128 bit con una *entropia* effettiva di 20)

- In un sistema di calcolo la vera casualità è impossibile
- Ottimi generatori sono i sistemi naturali (decadimento radioattivo, rumore termico, …) ***ma*** attenzione alle elaborazioni successive
- Caratteristiche necessarie per una sequenza *pseudo-random:*
  - Sembra casuale (agli occhi dei test sulle distribuzioni di 0 e 1)
  - E' impredicibile
  - E' non riproducibile

Problema: testare efficacemente queste caratteristiche

Comunicazione delle chiavi simmetriche

- Chiavi simmetriche
  - Necessità di segretezza
  - Necessità di evitare che troppo materiale sia cifrato con la stessa chiave

Soluzioni possibili:

- Scambio sporadico e diretto di una master key, poi
- Scambio frequente di chiavi cifrate con la master key

Comunicazione delle chiavi simmetriche

Ulteriore problema dei sistemi simmetrici: serve una chiave per ogni coppia di utenti

Il numero di chiavi cresce come il quadrato degli utenti

Soluzione: Key Distribution Center

- ogni utente condivide una chiave sola con KDC
- si usa questa chiave per comunicare una chiave di sessione tra due utenti

Problema: replay attacks, colli di bottiglia, sicurezza e fiducia del KDC

Scambio di chiavi simmetriche

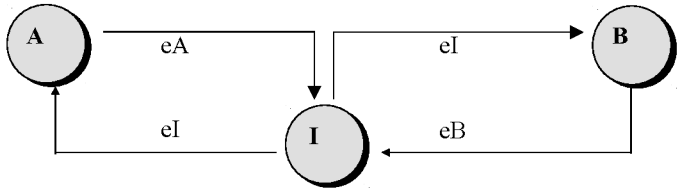
La crittografia asimmetrica può aiutare quella simmetrica:

- una chiave simmetrica può essere inviata cifrandola con una chiave pubblica
- una chiave simmetrica può essere scambiata direttamente con Diffie-Hellman:

1. **A** genera a caso X, **B** genera a caso Y
2. **A** e **B** concordano pubblicamente un numero grande q e un numero g < q
3. **A** manda a **B**: $x = g^X \bmod q$, **B** manda a **A**: $y = g^Y \bmod q$
4. **A** calcola $k = y^X \bmod q = g^{YX} \bmod q$
5. **B** calcola $k = x^Y \bmod q = g^{YX} \bmod q$

Scambio di chiavi asimmetriche

L'invio di una chiave pubblica e lo scambio di DH non hanno problemi di riservatezza, però sono esposti all'attacco *dell'uomo in mezzo* (man in the middle) che ne può compromettere l'utilizzo.



→ il problema è l'autenticità

Memorizzazione delle chiavi

- Originariamente:
  - Lo scopo della crittografia era *ricordare* la chiave invece del messaggio
- Con la crittografia moderna:
  - Le chiavi sono lunghe sequenze di numeri pseudo-casuali

Soluzioni possibili:

- Memorizzazione sul sistema + controllo dell'accesso
- Memorizzazione sul sistema + cifratura con chiave mnemonica
  - Problema: dictionary attack
  - Vantaggio: difficile distinguere una chiave ben decifrata da una no
- Memorizzazione in dispositivi tamper-proof

Lunghezza delle chiavi

Il miglior attacco contro gli algoritmi simmetrici moderni è la forza bruta.
Esempi di tempi di ricerca con le tecnologie attuali:

	Lunghezza della chiave in bit		
Costo	56	80	128
1 K€ (individuo)	38 anni	640 milioni di anni	10^{21} anni
1 M€ (impresa)	19 giorni	100.000 anni	10^{18} anni
1 G€ (NSA)	12 secondi	6 anni	10^{14} anni

- Attenzione alle ricerche con tempo di calcolo gratis (lotteria cinese, virus) e alla sfortuna!
- C'è un limite invalicabile: la termodinamica
  - Limite di Landauer: per cambiare 1 bit almeno $k \times T \times \ln(2)$ (3×10^{23} J a 3°K)
  - Tutta l'energia emessa dal Sole in un anno = 1.2×10^{34} J
    → 4×10^{56} bit flip, come contare da 0 a 2^{188}
  - Energia emessa dall'esplosione di una supernova = 2×10^{44} J
    → 7×10^{66} bit flip, come contare da 0 a 2^{222}

Lunghezza delle chiavi

Il miglior attacco contro RSA non è la ricerca a forza bruta, ma la fattorizzazione del modulo con GNFS.

La complessità di GNFS è sub-esponenziale, ed è stimata dalla formula a lato, in cui n è il numero da fattorizzare, la cui lunghezza è quindi circa $\log_2 n$

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\ln n)^{\frac{1}{3}}(\ln \ln n)^{\frac{2}{3}}\right)$$

Applicando questa formula si può stimare che

Una chiave RSA con modulo di … bit	È tanto sicura quanto una chiave simmetrica di … bit
1024	80
2048	112
3072	128
7680	192
15360	256

Note:

- un sistema ibrido verrà attaccato sul punto più debole
- meglio scegliere la chiave asimmetrica più robusta (di solito deve durare di più)

Recupero delle chiavi

L'utilizzo della cifratura pone due ordini di problemi legati al recupero dei messaggi:

- Recupero in caso di smarrimento della chiave di decifrazione
- Recupero da parte delle forze dell'ordine

→ key escrow

→ key sharing

Le chiavi di firma invece non hanno bisogno di essere recuperate:

- se viene smarrita una chiave si produce una nuova coppia
- le vecchie firme continuano a essere verificabili con la vecchia chiave pubblica

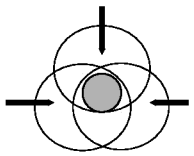
Osservazione: anche se possibile tecnicamente, è evidentemente sconsigliabile usare la stessa chiave privata asimmetrica per la firma e per la decifrazione

Autenticazione

Come provare l'identità

La conformità di una caratteristica fisiologica o comportamentale con un dato "biometrico" di riferimento.

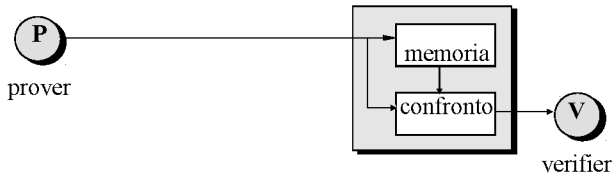
La conoscenza di un dato segreto concordato in precedenza (*password* o *personal*



Il possesso di un oggetto riconoscibile da parte della macchina (una scheda a *banda magnetica* o una *smart card*)

Prova della conoscenza di segreti

Autenticazione *passiva*: comunicazione sempre uguale ed esplicita del segreto
Autenticazione *attiva*: comunicazione sempre diversa che non svela il segreto (*zero knowledge proof*)



Autenticazione passiva

Problemi:

- Tentativi di indovinare il segreto → criteri di "educazione" alla scelta delle password
- intercettazione e replay → nessuna soluzione a parte la cifratura del canale
- furto del segreto sul server → rendere inutile il furto

Le password di Unix

- Il sistema non deve conoscere le password
- Il sistema deve discriminare una password corretta da una errata

Non si memorizza la password ma la sua impronta

Problema: dictionary attack, stessa password su macchine diverse
Soluzione: salt

Autenticazione attiva - SKEY

- Un utente conosce il proprio segreto X
- Il sistema conosce il risultato della ripetuta applicazione di una funzione hash a X
  $X \rightarrow F(X) \rightarrow F^2(X) \rightarrow F^3(X) \rightarrow \ldots \rightarrow F^{n-1}(X) \rightarrow F^n(X)$
- Alla prima autenticazione l'utente invia **$F^{n-1}(X)$**
- Il sistema verifica facilmente che **$F(F^{n-1}(X)) = F^n(X)$**
- Il sistema scarta **$F^n(X)$** e ricorda **$F^{n-1}(X)$**

L'hash è

- facile da calcolare → efficiente
- difficile da invertire → sicuro

Il sistema va però reinizializzato dopo *n* passi

Autenticazione attiva - challenge/response

- Il sistema genera un *nonce* e lo invia all'utente
- L'utente cifra con la chiave simmetrica/firma con la chiave privata il nonce e lo rinvia
- Il sistema decifra la risposta/verifica la firma confrontando il risultato col nonce originale

Pericoli:

- Cifratura asimmetrica:
  - attacchi condotti facendo cifrare all'utente ignaro valori costruiti ad arte → decifrazione di messaggi
  - attacco dell'uomo in mezzo
- Cifratura simmetrica:
  - attacco dell'uomo in mezzo

SSH – Secure SHell

Origini e motivazione

- L'amministrazione di sistemi remoti tipicamente utilizzava TELNET
  - Nessuna confidenzialità del canale
  - Nessuna autenticazione dell'host
  - Autenticazione passiva dell'utente

SSH nasce come alternativa sicura a TELNET per affrontare principalmente le debolezze sopra indicate, e viene reso ancora più utile dalla possibilità di eseguire tramite il canale sicuro non solo una sessione di lavoro interattiva ma qualunque tipo di comando.

Sicurezza

- Il collegamento SSH tra client (*ssh*) e server (*sshd*) avviene attraverso questi passi essenziali
  - Negoziazione dei cifrari disponibili
  - Autenticazione dell'host remoto per mezzo della sua chiave pubblica
  - Inizializzazione di un canale di comunicazione cifrato
  - Negoziazione dei metodi disponibili per l'autenticazione dell'utente
  - Autenticazione dell'utente

Ognuno dei passi elencati può essere portato a termine in modo configurabile, al fine di garantire il compromesso tra sicurezza e flessibilità più adatto al contesto.

Autenticazione dell'host

- L'autenticazione dell'host remoto è importante per evitare di cadere nella trappola tesa da un eventuale uomo nel mezzo, che potrebbe così catturare la password dell'amministratore spacciandosi per l'host su cui egli vuole effettuare il
  - Non è previsto un sistema centralizzato di attestazione dell'autenticità della chiave dell'host
  - Alla prima connessione l'amministratore deve utilizzare un metodo out-of-band per determinare la correttezza della chiave pubblica presentata dall'host
  - Alle connessioni successive la chiave pubblica memorizzata dal client dell'amministratore permette di effettuare un'autenticazione attiva

Le chiavi pubbliche vengono memorizzate nel file *known_hosts* nella directory *.ssh* posta nella home dell'utente sul client.

Autenticazione dell'utente

- Ci sono due possibilità per l'autenticazione dell'utente sull'host remoto
  - Autenticazione passiva, tradizionale, con username e password – i dati sono trasmessi all'host autenticato su di un canale cifrato, quindi con buon livello di sicurezza
  - Autenticazione attiva, per mezzo di un protocollo challenge-response a chiave pubblica – presuppone che l'utente si doti della coppia di chiavi, e che installi correttamente sull'host remoto la chiave pubblica

Autenticazione dell'utente

In entrambi i casi, l'identità dell'utente con cui viene tentato il login sull'host remoto può essere selezionata: in assenza di indicazioni specifiche verrà usato lo stesso nome utente con cui l'operatore sta lavorando sul client

Es:

utente "marco" sul client esegue "ssh remoteserver"

→ si presenta come utente "marco" su "remoteserver" e si deve autenticare di conseguenza

utente "marco" sul client esegue "ssh root@remoteserver"

→ si presenta come utente "root" su "remoteserver" e si deve autenticare di conseguenza

Generazione e gestione delle chiavi

- Per poter effettuare l'autenticazione attiva un utente deve generare una coppia di chiavi asimmetriche ed installare sull'host remoto la chiave pubblica. Lo standard considerato ora più sicuro e supportato dalla versione 2 del protocollo (le versioni precedenti sono del tutto sconsigliate) è DSA. I passi sono:
  - Sul client: *ssh-keygen –t dsa –b 2048*
  - Scegliendo tutti i valori di default la chiave privata sarà memorizzata in *.ssh/id_dsa* e la chiave pubblica in *.ssh/id_dsa.pub* sotto la home dell'utente
  - Il contenuto di *.ssh/id_dsa.pub* deve essere copiato sull'host remoto, nel file *.ssh/authorized_keys2* sotto la home dell'utente a nome del quale si vuole entrare sull'host medesimo

Esempio di setup per l'autenticazione attiva

L'utente *marco* che lavora sulla postazione *deis118* vuole amministrare come *root* l'host remoto *liaserver*. Su *deis118* lancia la generazione delle chiavi e poi:

scp .ssh/id_dsa.pub root@liaserver:.ssh/authorized_keys2

oppure

cat .ssh/id_dsa.pub | ssh root@liaserver "cat >> .ssh/authorized_keys2"

(chiarimento in seguito)

Ovviamente a questo punto, non essendo ancora le chiavi installate, l'operazione richiede la password di root di liaserver. Da questo momento in poi però non sarà più necessaria:

password-less login - straordinariamente comodo per automatizzare task di monitoraggio e controllo remoto

Avvertenze

Il ruolo autenticante della password viene sostituito dalla presenza della chiave privata dell'utente sul client – la segretezza della password è quindi sostituita dalla riservatezza del file che contiene la chiave privata

- Grande cura nell'impostazione dei permessi di file e directory (nota di tipo pratico: spesso il passwordless login non funziona semplicemente perché i permessi sulla directory *.ssh* dell'host remoto sono troppo larghi, e quindi il server sshd "non si fida" dell'integrità del suo contenuto)
- Possibilità di proteggere la chiave privata con una password
  - Priva della possibilità di passwordless login
  - Più sicuro comunque che utilizzare direttamente la password dell'account remoto, e più pratico se si amministrano molti host remoti

Esecuzione remota

Lanciando *ssh utente@host* si ottiene un *terminale remoto* interattivo.

Aggiungendo un ulteriore parametro, viene interpretato come comando da eseguire sull'host remoto al posto della shell interattiva; gli stream di I/O di tale comando vengono riportati attraverso il canale cifrato sul client.

Es: *ssh root@server "grep pattern"*

I dati forniti attraverso STDIN al processo ssh sul client vengono resi disponibili sullo STDIN del processo grep sul server

STDOUT e STDERR prodotti dal processo grep sul server "fuoriescono" dagli analoghi stream dal processo ssh sul client