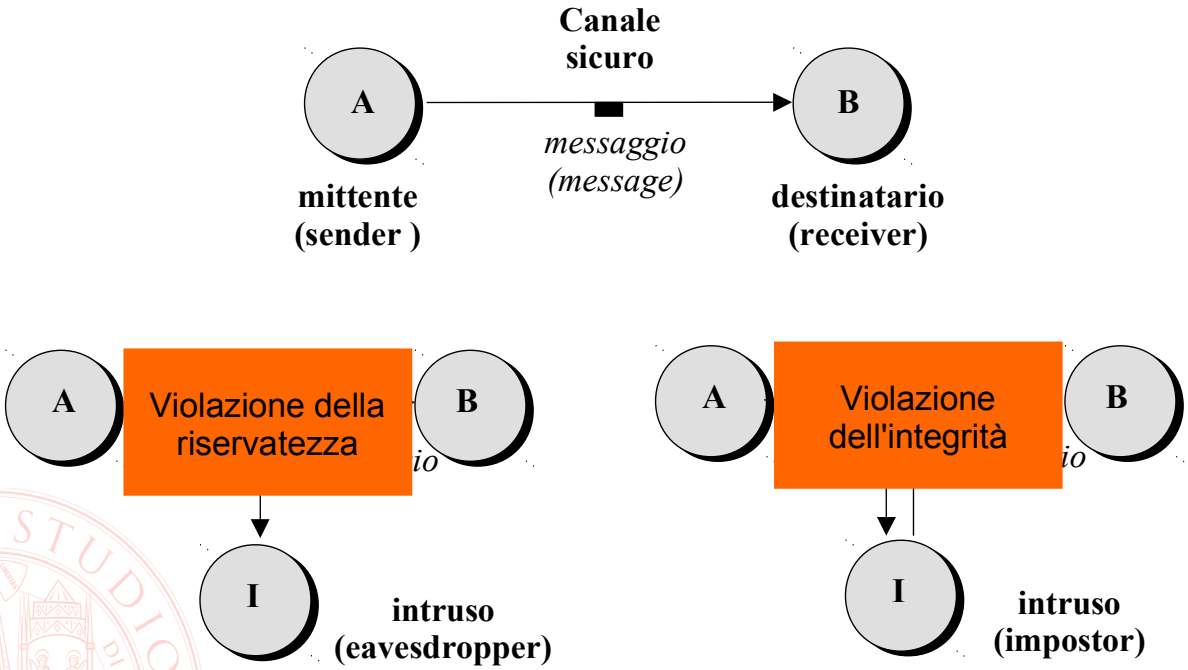
Cenni di crittografia

Marco Prandini
DISI – Università di Bologna

Da cosa è fatta la sicurezza delle informazioni

- Confidentiality (riservatezza)
- Integrity (integrità)
  - Authenticity (paternità)
- Availability (disponibilità)

Mondi ideali e reali



Soluzione: crittografia

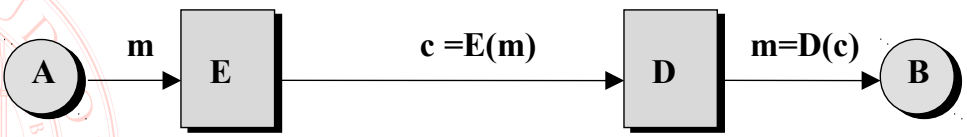
- Un'elaborazione matematica e algoritmica della codifica delle informazioni
- Prevenire la violazione della riservatezza (una rilevazione a posteriori sarebbe inefficace!): →alterare il codice in modo da renderlo incomprensibile a chi non ha diritto di apprendere le informazioni
- Rilevare la violazione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

Una novità introdotta da Internet?

- 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

Cifrari per la riservatezza

- Due operazioni
  - Cifratura
    converte il testo in chiaro in testo cifrato
  - Decifrazione
    converte il testo cifrato in testo in chiaro



I principi di Kerckhoffs (1883)

1) Le système doit être matériellement, sinon mathématiquement, indéchiffrable ;
   - Sicurezza *computazionale* o *assoluta*
2) Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;
   - ~~segreto=algoritmo~~ segreto=chiave!
3) La clef doit pouvoir en être communiquée et etenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants;
   - Cifratura = ricordare un segreto semplice per poter scambiare molti segreti arbitrari

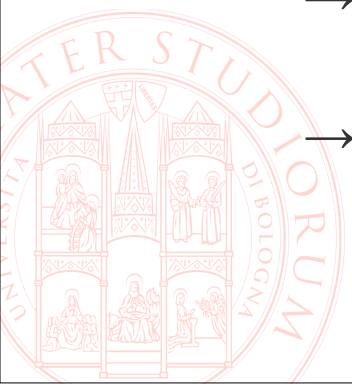
Crittoanalisi

- Di fronte a un testo cifrato con algoritmo noto, cosa può sempre fare un crittoanalista?
  - Analizzare le proprietà statistiche del testo

    → robustezza = capacità dell'algoritmo di occultare le proprietà del testo in chiaro
  - Cercare la chiave tra tutte quelle possibili

    → sicurezza assoluta = rendere totalmente indistinguibile la chiave giusta dalle altre

    → sicurezza computazionale = rendere il processo di ricerca della chiave troppo oneroso



Sostituzione monoalfabetica

- Cifrario di Cesare, Agony Columns del Times, parole crociate crittografate della Settimana Enigmistica, …

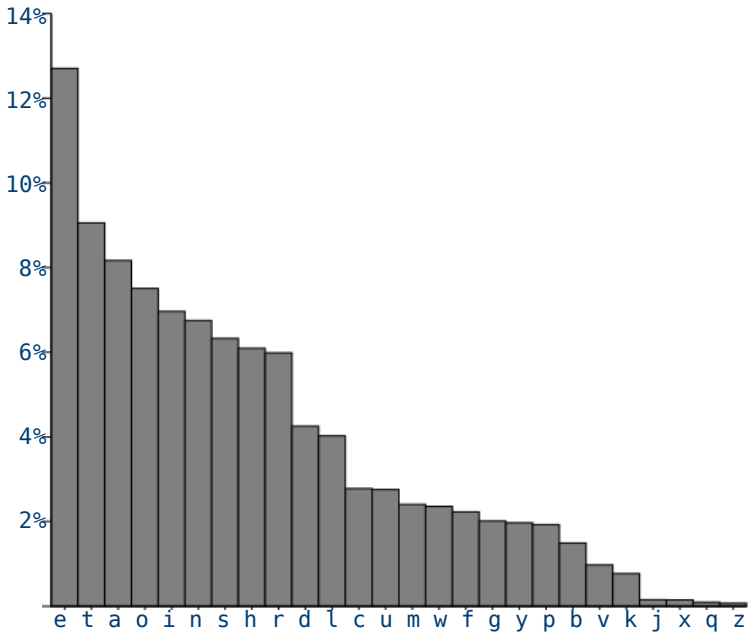
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Q	W	E	R	T	Y	U	I	O	P	L	K	J	H	G	F	D	S	A	Z	X	C	V	B	N	M

- **CRITTOGRAFIA →ESOZZGUSQYOQ**
- Ricerca della chiave: spazio di 26! ≈ $4*10^{26}$ ≈ 2^{88}
- Robustezza…

Attacco alla sostituzione

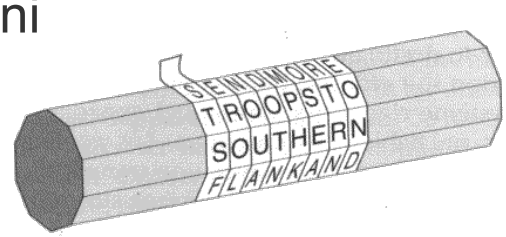
Nel linguaggio naturale, estremamente facile con le statistiche di frequenza dei caratteri (in figura il grafico per la lingua inglese)

Nel mondo binario, la "lettera" può essere un lungo blocco di bit → frequenze basse e uniformi (compressione) → buona efficacia!



Trasposizione

- La scitala degli Spartani



- Algoritmicamente basta una tabella scritta per colonne e letta per righe

  ALLE PROSSIME ELEZIONI MI PRESENTO ANCHE IO

A	P	I	L	N		E	A	
L	R	M	E	I	P	N	N	I
L	O	E	Z		R	T	C	O
E	S		I	M	E	O	H	
	S	E	O	I	S		E	

  APILN EA LRMEIPNNI LOEZ RTCOES IMEOH SEOIS E

Trasposizione

- Ricerca della chiave:
  - dimensione della tabella
  - ordine di lettura delle righe
- Robustezza
  - Statistiche dei *digrammi* e *trigrammi*
    - Permettono di dedurre la dimensione della tabella
  - Per nulla banale se applicata ripetutamente

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)

Sostituzione polialfabetica

Es. 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 derivano da contributi di diversi caratteri in chiaro

Key flow:	C	I	A	O	C	I	A	O	C	I	A	O	C	I	A	O	C	I	A	O
Message:	D	O	M	A	N	I	N	O	N	P	O	S	S	O	P	A	S	S	A	R
	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|
	F	W	M	O	P	Q	N	D	P	X	O	H	U	W	P	O	U	B	A	G

Attaccabile grazie al ripetersi periodico delle sostituzioni

Attaccabile facendo ipotesi sul contenuto del messaggio (*cribs*)
- Trattato sulla crittoanalisi di Charles Babbage (1853)
- Decifrazione rapida di Enigma ad opera di Alan Turing (WWII)

One-time pad

- Vernam/Mauborgne (1917)
- Polialfabetica con chiave
  - Scelta perfettamente a caso
  - Lunga quanto il messaggio
  - Mai riutilizzata

Ma che fatica!

Testo in chiaro FRA, Testo cifrato: WPE

Tutte equiprobabili

Chiavi possibili AAA … EVT … DYE … **RYE** … FHQ …

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

Ipotesi valide: ***tutte*** quelle della lingua considerata

→ Quella giusta è indistinguibile

Sicurezza perfetta!

Cifrari simmetrici moderni

- Applicano gli stessi principi di confusione e diffusione
  - Reiterando sostituzioni e trasposizioni
- 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/

Robustezza dei cifrari simmetrici

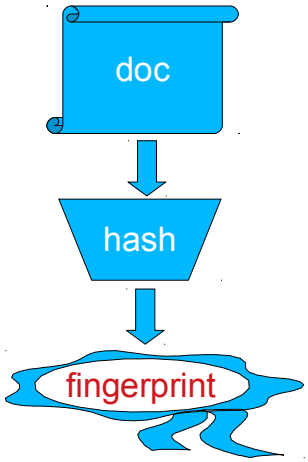
- Il miglior attacco è la forza bruta.
  Esempi di tempi di ricerca con tecnologie recenti:

	Lunghezza della chiave in bit		
Budget	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}

Funzioni hash

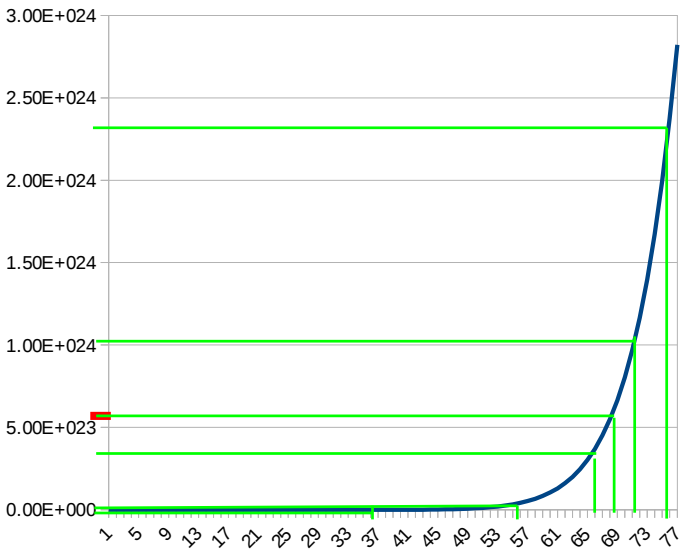
- Gli stessi principi possono essere usati senza chiave per ottenere "impronte digitali" compatte di documenti di dimensione arbitraria
- Fingerprint:
  - dimensione fissa (f. non biunivoca)
  - f. pubblica, senza chiave
  - difficili da falsificare
    - Non si riesce a trovare doc dato fingerprint
    - Non si riesce a trovare coppia di doc con lo stesso fingerprint
- Uso: autenticazione (memorizzazione password, firma digitale)



Problemi difficili e trabocchetti

- Operazioni facili in un verso e (speriamo) computazionalmente infattibili nell'altro
  - A meno di conoscere un segreto
- Fattorizzazione di grandi numeri
- Molte operazioni in aritmetica modulare
  - Numeri interi
  - Come risultato di un'operazione si prende il resto della divisione per un *modulo* fisso

Intuitivamente



y=x^13

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

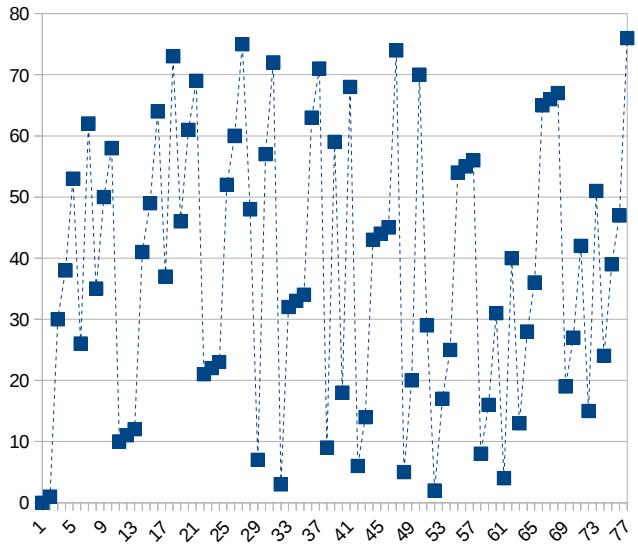
Per una funzione monotona, si parte dagli estremi del dominio, e si valuta la funzione nel punto medio del dominio.
In questo esempio, valutiamo la funzione per {0,76} → {38,76} → {57,76} → {66,5,76} → {66,5,71,25}

Per x=68.875 otteniamo il risultato

Intuitivamente

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 → non è possibile una ricerca efficiente

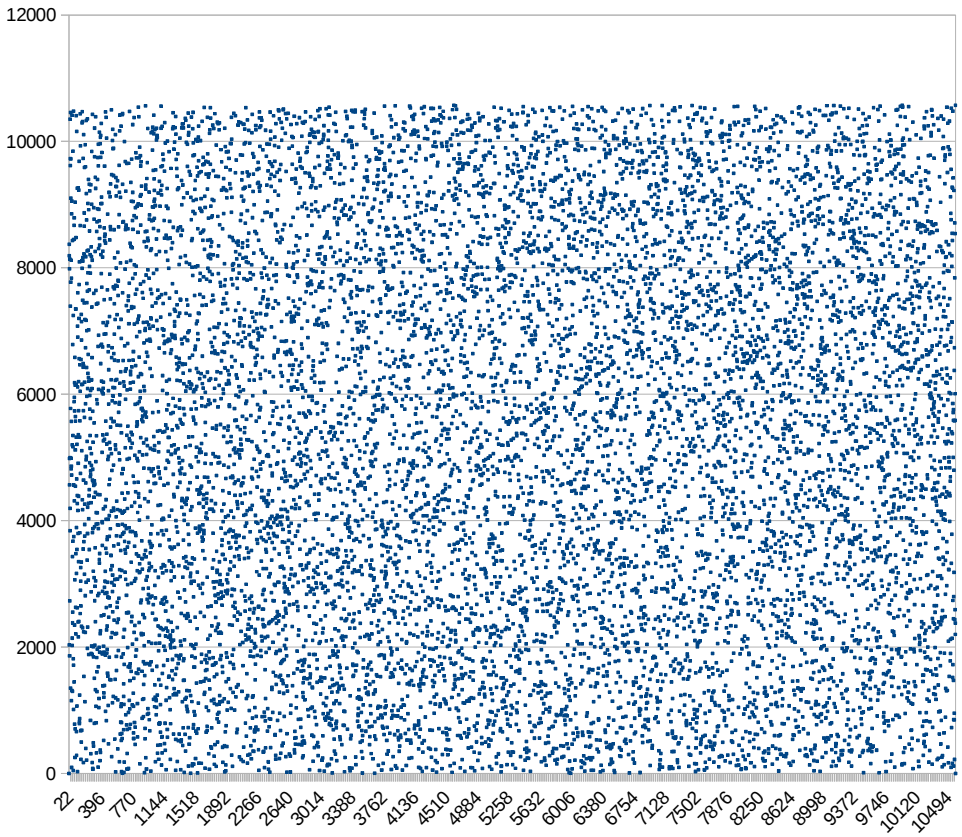


Crittografia asimmetrica: RSA (1977)

- Generazione delle chiavi:
  1. si scelgono due numeri primi p e q
  2. il modulo viene calcolato come n = p·q
  3. si sceglie a caso un numero d e si calcola un numero e tale che e·d mod (p-1)(q-1) = 1
     - Facile solo conoscendo p e q, che vengono poi dimenticati

- La chiave pubblica è (e, n), la chiave privata (d, n)
- Cifratura: $c = m^e \bmod n$
- Decifrazione $m = c^d \bmod n$

Visivamente

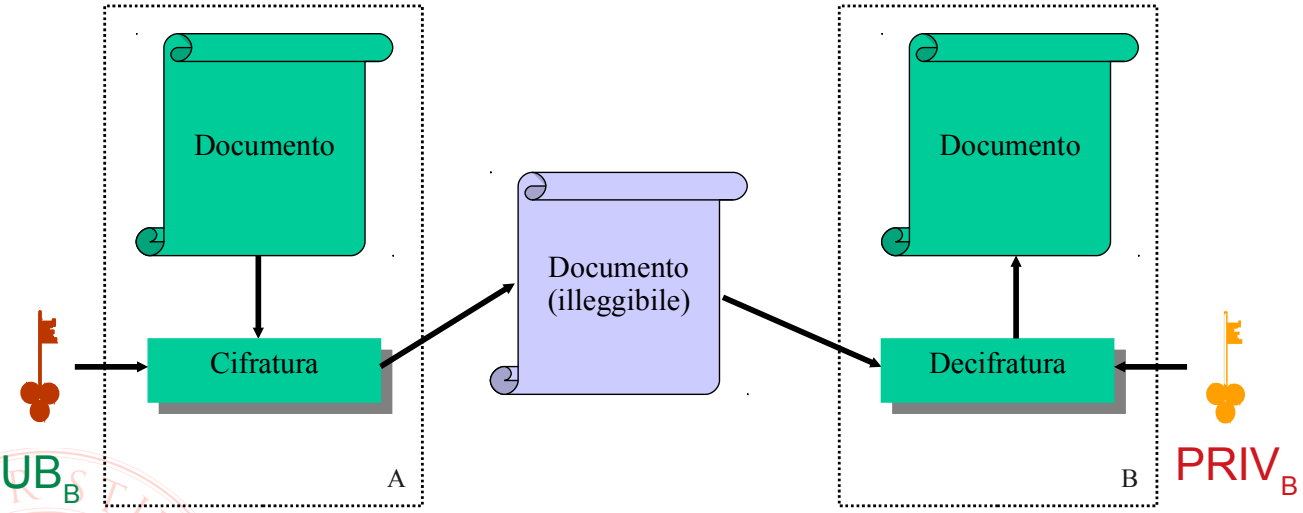


E=13
P=97
Q=109

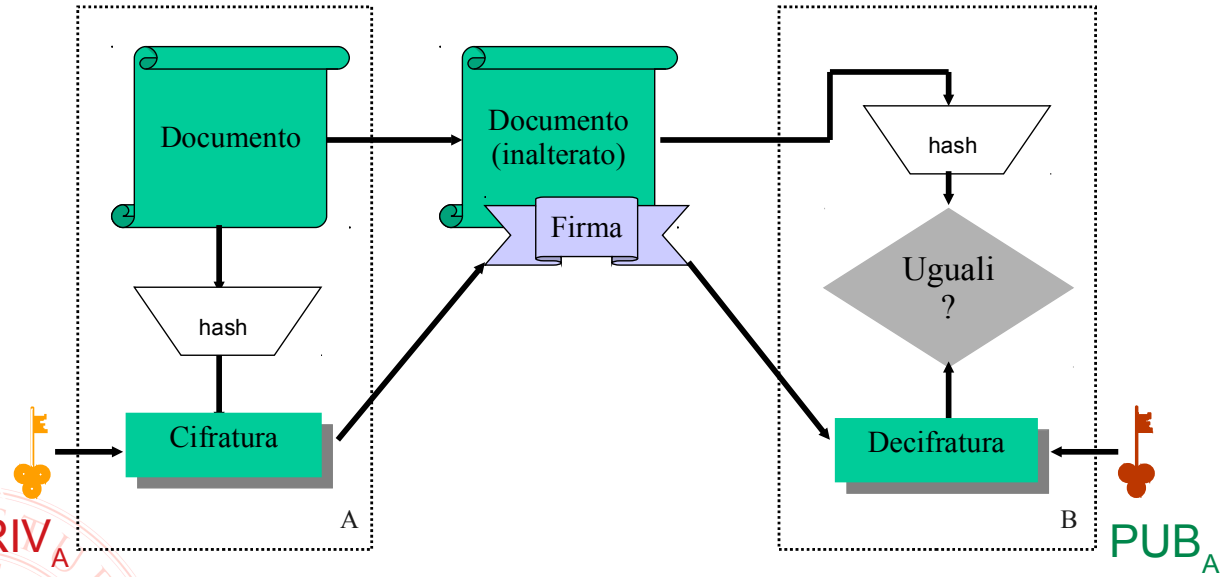
Vantaggi della c. asimmetrica

- Le chiavi usate per cifrare e decifrare sono diverse
- La chiave pubblica può essere distribuita
  - Da essa non è derivabile la chiave privata
  - Chiunque può usarla per cifrare
- La chiave privata corrispondente è l'unica che può decifrare
- La chiave privata è specifica di un solo utente quindi utile anche per *autenticare*

C. asimmetrica per la riservatezza



C. asimmetrica per l'integrità e l'autenticità



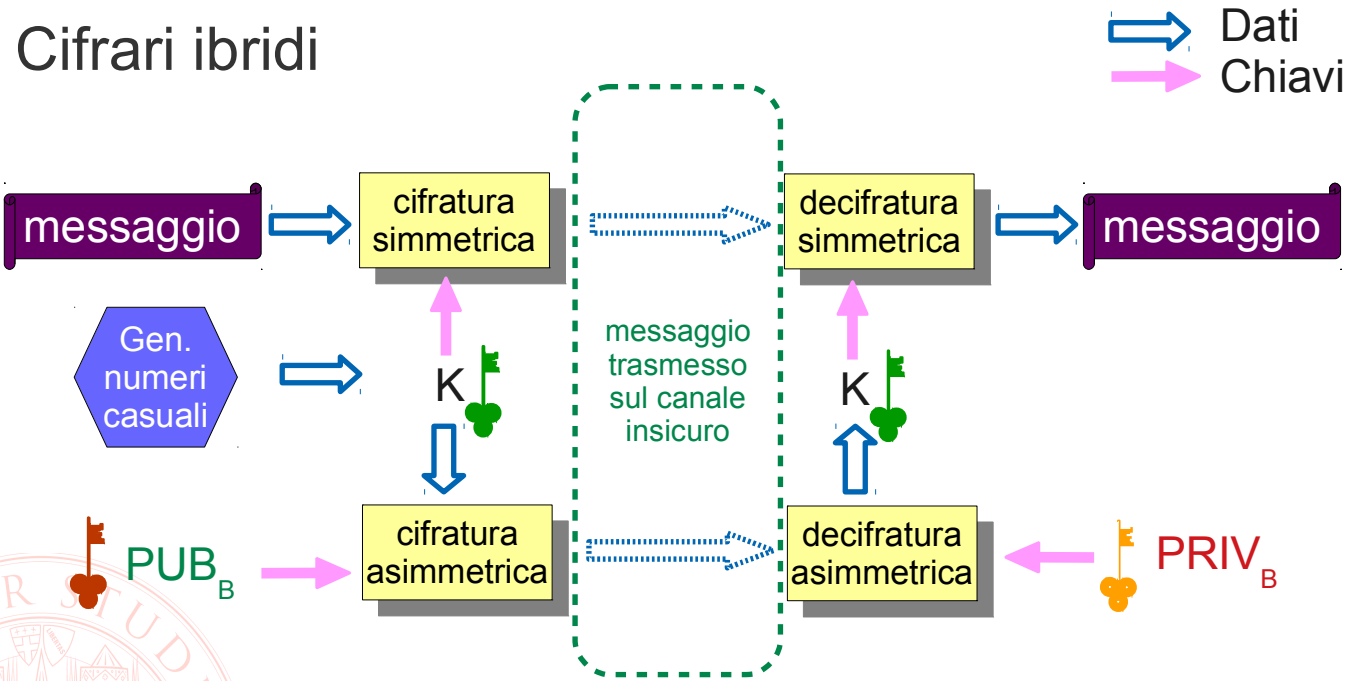
Il successo della verifica garantisce che si è usata la chiave pubblica corrispondente a quella privata usata per firmare… ma chi garantisce che sia davvero dell'utente A?

C. asimmetrica - pregi e difetti

- Grandi vantaggi:
  - distribuzione delle chiavi
  - utilità per tutte le proprietà di sicurezza
- Punti deboli:
  - Prestazioni (5-10 volte più lento di AES)
    - Sistemi ibridi
  - alcuni attacchi specifici (known plaintext)

Aggiungere prestazioni e flessibilità

Cifrari ibridi



Più destinatari = un solo messaggio cifrato &
più copie di K cifrate con la chiave pubblica di ognuno

Utilizzo della c. asimmetrica per l'autenticazione

- Autenticazione passiva (es. password)
  - svela il proprio segreto a chi verifica
  - sempre uguale → replay attack (una password intercettata può essere riutilizzata)
- Autenticazione attiva con cifrari asimmetrici (in principio)
  - il possesso della chiave privata identifica univocamente un utente o sistema, poiché non viene mai condivisa
  - il *prover* (P) dimostra al *verifier* (V) di possedere una certa chiave privata rispondendo a una *sfida*:
    - V genera un numero random grande R
    - V cifra R con la chiave pubblica di P
    - V manda a P la sfida
    - P se è autentico riesce a decifrare e risponde con R
  - Non svela il segreto!
  - R non prevedibile, non riutilizzato, intercettarlo e rigiocarlo è inutile
  - **Come nella firma, chi garantisce a V di detenere effettivamente la chiave pubblica di P e non quella di un impostore?**

Robustezza della c. asimmetrica

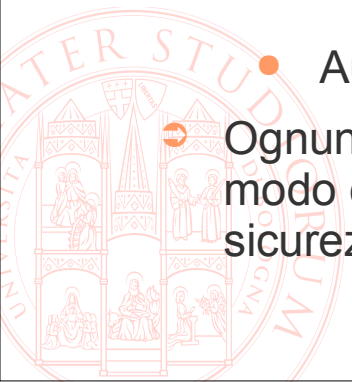
- Le chiavi non sono numeri casuali, esiste una relazione matematica che facilita l'attacco
- Il metodo più efficiente è tentare la fattorizzazione del modulo (SNFS)
- Per questo la lunghezza consigliata del modulo è di 2048 bit e oltre
- Esiste inoltre un problema di autenticità della chiave

Secure Shell

- Necessità: amministrazione remota
- Predecessori: TELNET
  - Nessuna confidenzialità del canale
  - Nessuna autenticazione dell'host
  - Autenticazione passiva dell'utente

Secure Shell

- 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.



Secure Shell – host authentication

- 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 login
  - 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.

Secure Shell – user authentication

- 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

Secure Shell – user authentication

- 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

Secure Shell – key generation

- Per poter effettuare l'autenticazione attiva un utente deve
  - generare una coppia di chiavi asimmetriche
    - comando `ssh-keygen -t rsa -b 2048`
  - installare sull'host remoto la chiave pubblica.
    - file locale `.ssh/id_rsa.pub`
    - copia su host remoto

      `scp .ssh/id_rsa.pub user@remote:`
    - append alla lista di utenti autorizzati (su *remote*)

      `cat id_rsa.pub >> .ssh/authorized_keys`

Secure Shell – 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
  - Vi 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

Secure Shell – 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