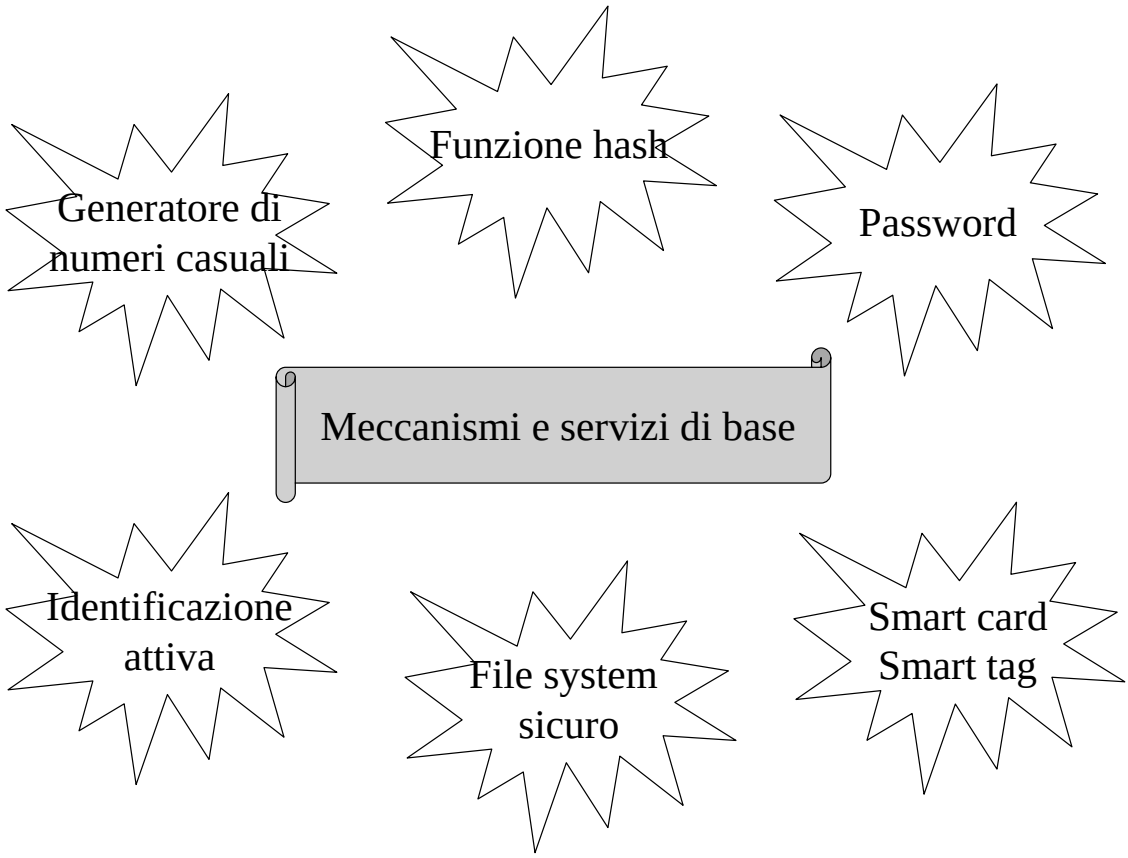
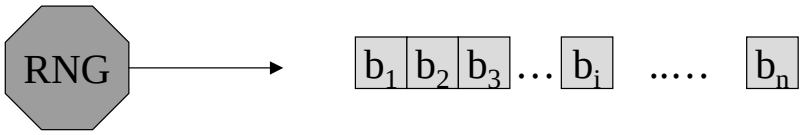


Random Number/Bit Generator



1- Proprietà

•**<u>Casualità</u>**: indipendenza statistica ed equiprobabilità dei bit

<u>Test statistici (FIPS 140-1): $\chi 2$</u>
- Monobit: *numero* di 1 e di 0
- Pocker: *sequenze* di M bit
- Run: *block* (1) e *gap* (0)
- Long Run: *block* più lungo

<u>Altri test statistici</u>:
- Autocor.: *differenze* dopo shift
- TdF: *distanza* pattern ripetitivi
- Compressione LZ: *lunghezza*
- …

•**<u>Imprevedibilità</u>** del nuovo valore dai precedenti**(CSPRBG)**

<u>Next-bit test</u>: dati L bit della stringa non deve esistere un algoritmo polinomiale in grado di predire il bit L+1-esimo con $p > 0,5$

Generatori Hw e Sw

•Fenomeni fisici:

Decadimento radioattivo, rumore termico, turbolenza di un fluido

•Segnali di apparati elettronici:

Microfono, telecamera, oscillatore

•Programmi di estrazione di rumore dal funzionamento del computer

Tastiera, Mouse, n° di processi attivi, traffico di rete

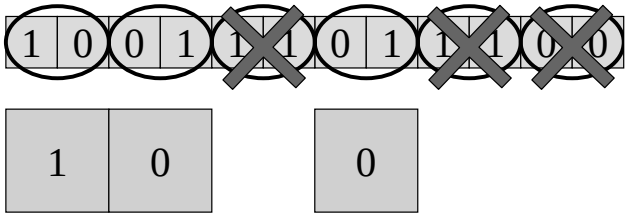
Bassa bit rate

memorizzazione

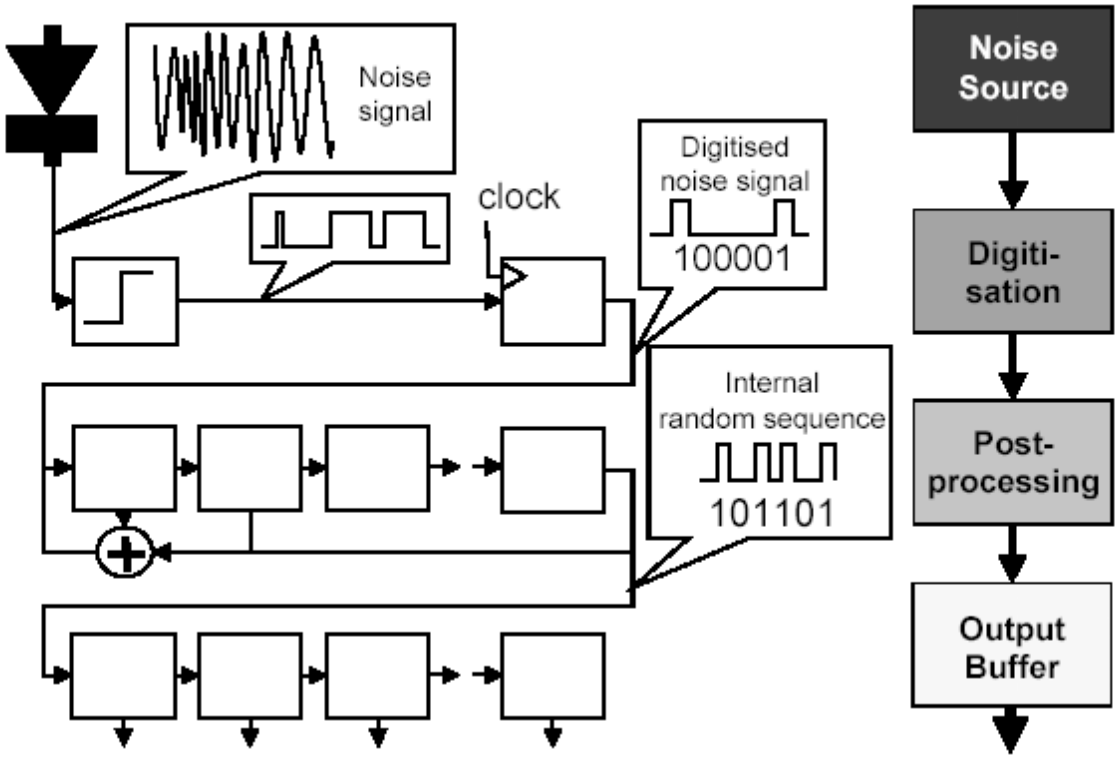
inizializzazione

Polarizzazione

deskewing

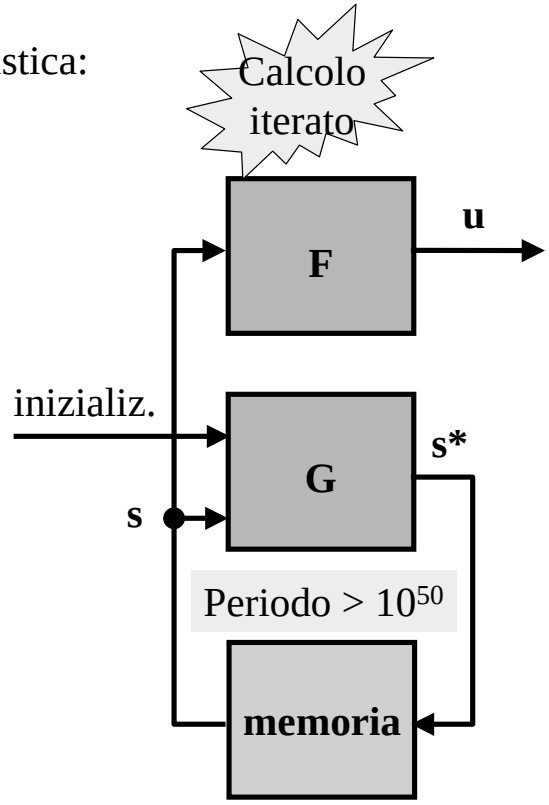


Elaborazione del rumore



Pseudo Random Number Generator

Simulazione & Elaborazione statistica:
algoritmi deterministici



Congruenza lineare
$X_{i+1} = (a.X_i + b) \bmod m$

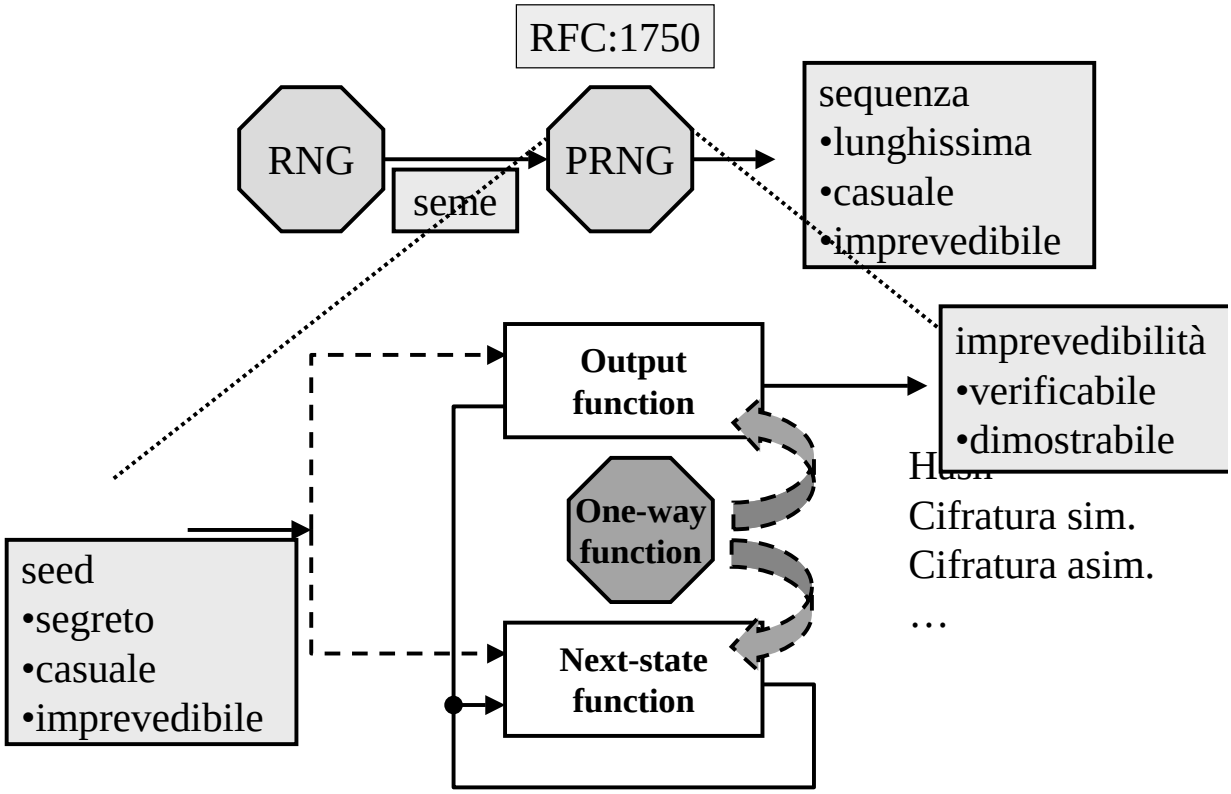
Casualità: SI
Imprevedibilità: NO

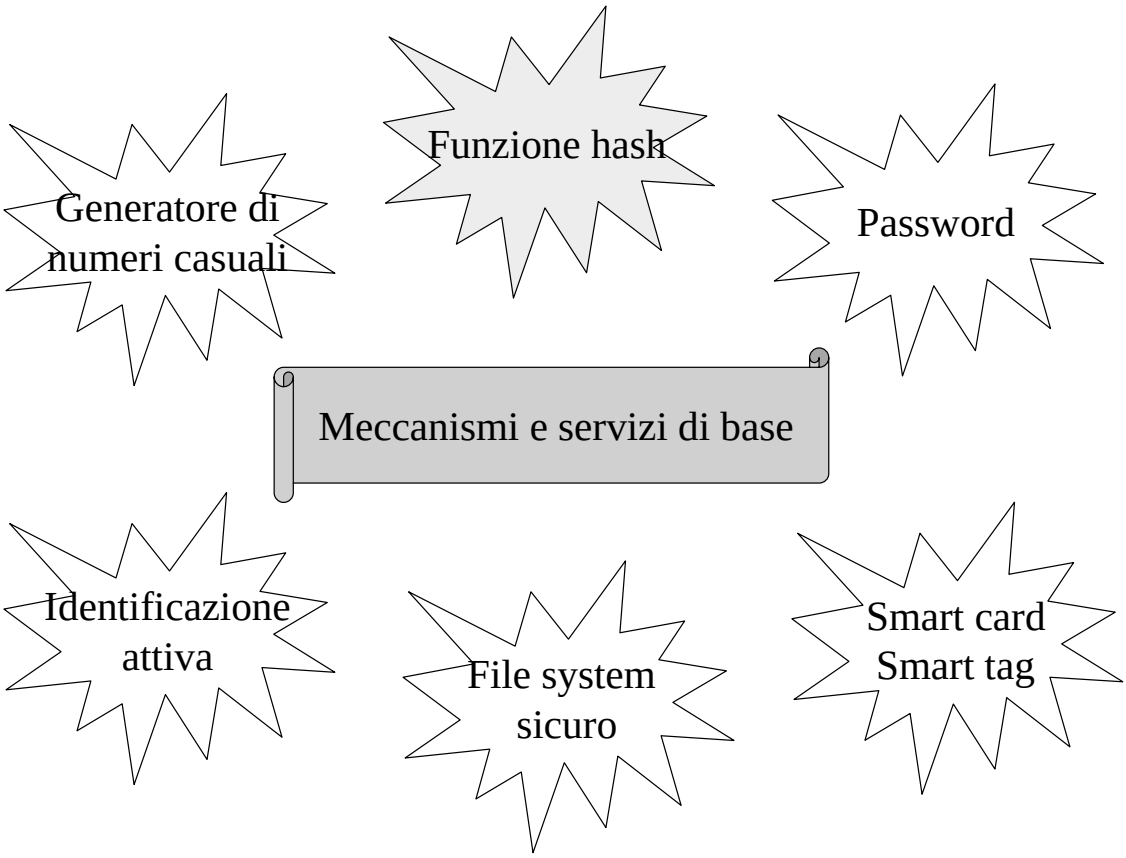
$m = 2^{31}-1$, $a = 7^5$, $b = 0$

$X_1 = (a.X_0 + b) \bmod m$
$X_2 = (a.X_1 + b) \bmod m$
$X_3 = (a.X_2 + b) \bmod m$

PRNG crittografico

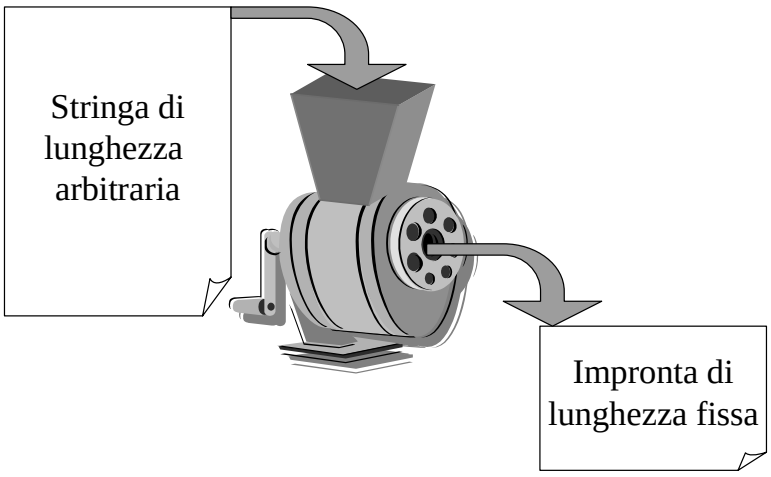
RFC:1750

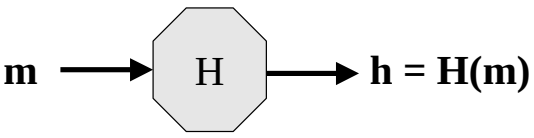




La funzione hash

riassunto (*digest*) o impronta (*finger-print*) di un'informazione

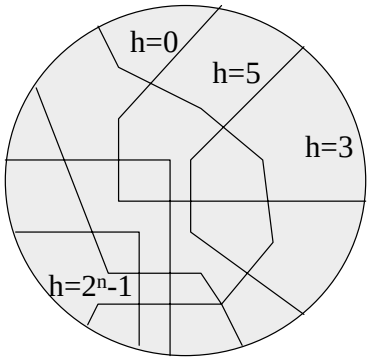




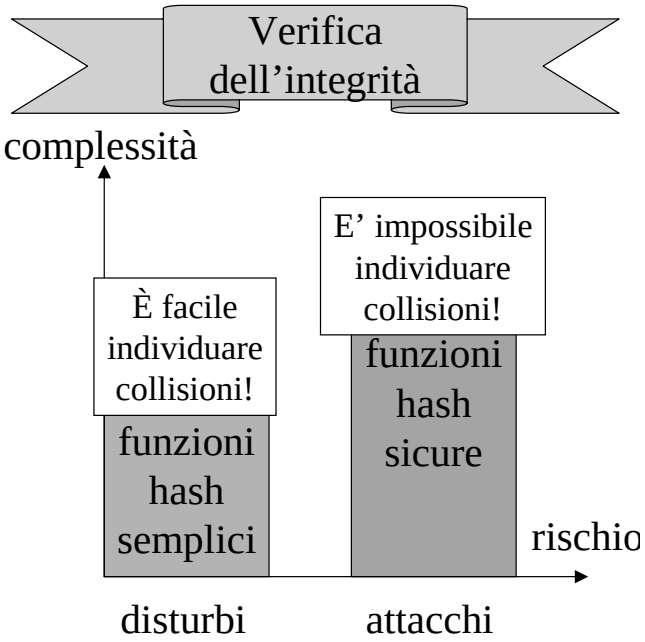
Collisioni

n bit di hash: 2^n valori d'uscita

Input space: 2^m con **m > n**



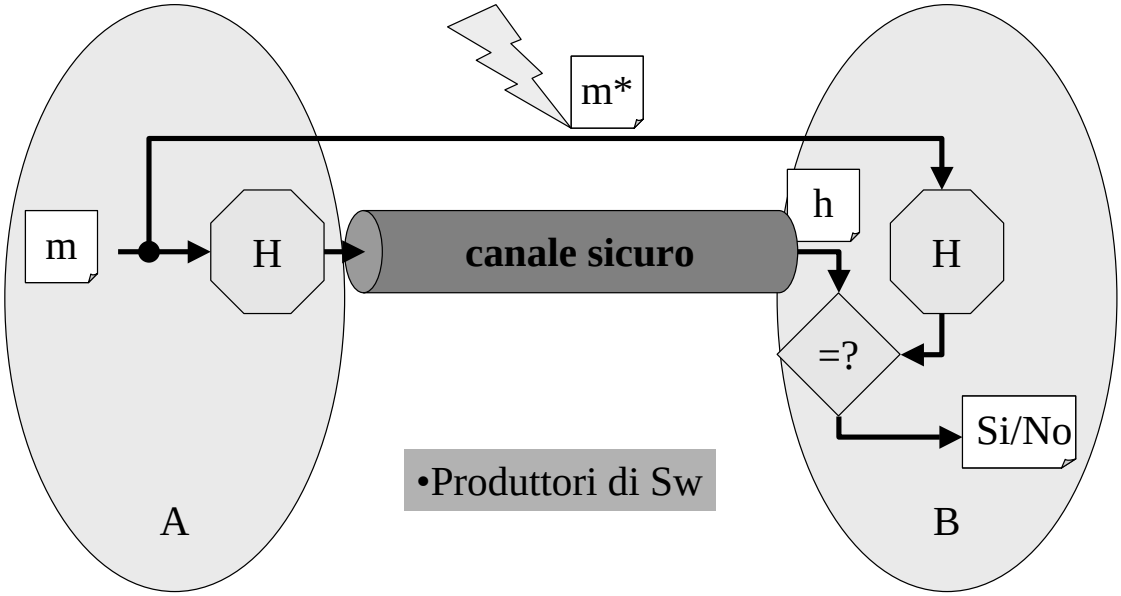
Le stringhe d'ingresso
che forniscono uguale impronta
sono dette essere in collisione



Proprietà di una funzione hash sicura

- R18: "*il calcolo di H(x) deve essere facile per ogni x*"
- R19: "*per ogni x deve essere computazionalmente infattibile trovare un y ≠ x tale che H(y) = H(x)*"
- R20: "*deve essere computazionalmente infattibile trovare una qualsiasi coppia y, x tale che H(y) = H(x)*"
- R21: "*per ogni h deve essere computazionalmente infattibile trovare un x tale che H(x) = h*"

R19: resistenza debole alle collisioni



•Produttori di Sw

L'intruso non deve poter forgiare un m* in collisione con m

Complessità del calcolo di una collisione

IPOTESI: una funzione hash sottoposta ad ingressi scelti a caso restituisce, con eguale probabilità, uno dei suoi 2^n valori d'uscita.

Problema: individuare un ingresso che fornisca un'impronta assegnata

un tentativo: probabilità di successo $\mathbf{P_1(2^n, 1) = 2^{-n}}$,
probabilità di insuccesso $\mathbf{1 - 2^{-n}}$.
k tentativi: probabilità di successo $\mathbf{P_1(2^n, k) = 1 – (1 - 2^{-n})^k}$

Teorema binomiale: $\mathbf{(1-a)^k = 1- ka + k(k-1)a^2/2! – k(k-1)(k-2)\, a^3/3! +..}$

$\mathbf{P_1(2^n, k)} = \mathbf{k.\, 2^{-n} – k.(k-1).\, 2^{-2n}/2 + k.(k-1).(k-2).\, 2^{-3n}/6 - ..}$ ecc.
$= \mathbf{k.\, 2^{-n}}$ quando $\mathbf{2^{-n}}$ è molto piccolo

S: probabilità di successo desiderata

$$\mathbf{S = P_1(2^n, k) \rightarrow k = S.2^n}$$
$$\mathbf{O(exp(n))}$$
$$\mathbf{n \geq 80}$$

Il paradosso del giorno del compleanno

Birthday paradox

Nell'ipotesi che le date di nascita siano equiprobabili, è sufficiente scegliere a caso **253** persone per avere una probabilità > 0,5 che una di queste compia gli anni in un dato giorno.

Sono invece sufficienti **23** persone scelte a caso per avere una probabilità > 0,5 che due o più compiano gli anni nello stesso giorno.

R20: resistenza forte alle collisioni

$P_2(2^n, k)$ probabilità di due uscite identiche con $k \leq 2^n$ ingressi scelti a caso
- sequenze d'uscita possibili: $(2^n)^k$ differenti
- sequenze con valori tutti diversi: $2^n! / (2^n - k)!$

$$P_2(2^n, k) = 1 - 2^n! / (2^n)^k (2^n - k)! = 1 - 2^n \times (2^n-1) \times (2^n-2) \times \ldots \times (2^n-k+1)/2^{n\,k}$$
$$= 1 - (1-1/2^n)(1-2/2^n) \ldots (1-(k-1)/2^n)$$

N.B. $(1-x) \leq e^{-x}$, *valida per* $x \geq 0$, *è una buona approssimazione per* $x < 0,3$

$$P_2(2^n, k) \cong 1-\exp[-2^{-n}(1+2+\ldots+(k-1)]$$
$$= 1 - \exp[-2^{-n}(k(k-1)/2)] \text{ e per k grande}$$
$$\cong 1 - \exp[-2^{-n}(k^2/2)]$$

IPOTESI: $P_2 = ½$

$$1 - ½ = \exp[-2^{-n}(k^2/2)]$$
$$\ln 2 = 2^{-n} \times (k^2/2)$$
$$k = \text{sqrt}[2 \times (\ln 2) \times 2^n] = 1,18 \times 2^{n/2}$$
$$O(\exp(n/2))$$

Paradosso del compleanno: **k = sqrt(2.ln(2)).sqrt(365) = 22,54**.

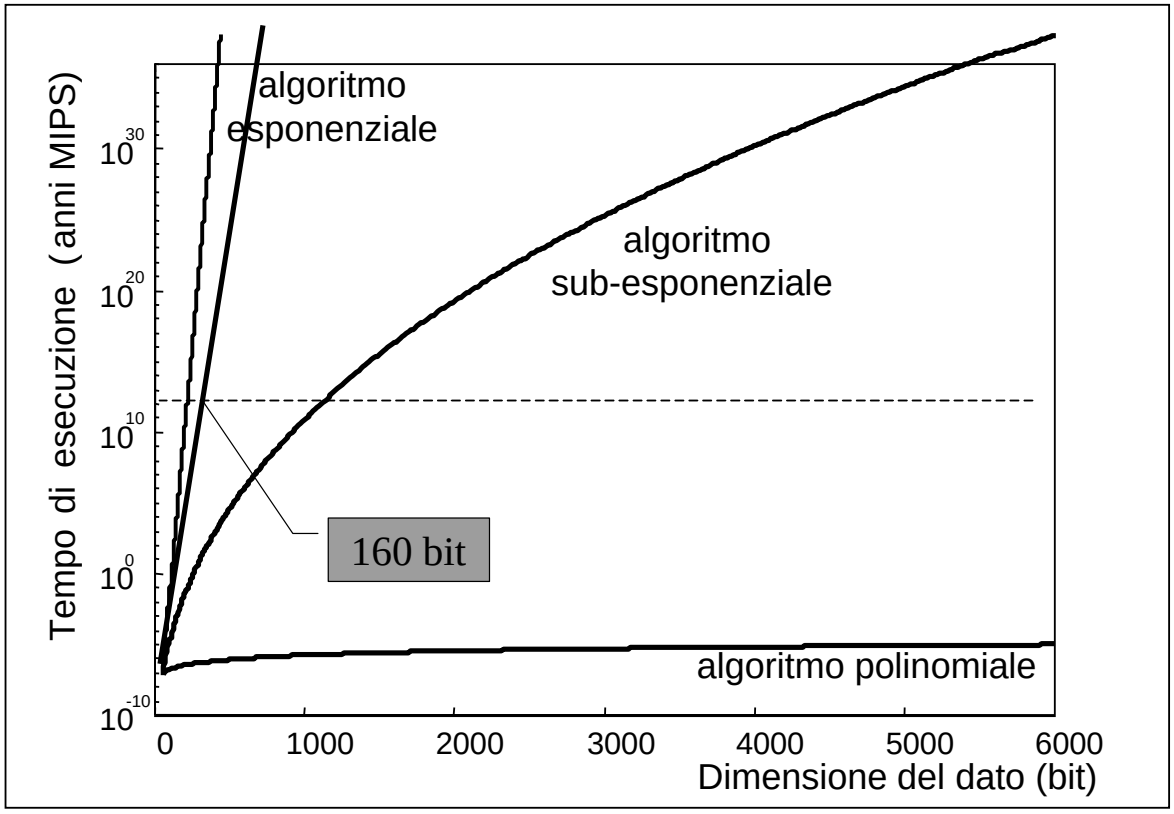
Birthday attack

L'attaccante che vuole due messaggi con la stessa impronta
1. genera $2^{n/2}$ piccole varianti del primo messaggio
2. calcola e memorizza gli hash
3. Modifica lievemente il secondo messaggio, calcola l'hash e controlla se è in memoria; in caso contrario ripete

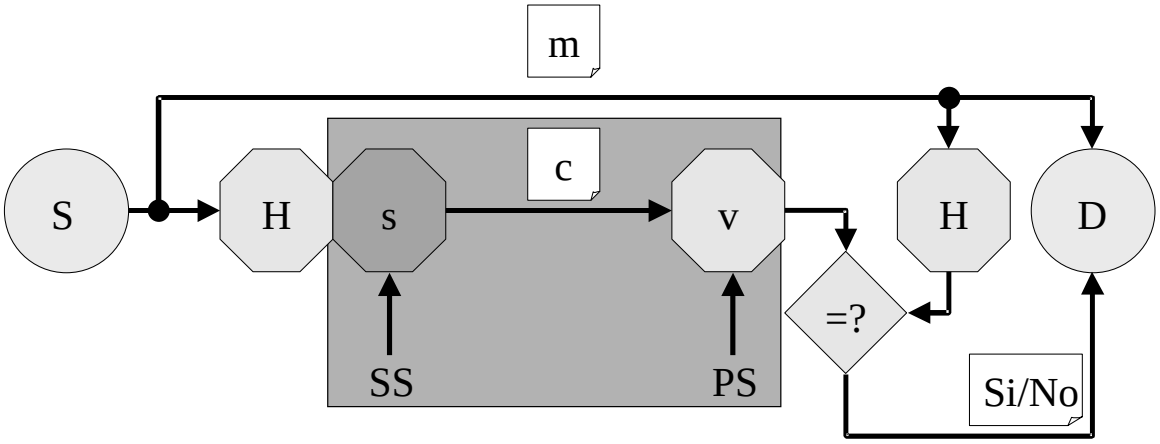
O(exp(n/2))

Programma di prova in Java per hash di 40 bit

Robustezza forte alle collisioni



Resistenza alle collisioni e firma digitale



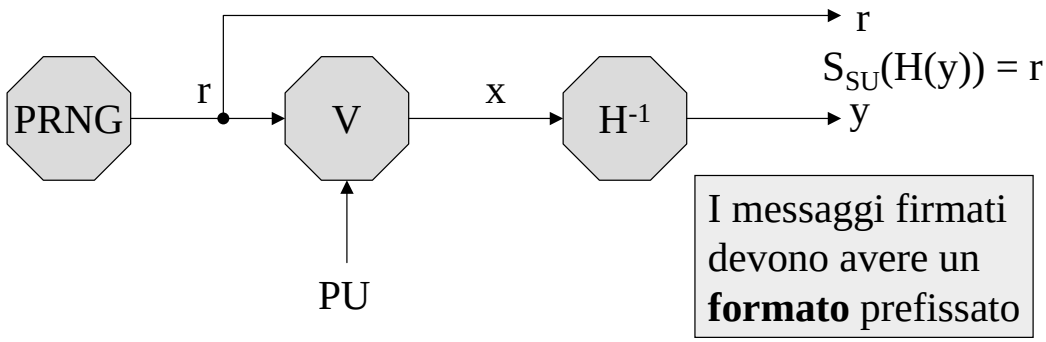
Quando valgono R19, R20 l'impronta **H(m) identifica m**

Con la firma di H(m) si ottiene:

- efficienza
- individuazione di modifiche a **m** e/o a **c** apportate dall'intruso
- S non può sostenere di aver inviato **m*** e non **m**
- D non può sostenere di aver ricevuto da S un **m*** da lui inventato

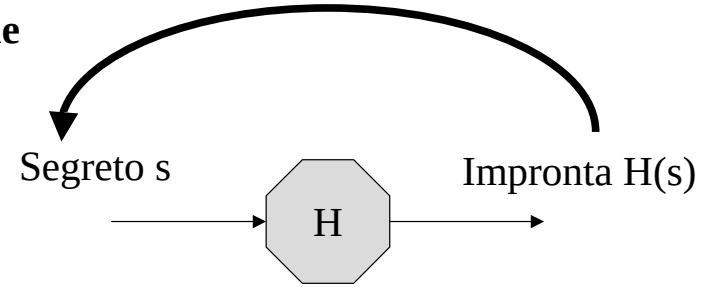
R21: unidirezionalità (non invertibilità)

Firma



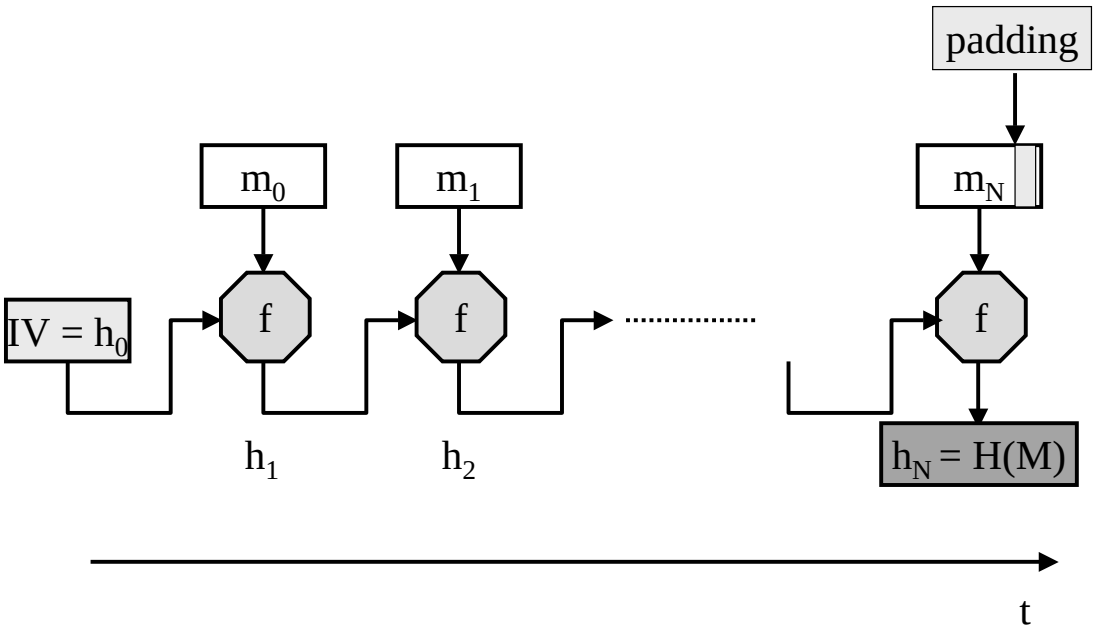
I messaggi firmati devono avere un **formato** prefissato

Identificazione



Hash: cifratura senza decifrazione!

Algoritmi di hash: compressione iterata

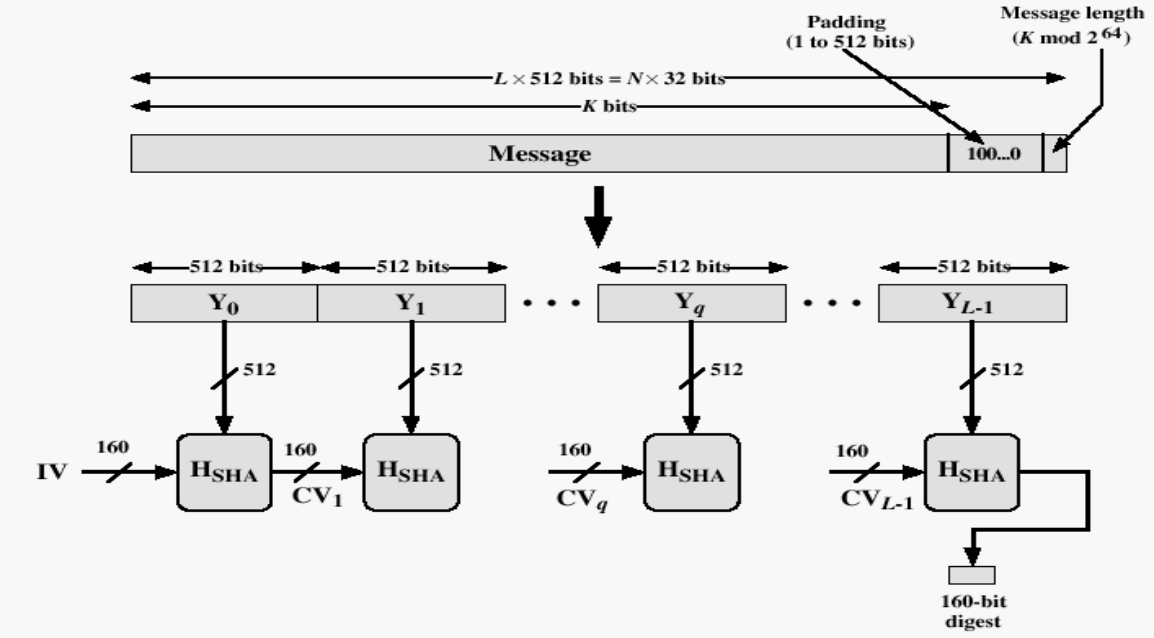


Secure HASH functions

Algoritmi di hash più noti ed usati	**MD5 1991**	**SHA-1 1994**	**RIPEMD-160 1996**
Digest length	128 bits		160 bits
Basic unit of processing	512 bits		512 bits
Number of steps	64 (4 rounds of 16)	80 (4 rounds of 20)	160 (5 paired rounds of 16)
Maximum message size	∞	2^{64}-1 bits	∞

NIST 2002:
SHA-256
SHA-384
SHA-512

Message Digest Generation Using SHA-1



SHA-1 Processing of single 512-Bit Block

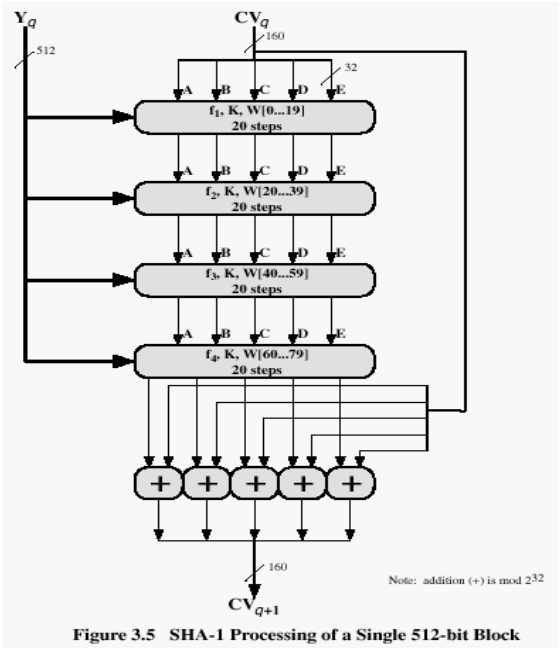


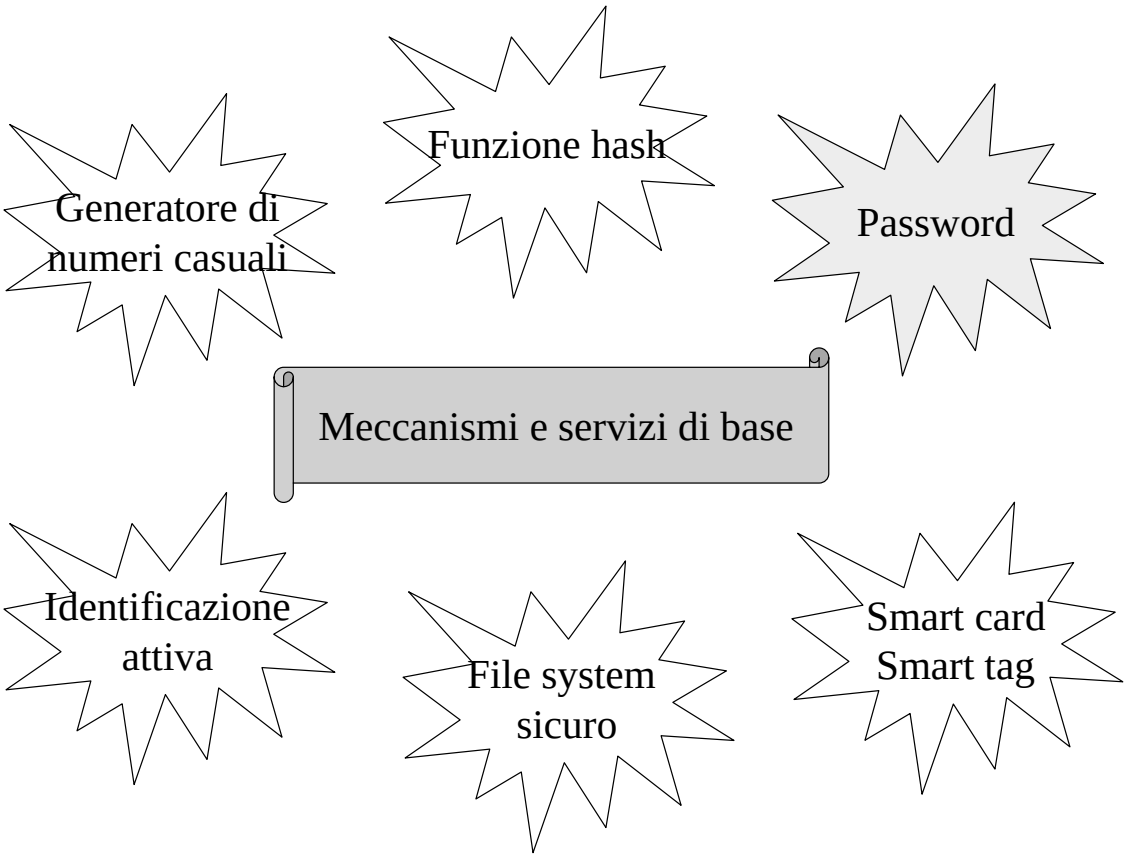
Figure 3.5 SHA-1 Processing of a Single 512-bit Block

Ogni bit di hash dipende
da ogni bit di input.

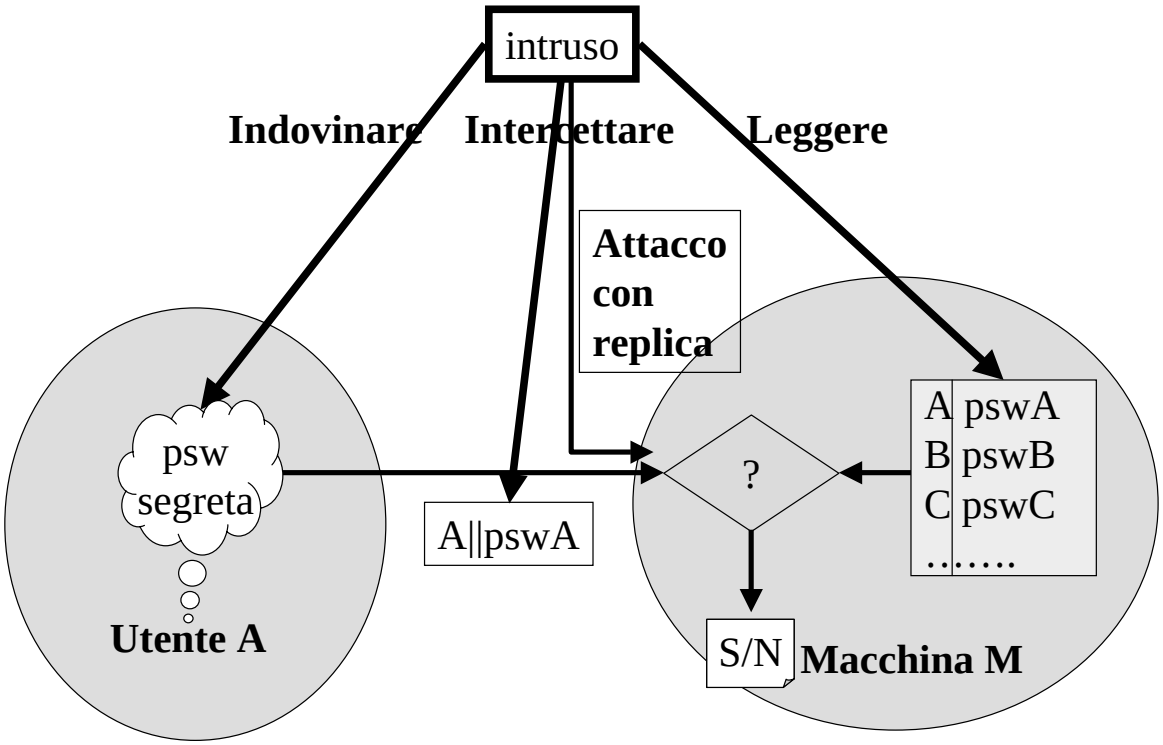
<u>Complessità degli attacchi</u>

2^{80}: numero di operazioni
necessario per ottenere
due messaggi con lo stesso hash

2^{160}: numero di operazioni
necessario per ottenere
un messaggio con un dato hash



Identificazione passiva: la password



Difesa della password

Memorizzazione a prova di furto

File di password cifrate

Segreto non indovinabile

lunga stringa casuale
utente, sistema, passphrase

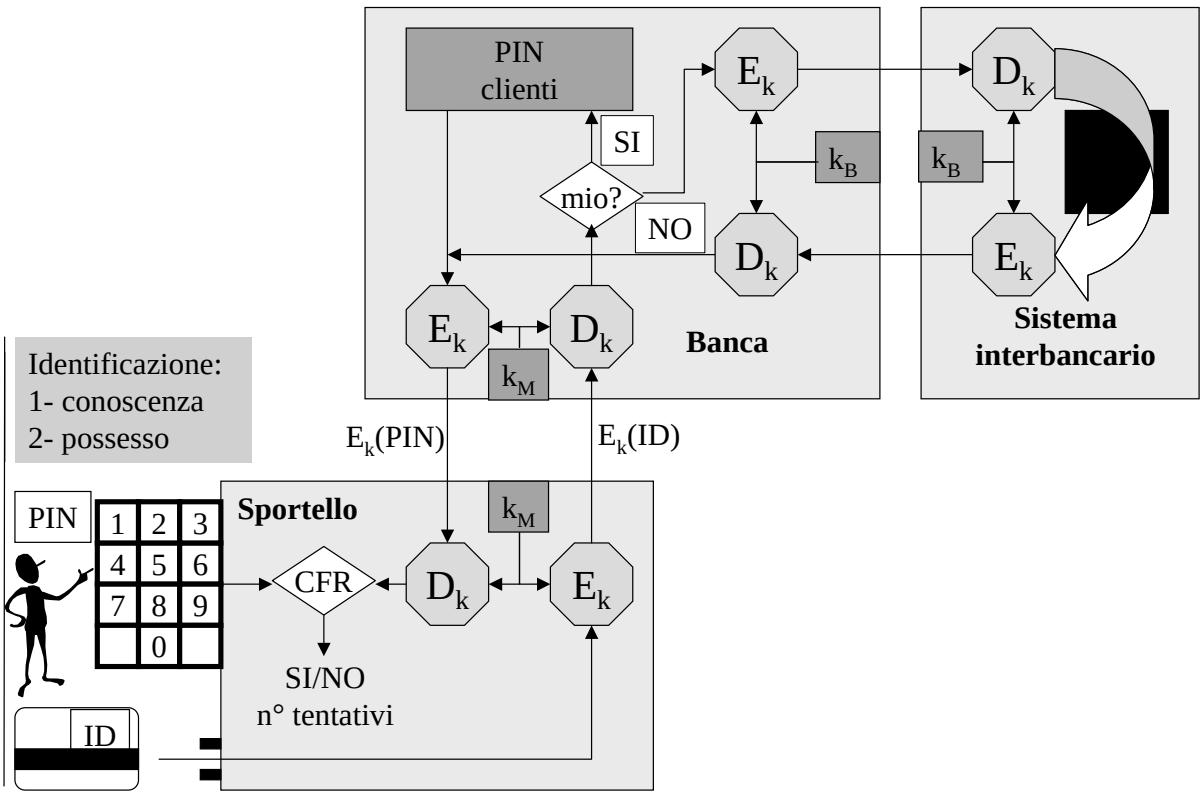
R22: almeno 8 simboli casuali
L.196/03

Canale a prova di intercettazione

“eco”, sportello, inaccessibilità fisica

N.B. la cifratura è inutile!

Bancomat: identificazione del cliente (cap.7)



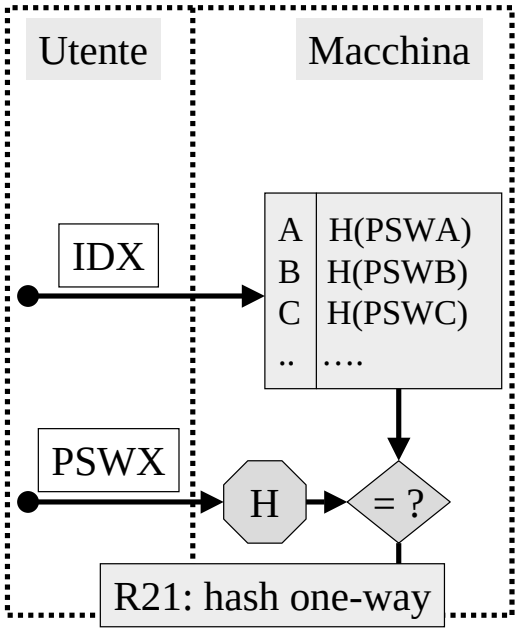
Difesa del file delle password

R14bis: "*L'accesso in scrittura al file delle psw deve essere consentito solo all'amministratore del servizio*"

R15bis: "*L'accesso in lettura al file delle psw deve restituire all'intruso dati non utilizzabili per farsi identificare come utente legittimo*"

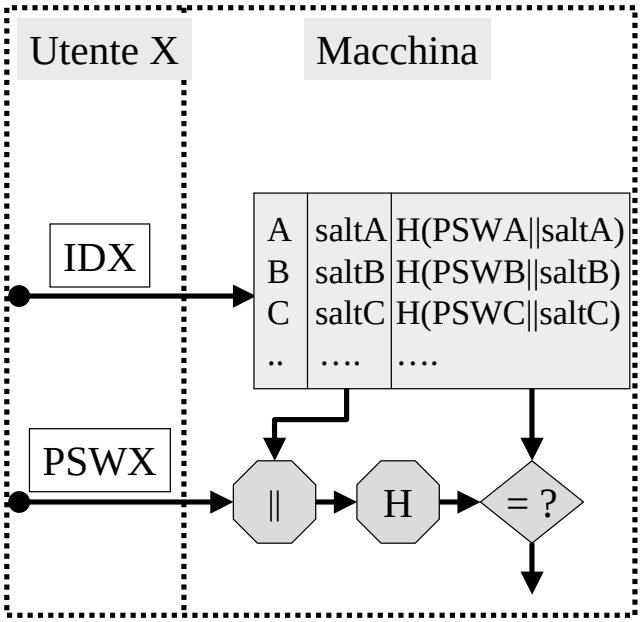
Hash della password: forma di cifratura per cui non esiste decifrazione!

hash delle password

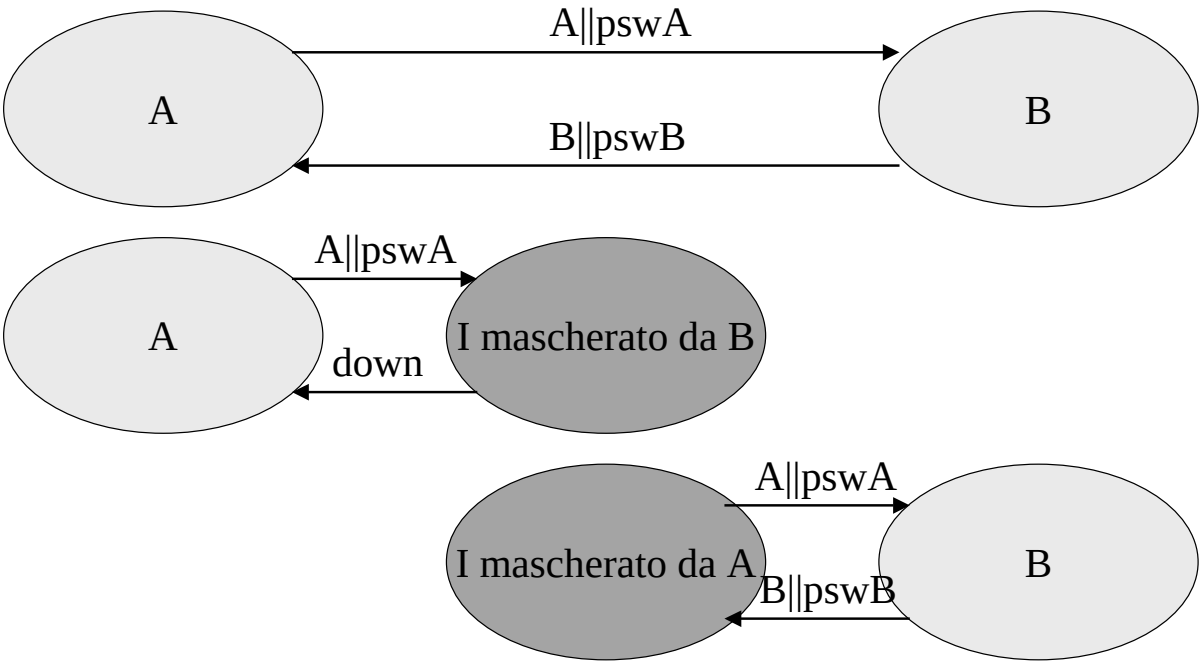


Attacco con dizionario

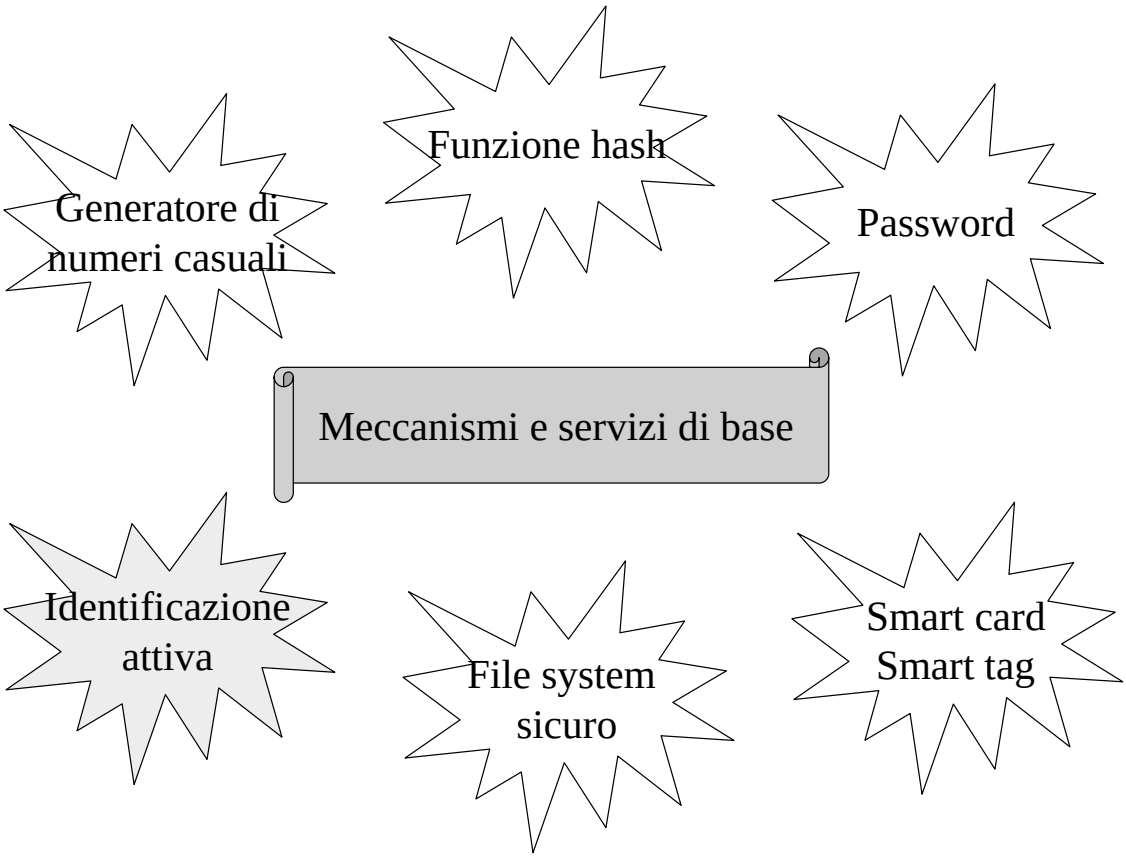
hash delle password e dei salt



Identificazione unilaterale e reciproca



Solo l'identificazione attiva può essere anche mutua



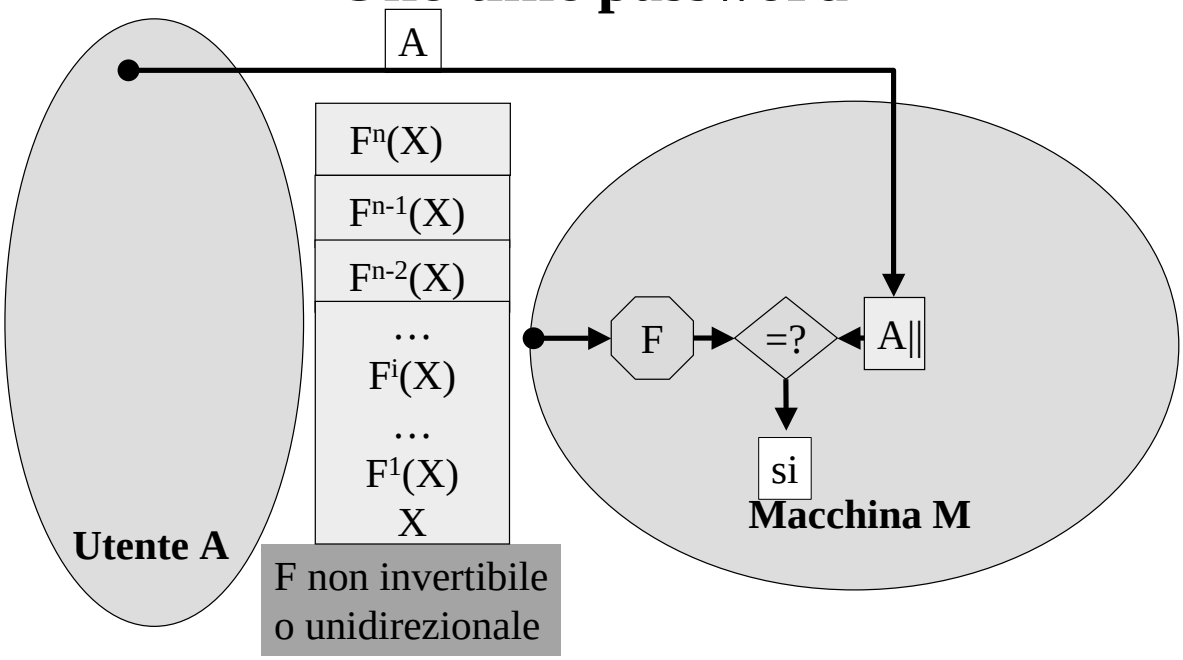
Identificazione attiva

attiva (o forte)
A→B: F^i(segreto)

- one-time psw
- challenge/response
- zero knowledge

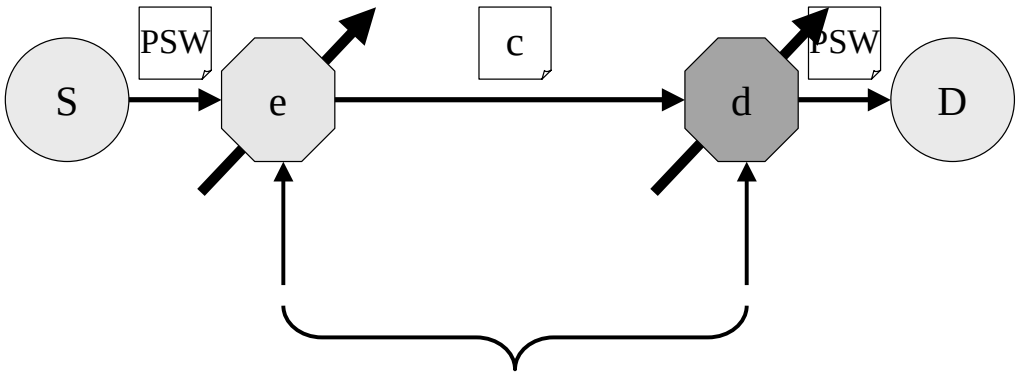
R23: "*Il calcolo della prova d'identità da fornire* ***di volta in volta*** *deve essere facile per chi conosce un'informazione segreta, difficile per chi dispone soltanto delle prove inviate in precedenza*"

One-time password



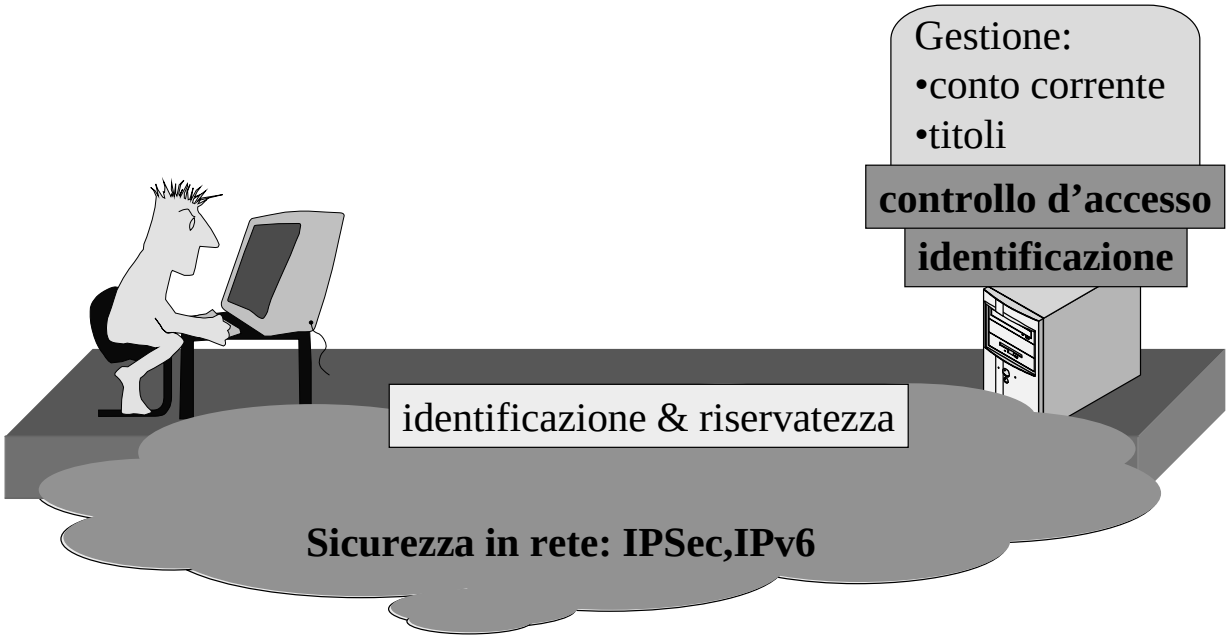
F non invertibile
o unidirezionale

Cifratura one-time

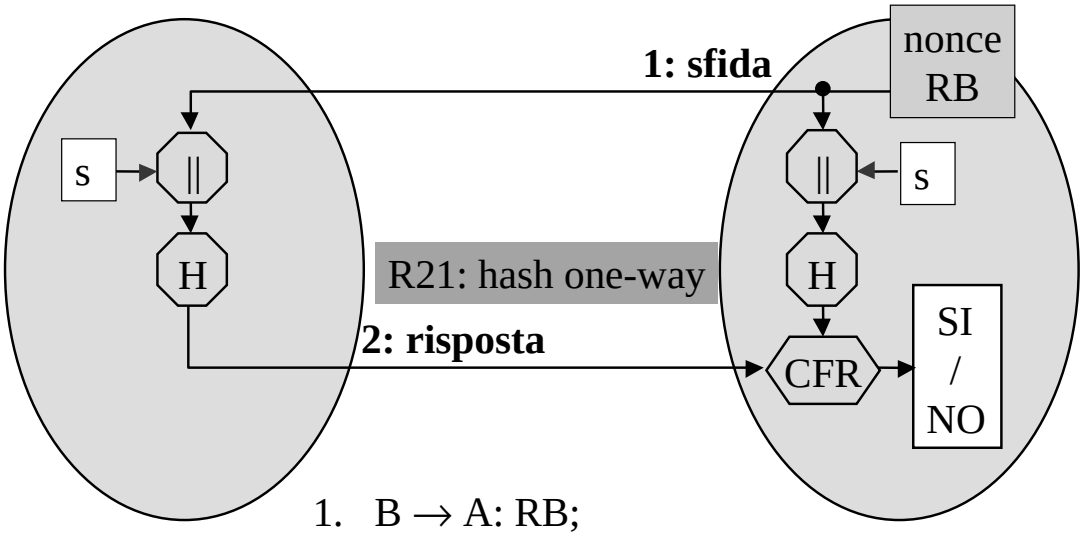


Session key
One-time key

Home banking (cap.7)



Sfida/Risposta (Hash)



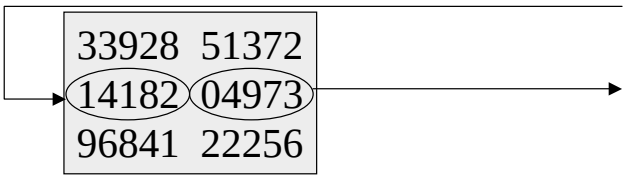
1. $B \rightarrow A$: RB;
2. $A \rightarrow B$: cA = H(RB || s).

Home banking: identificazione utente (cap.7)

1:ID & password

HTTPS: Cilma & Multitel

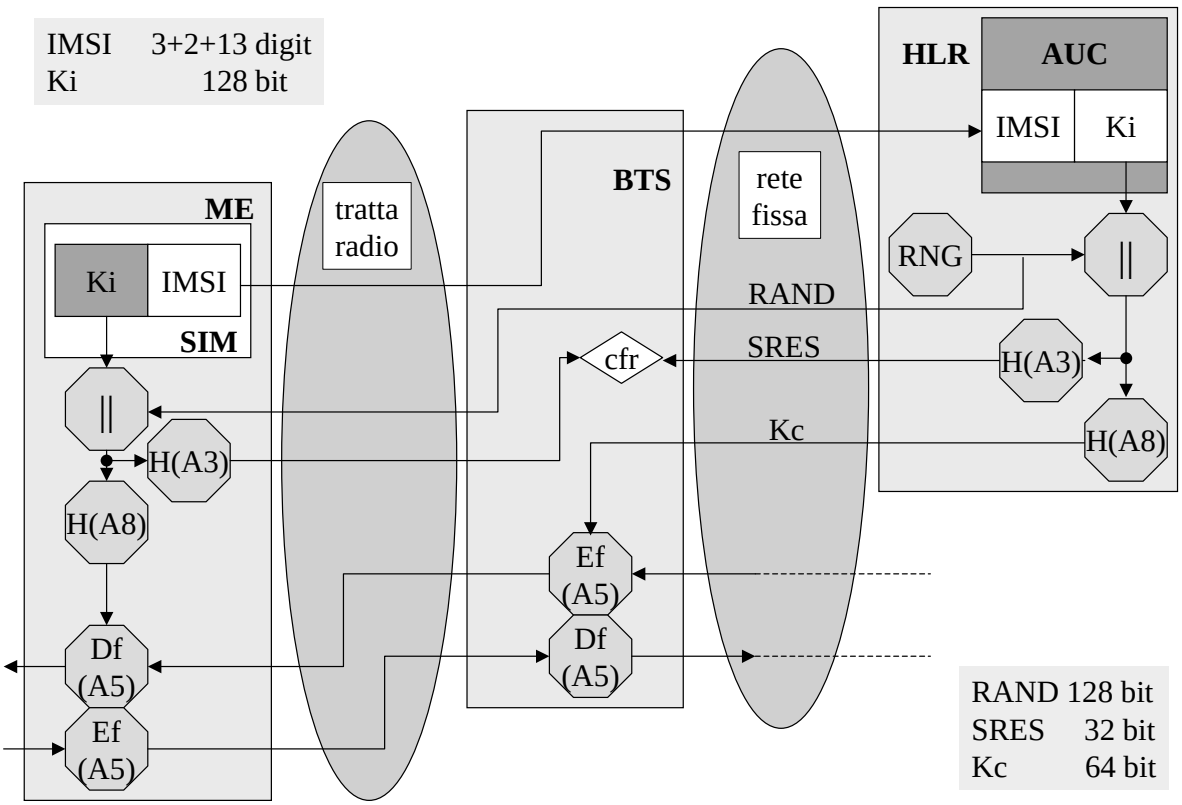
2: sfida/risposta



3: smart card

Sfida/Risposta “vera”

GSM: identificazione e riservatezza (cap.7)



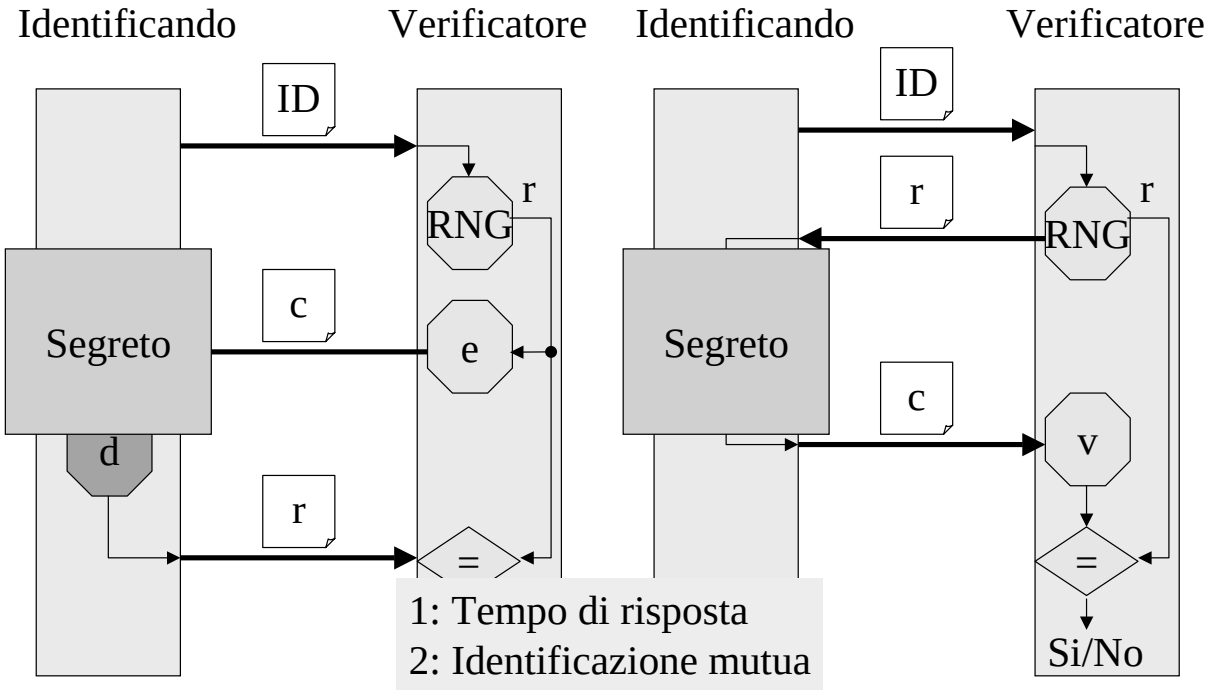
Identificazione mutua

1. B → A: RB;
2. A → B: cA = RA || H(RA || RB || **B** || s);
3. B → A: cB = H(RA || RB || **A** || s).

Il problema del Gran Maestro di scacchi – *A vuole spacciarsi per un grande esperto di scacchi pur non conoscendo il gioco. A sfida due Gran Maestri B e C, che sistema, senza che se ne accorgano, in due camere contigue: a B assegna i "bianchi", a C i "neri". Preso nota della prima mossa di B, A corre nell'altra stanza e la riproduce sulla scacchiera di C. Successivamente prende nota della contromossa di C e corre a riprodurla sulla scacchiera di B. Continuando così ottiene o due patte o un'incredibile vittoria.*

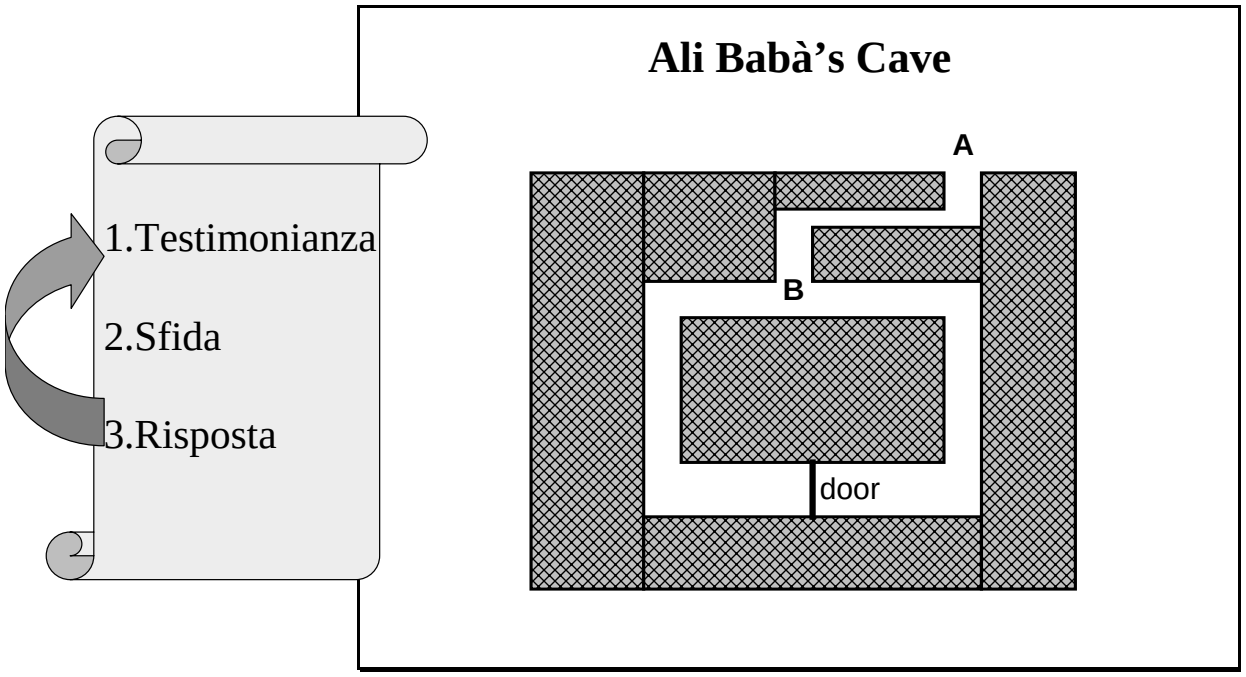
Marca temporale (time stamp) Numero di sequenza

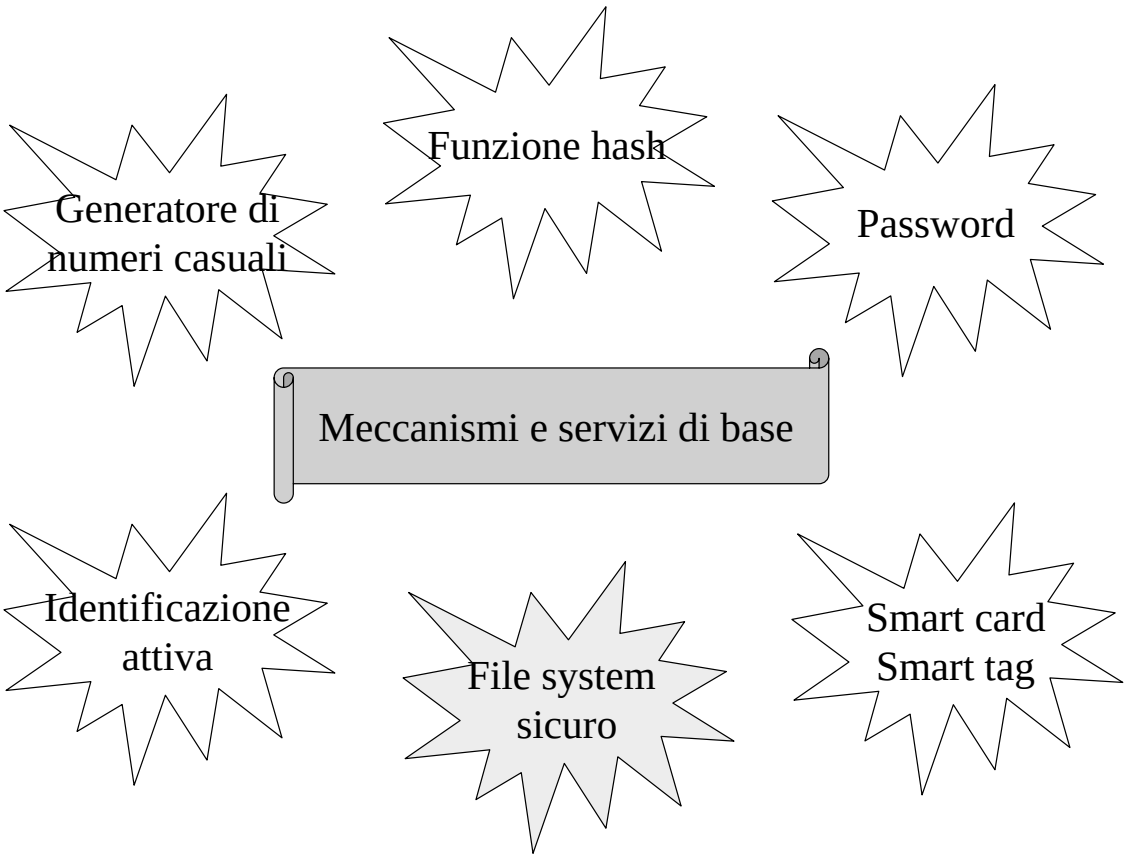
Il protocollo sfida/risposta (Cifrario, Firma digitale)



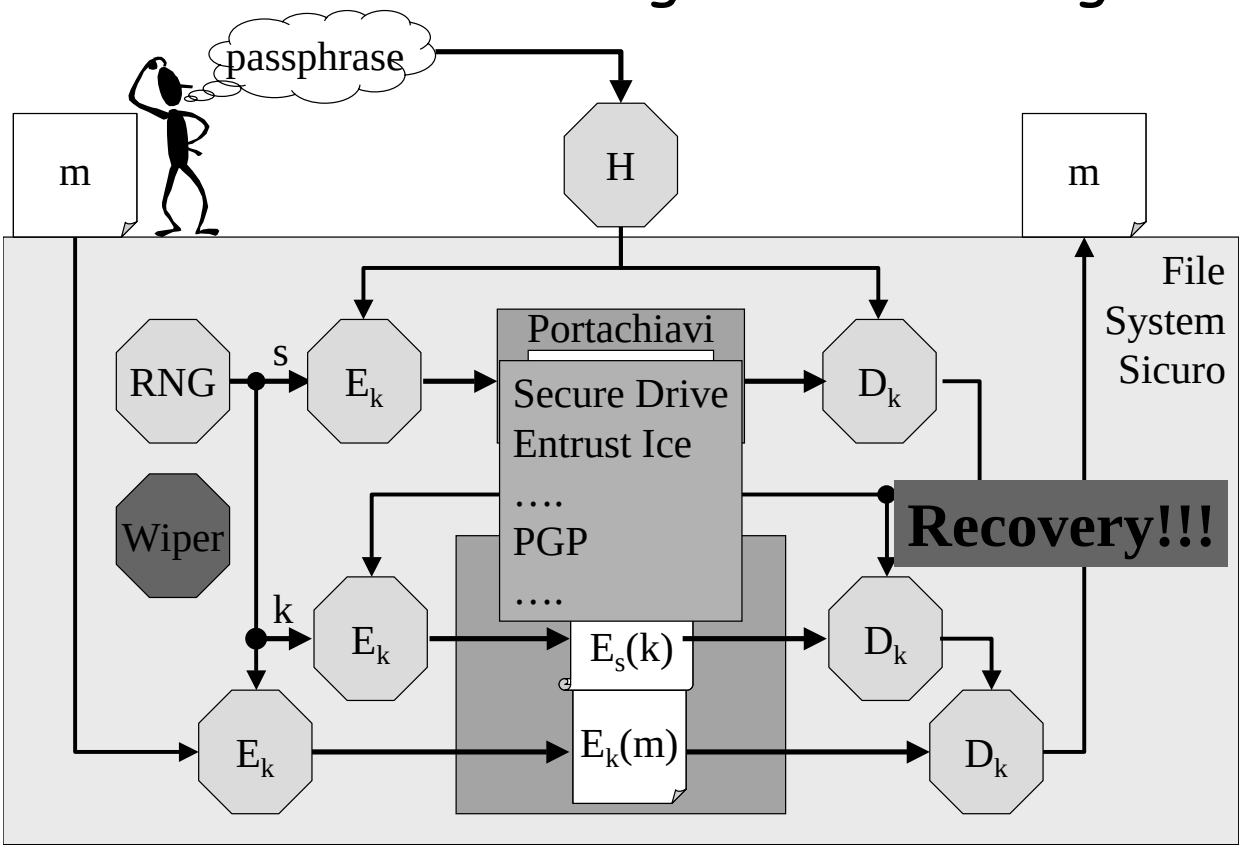
Zero-knowledge protocols

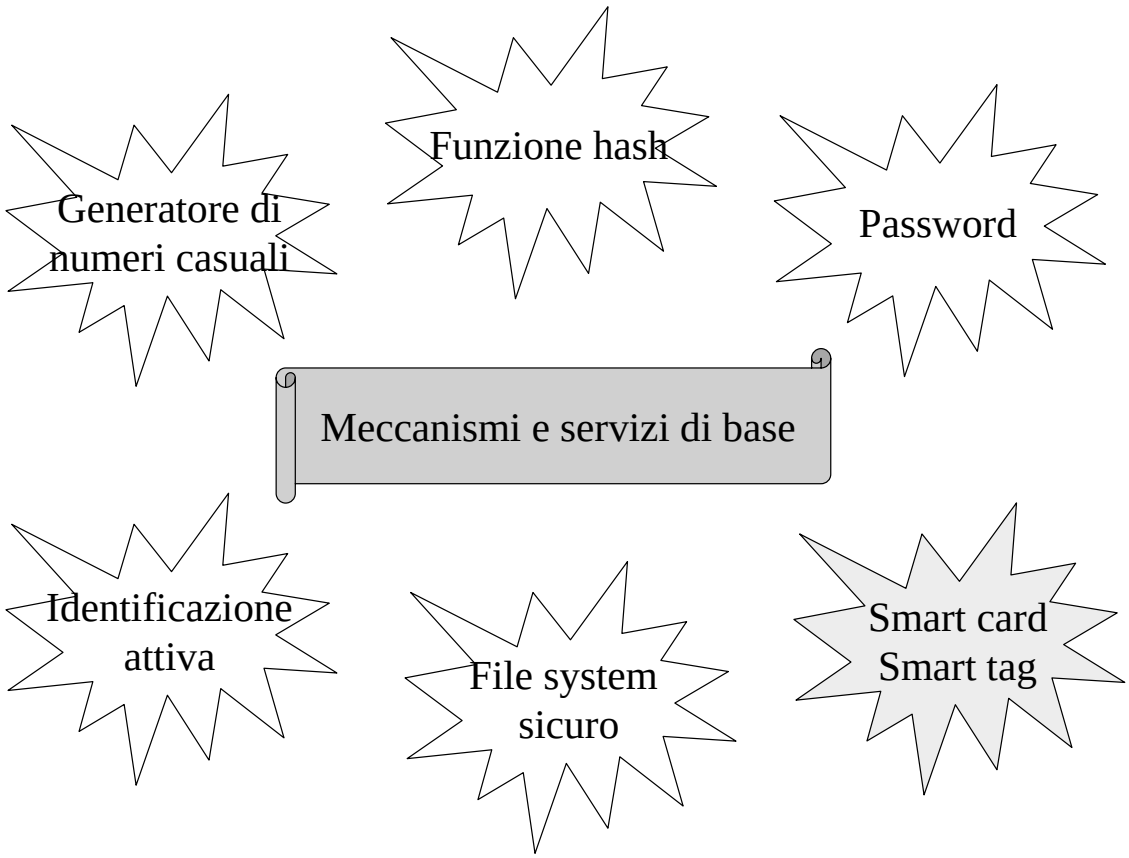
Principio: dare solo una testimonianza di saper risolvere facilmente un problema da tutti ritenuto difficile





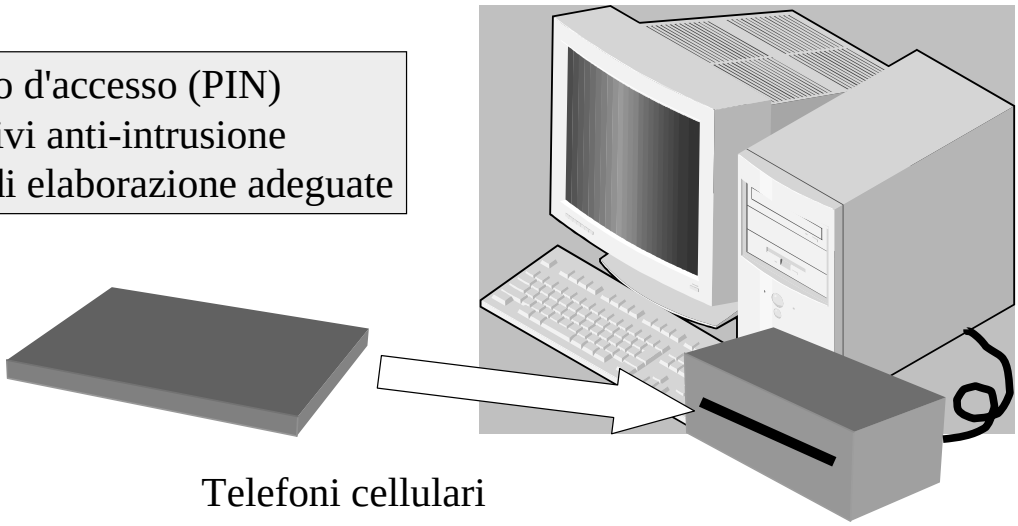
FSS: i tre livelli della gerarchia di segreti





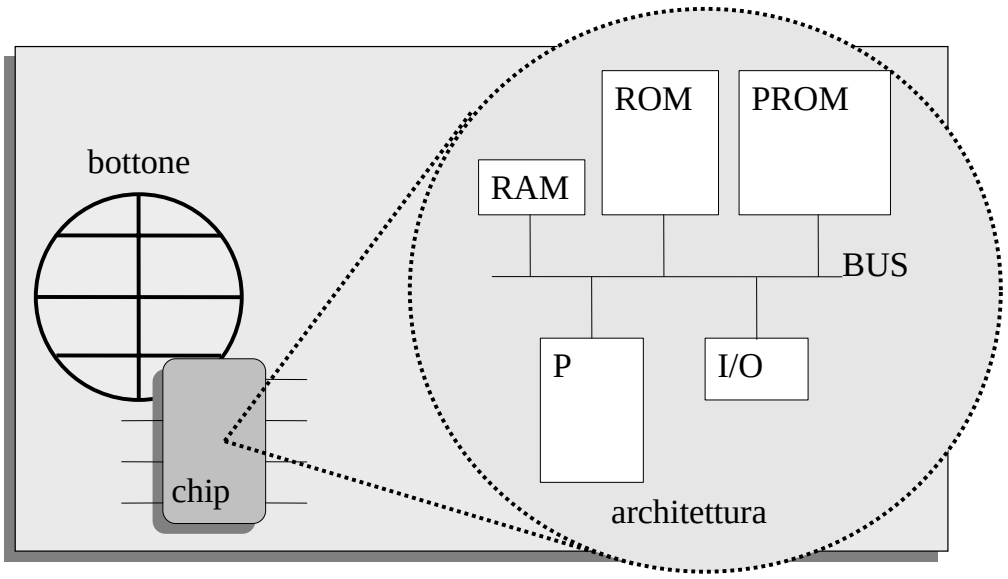
Il calcolatore portatile e personale

controllo d'accesso (PIN)
dispositivi anti-intrusione
risorse di elaborazione adeguate



Telefoni cellulari
Home banking
Carta d’identità
Passaporto europeo
….
Registrazione esami

Smart card a contatto



Tipi e Standard

- a contatto
- senza contatto
- a prossimità

- ISO 7816
- Microsoft Crypto API
- PKCS#11
- PKCS#15
- PC/SC Workgroup

Smart tag

Supermecato: bancone e magazzino
carrello e conto della spesa
cassa automatica (occupazione?)
Portafoglio (privacy?)
Pascolo
Maratona
Automazione di fabbrica
….

IBM, HP
RSA
Intel