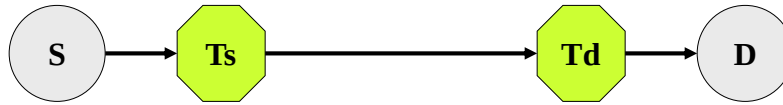




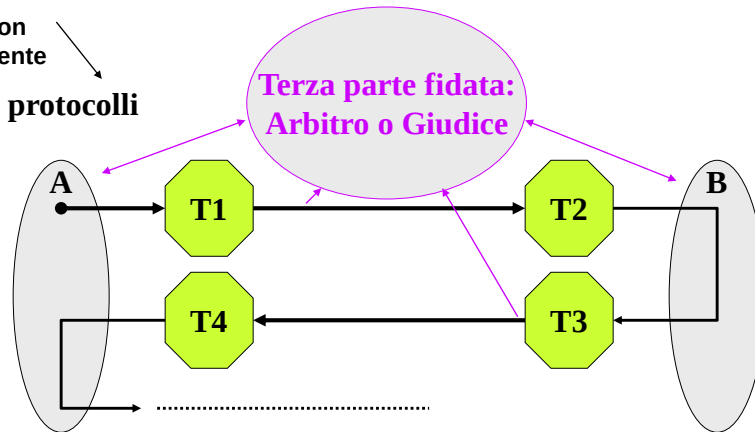
Trasformazioni per la sicurezza

1: algoritmi



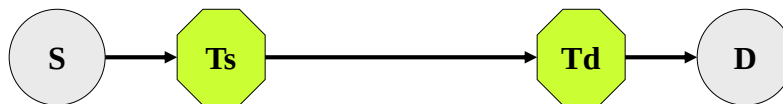
Se non
sufficiente

2: protocolli



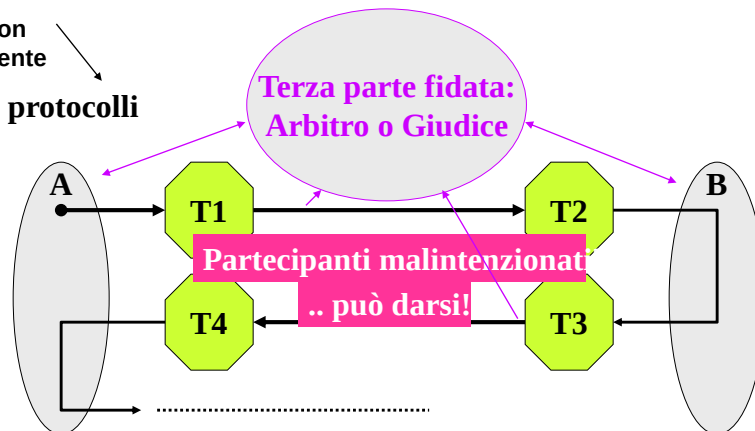
Trasformazioni per la sicurezza

1: algoritmi



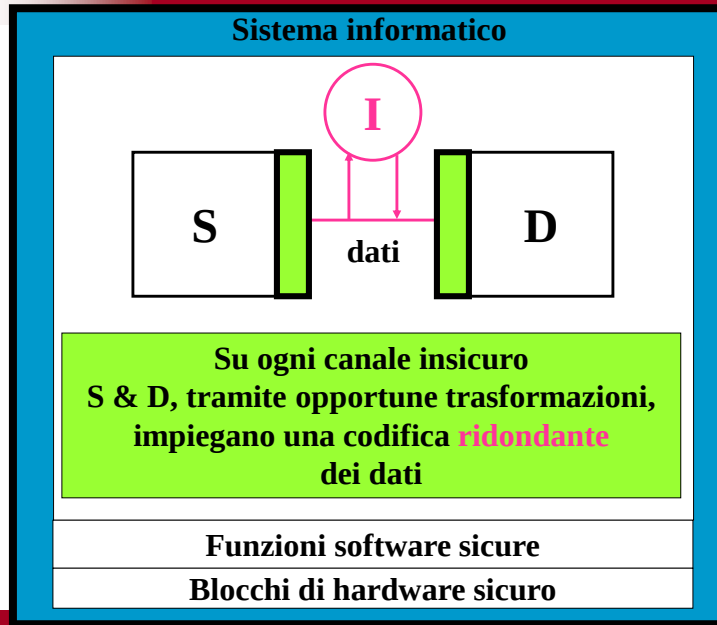
Se non
sufficiente

2: protocolli

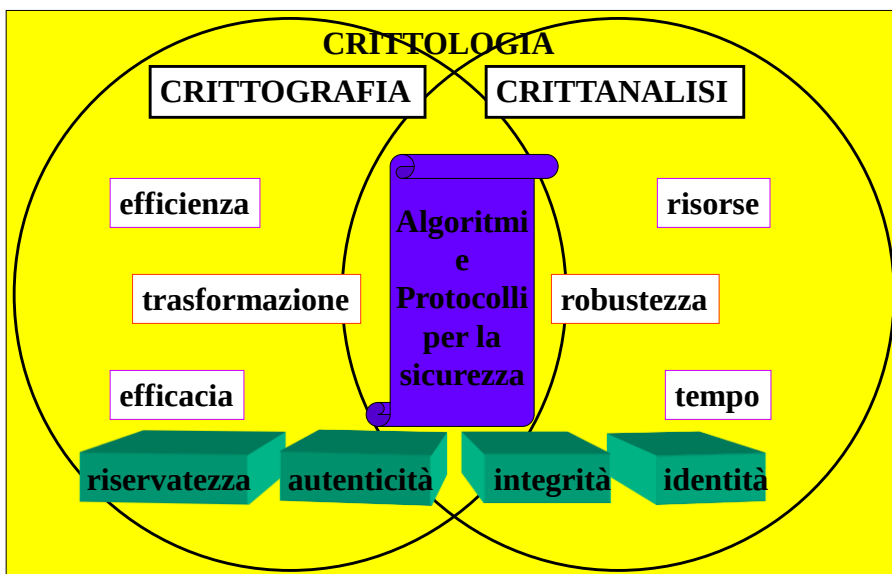




La tecnica che può rendere i dati sicuri



Crittografia e Crittanalisi





1940-44



Crittografia classica

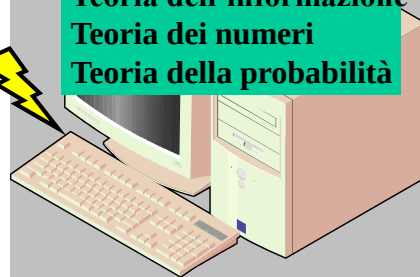
Algoritmi semplici



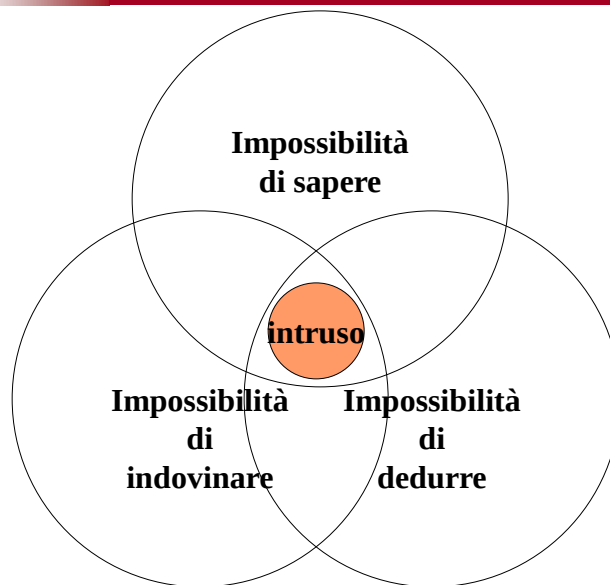
Algoritmi complessi

Crittografia moderna

Teoria degli algoritmi
Teoria dell'informazione
Teoria dei numeri
Teoria della probabilità



I principi della difesa





I principi della difesa

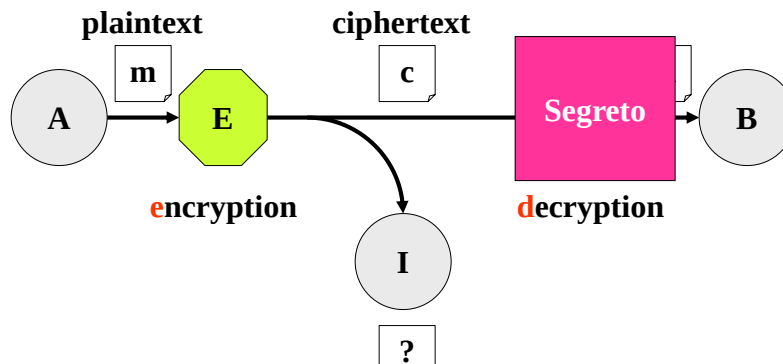


Riservatezza



Elaborazioni per la riservatezza di un messaggio

La sorgente trasforma la rappresentazione originaria delle informazioni riservate in una rappresentazione che le renda apparentemente incomprensibili; la destinazione è l'unica a saper eseguire la trasformazione inversa

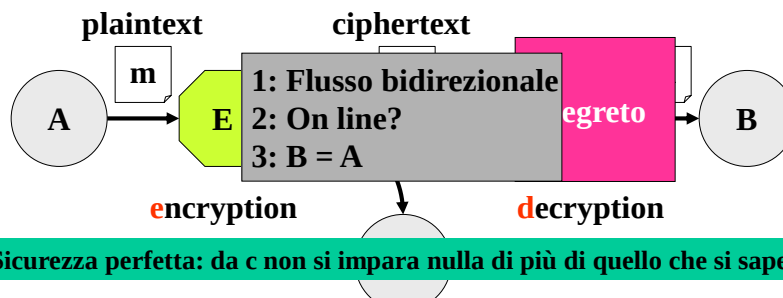


Intro al Corso – Sicurezza dell'Informazione M



Elaborazioni per la riservatezza di un messaggio

La sorgente trasforma la rappresentazione originaria delle informazioni riservate in una rappresentazione che le renda apparentemente incomprensibili; la destinazione è l'unica a saper eseguire la trasformazione inversa



Sicurezza perfetta: da c non si impara nulla di più di quello che si sapeva già

I calcoli per mettere in chiaro un testo cifrato senza conoscere l'algoritmo di decifrazione devono essere difficili

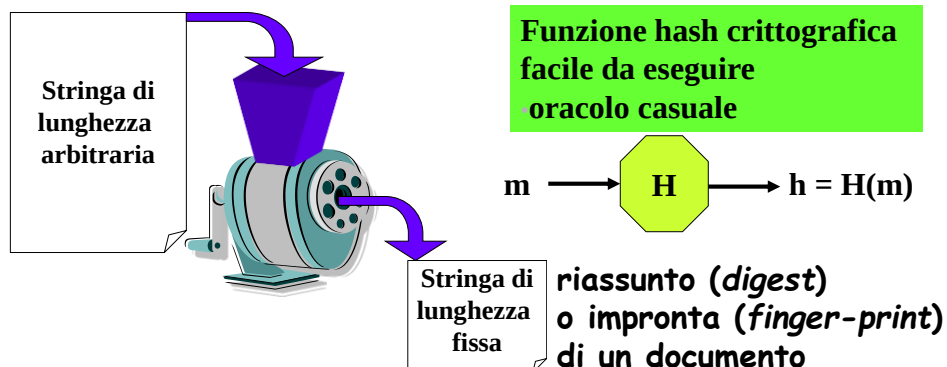


Integrità



Integrità: rilevazione di attacchi intenzionali

La sorgente deve affiancare al documento un “riassunto” che ne rappresenti in modo pressoché univoco il contenuto; la destinazione deve ricalcolare il riassunto e confrontare i due dati



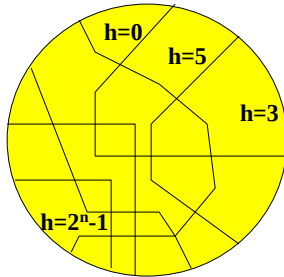
Resistenza alle collisioni: l'individuazione di due messaggi con la stessa impronta deve essere un calcolo difficile



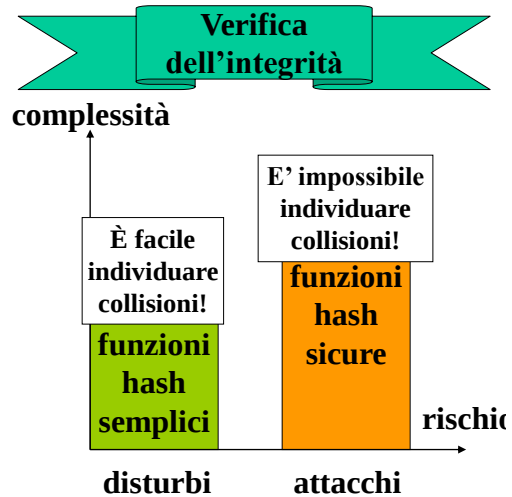
Collisioni

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

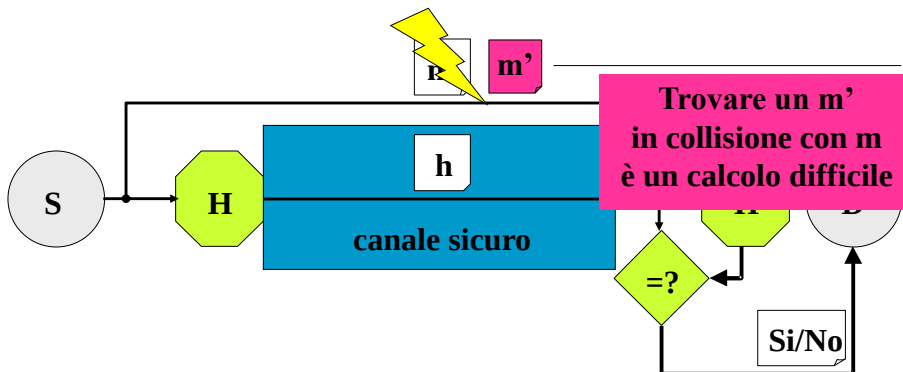
Input space: 2^m con $m > n$



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

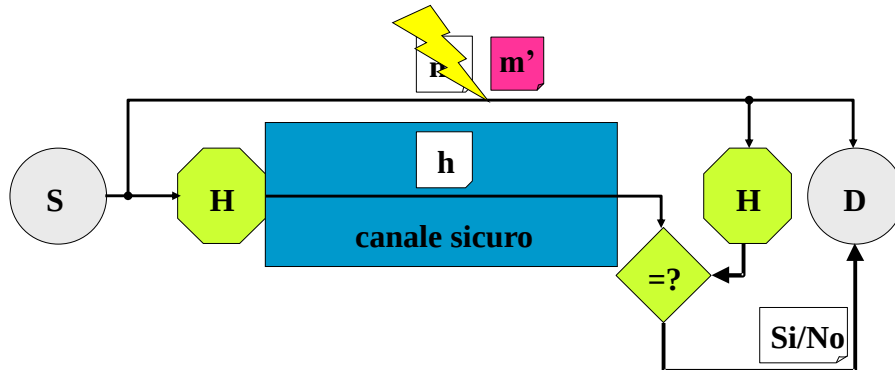


Attestazione ed accertamento dell'integrità





Attestazione ed accertamento dell'integrità

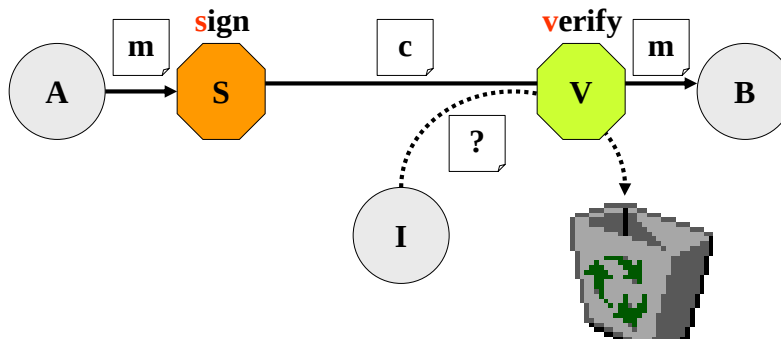


Produttori di software
Portachiavi del PGP



Elaborazioni per l'autenticazione di un messaggio

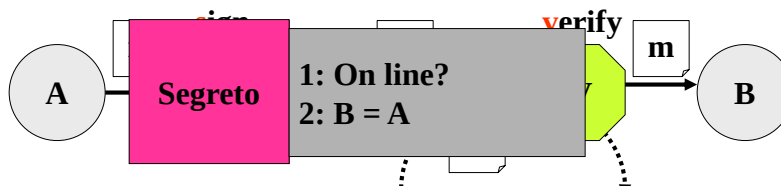
La sorgente aggiunge al documento informazioni non imitabili e atte ad attestare chi l'ha predisposto; la destinazione verifica che il documento ricevuto sia stato originato proprio da chi dichiara di averlo fatto".





Elaborazioni per l'autenticazione di un messaggio

La sorgente aggiunge al documento informazioni non imitabili e atte ad attestare chi l'ha predisposto; la destinazione verifica che il documento ricevuto sia stato originato proprio da chi dichiara di averlo fatto".



I calcoli per costruire un messaggio apparentemente autentico senza conoscere la trasformazione della sorgente devono essere difficili



Protocollo per la difesa di Riservatezza & Integrità

1. $p = m || H(m)$ *testo in chiaro concatenato con l'attestazione d'integrità*
2. $c = E(p)$ *testo cifrato trasmesso*
3. $p^* = D(c^*) = m^* || H^*(m)$ *testo in chiaro ricevuto*
4. $H^*(m) = ? H(m^*)$ *controllo d'integrità*

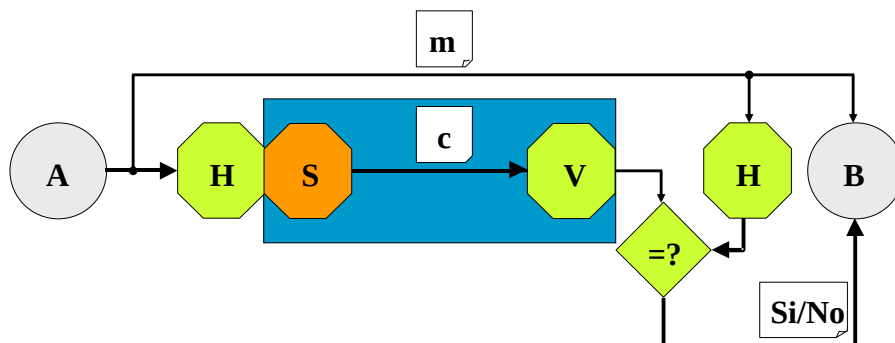


Autenticazione



Autenticazione di un messaggio in chiaro

1: schema firma digitale



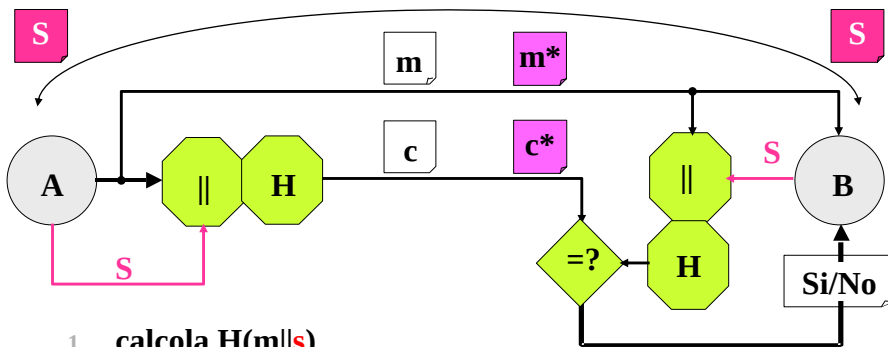
**NON
ripudiabile**

- a) canale sicuro
- b) canale reso sicuro



Autenticazione di un messaggio in chiaro

2: hash del messaggio e di un segreto



1. calcola $H(m||s)$

2. invia m e $H(m||s)$

3. riceve m^* e $H^*(m||s)$

4. calcola $H(m^*||s)$

5. $H^*(m||s) = ? H(m^*||s)$

ripudiabile

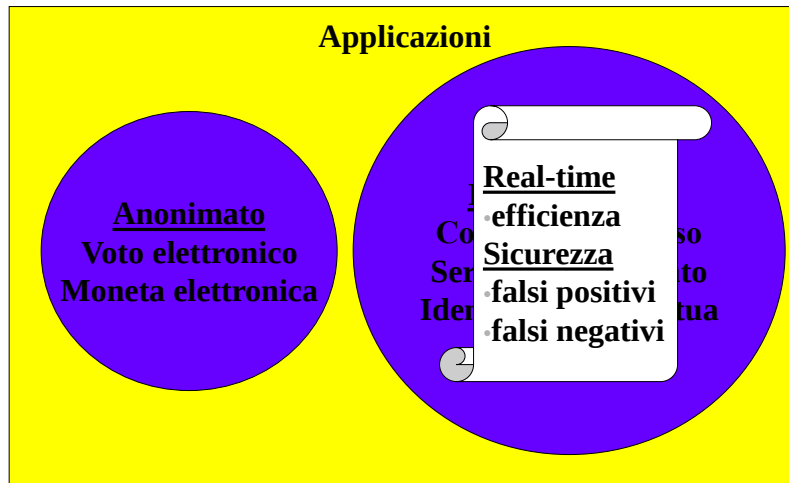
H^{-1}
difficile



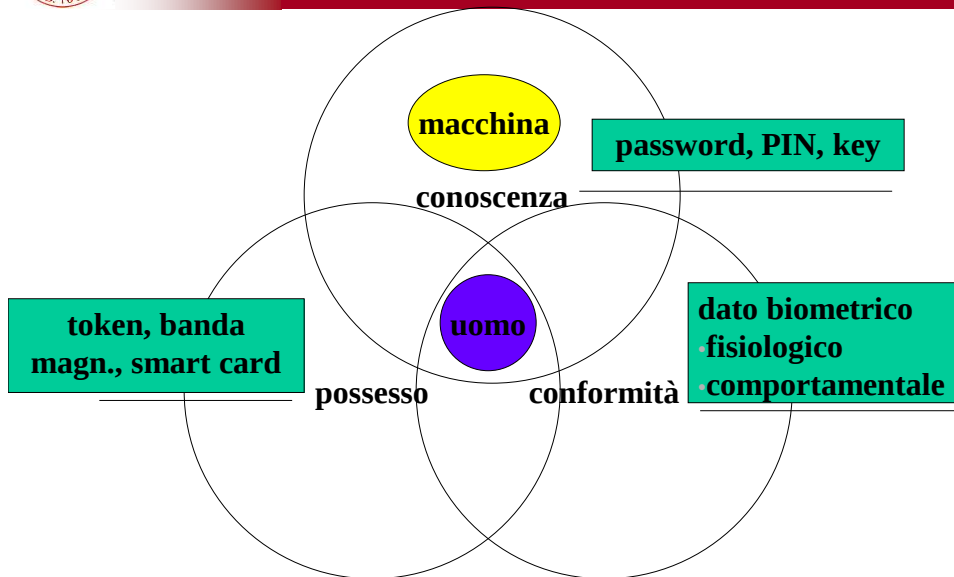
Identificazione



Anonimato/Identificazione

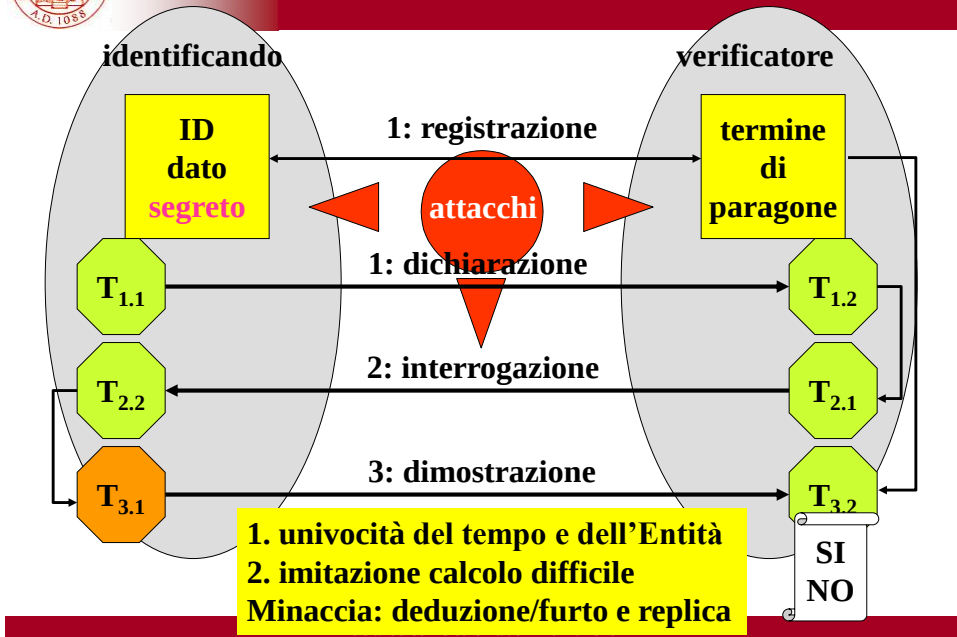


Strumenti di identificazione





Il protocollo d'identificazione



Funzioni one-way



Il modello della funzione unidirezionale (one-way function)

L'intruso non deve riuscire ad invertire

- una funzione hash
- una funzione di cifratura
- una funzione di verifica

.....

Una funzione f è detta **unidirezionale** se
è **invertibile**,
facile da calcolare
e se per quasi tutti gli x appartenenti al dominio di f è **difficile** risolvere per x il problema $y = f(x)$.



Funzione unidirezionale e pseudo-unidirezionale



Elenco telefonico

N° di X?

Ricerca binaria: $O(n)$

X di N°?

Ricerca esaustiva: $O(2^n)$

113? Carabinieri!

In teoria non esistono, in pratica SI:

- Manipolazioni a livello di bit
- Problemi difficili della Teoria dei numeri

Una funzione F è detta **pseudo-unidirezionale** (trapdoor one-way) se appare come unidirezionale per chiunque non sia in possesso di una particolare informazione sulla sua costruzione



Trasformazioni segrete



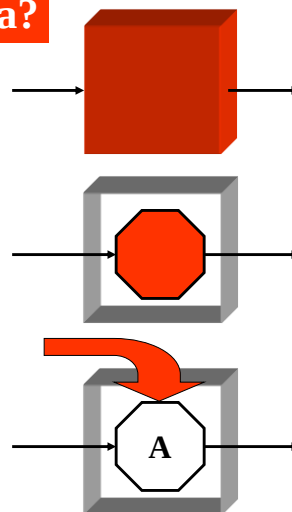
Trasformazioni segrete

Decifrazione

Autenticazione

Segretezza sì, ma di che cosa?

- **Algoritmo**
- **Parametro**





Vulnerabilità nelle macchine e degli algoritmi segreti



- Ispezione
- Installazione
- Progetto
- Produzione
- Certificazione



Algoritmo pubblico e parametro segreto

Kerckhoffs : “La cryptographie militaire”1883



Responsabilità dell'utente

cassaforte

Valutazione pubblica

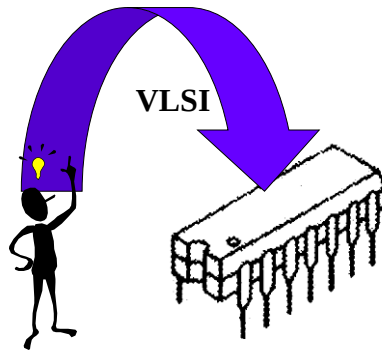
AES

Software open e free

JCE, JSSE



Algoritmi segreti e parametri segreti



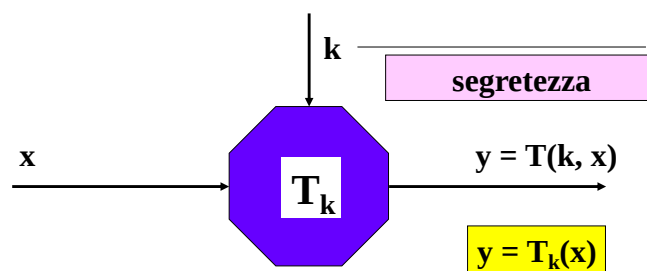
Clipper

GSM

Intro al Corso – Sicurezza dell'Informazione M



Segretezza di una trasformazione: algoritmo con chiave segreta!



insieme delle trasformazioni:
spazio delle chiavi:

$T = \{t_1, t_2, \dots, t_N\}$
 $K = \{k_1, k_2, \dots, k_N\}$

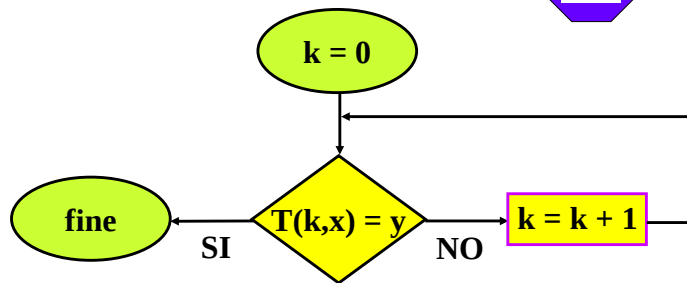
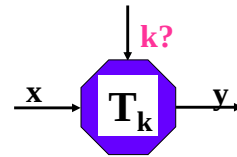
N grandissimo!



Attacco con forza bruta alla segretezza di una chiave

IPOTESI:

1. $0 \leq k \leq 2^n - 1$, $y = T(k, x)$
2. E' nota almeno una coppia x, y



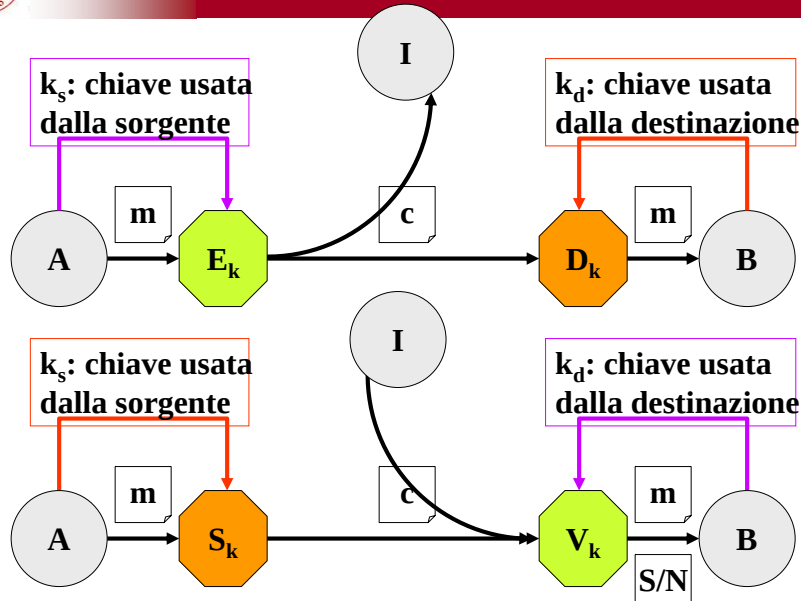
$O(\exp(n))$
 $n > 128$



Chiavi



Riservatezza e autenticazione



La relazioni tra le chiavi

$k_s \in K$
 $k_d \in K$

$K \rightarrow K : k_s = f(k_d).$

• **Algoritmo a chiavi simmetriche o simmetrico:**
 le chiavi k_s, k_d sono o *uguali* o *facilmente calcolabili* una dall'altra; la situazione usuale è
 $k_s = k_d$.

• **Algoritmo a chiavi asimmetriche o asimmetrico:**
 le chiavi k_s, k_d sono *diverse* ed una delle due è *difficilmente calcolabile* dall'altra

Autenticazione	Riservatezza
$k_d = f(k_s)$ facile!	$k_s = f(k_d)$ facile!
$k_s = f^{-1}(k_d)$ difficile!	$k_d = f^{-1}(k_s)$ difficile!

difficile!



La relazioni tra le chiavi

$ks \in K$
 $kd \in K$

$K \rightarrow K : ks = f(kd).$

Algoritmo a chiavi simmetriche o simmetriche
le chiavi ks , kd sono o *uguali* o *ks & kd segrete* *stabili*
una dall'altra; la situazione us
 $ks = kd.$

Algoritmo a chiavi asimmetriche o asimmetriche
 ks segreta
 kd pubblica
 kd sono *diverse* ed una *facile!* *difficile!*
ente *calcolabile* dall'altra
 $kd = f(ks)$ *facile!*
 $ks = f^{-1}(kd)$ *difficile!*
Riserva
 $ks = f(kd)$ *facile!*
 $kd = f^{-1}(ks)$

difficile!

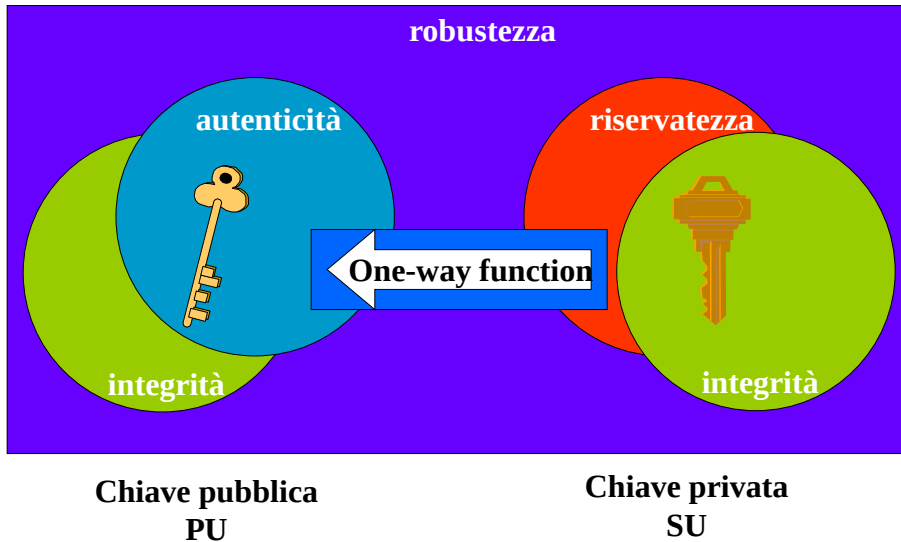


Proprietà delle chiavi simmetriche





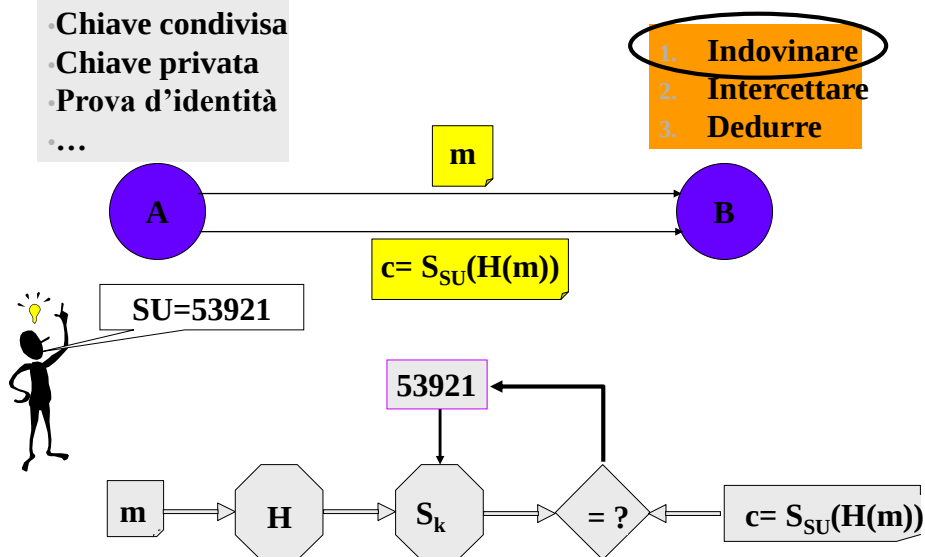
Proprietà delle chiavi asimmetriche



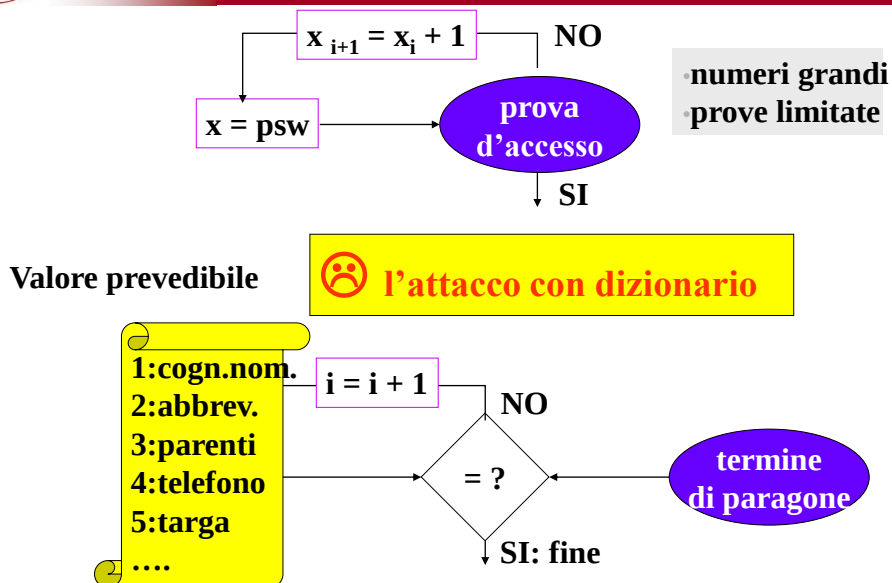
Crittanalisi



Dati segreti



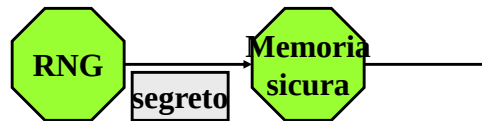
Attacco con forza bruta e con dizionario





Tirare ad indovinare

*I simboli della stringa che rappresenta un segreto devono essere
molti e scelti a caso*



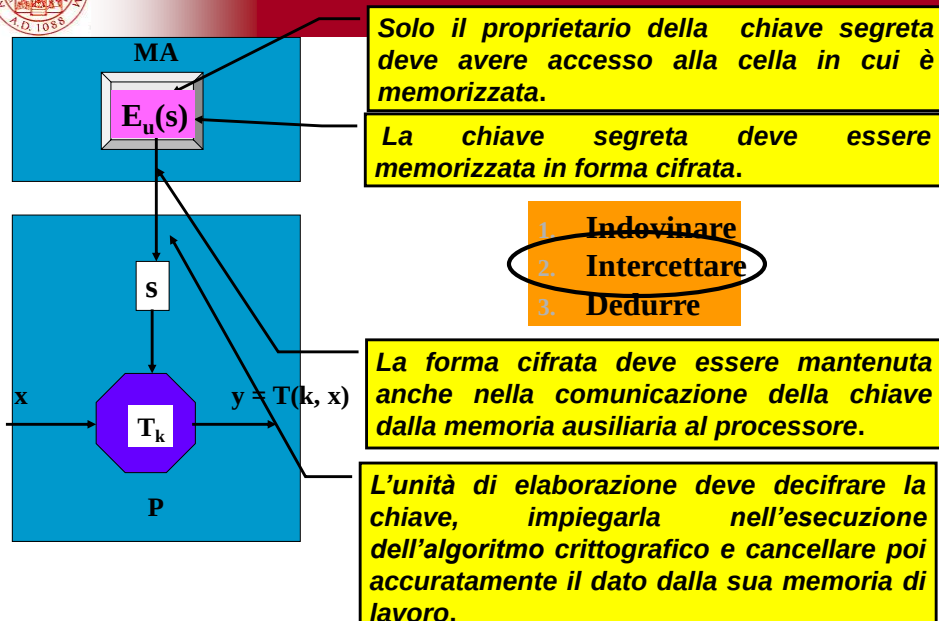
Valutazioni

- | | | |
|--------------------------|------------------|--|
| 1. Tirare a indovinare | 2^{-n} | $> 20 \text{ bit}$ |
| 2. Provare per k volte | $k \cdot 2^{-n}$ | $> 30 \text{ bit}$ |
| 3. Ricerca esauriente | $O(\exp(n))$ | $> 80 \text{ bit} \rightarrow 10^{11} \text{ anni MIPS}$ |

Un dato segreto deve essere frequentemente modificato

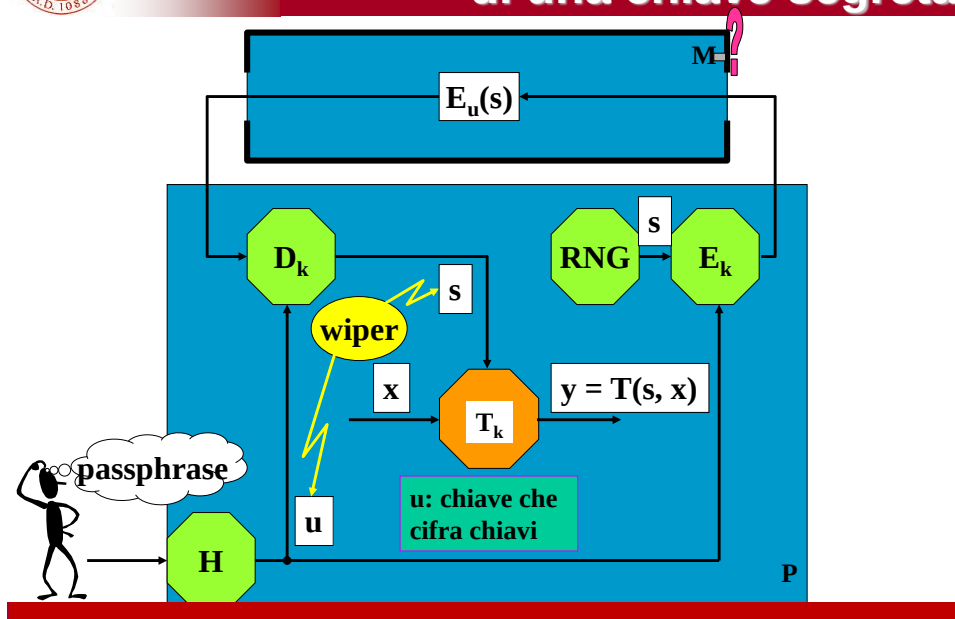


Memorizzazione ed uso di una chiave segreta

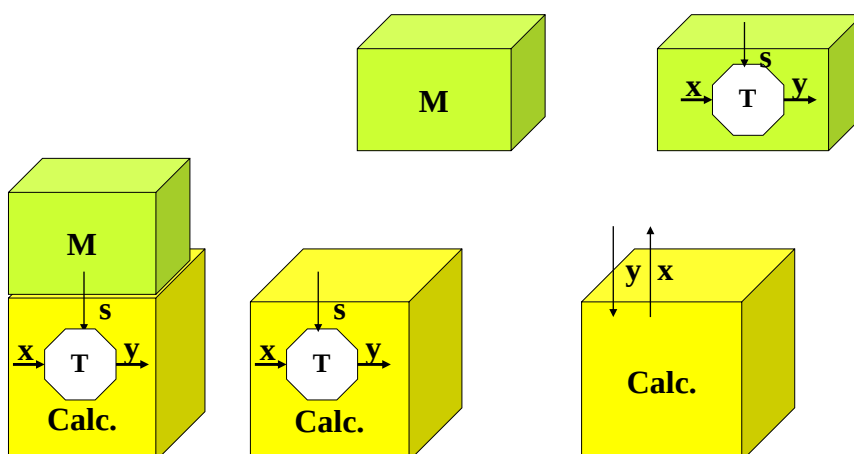


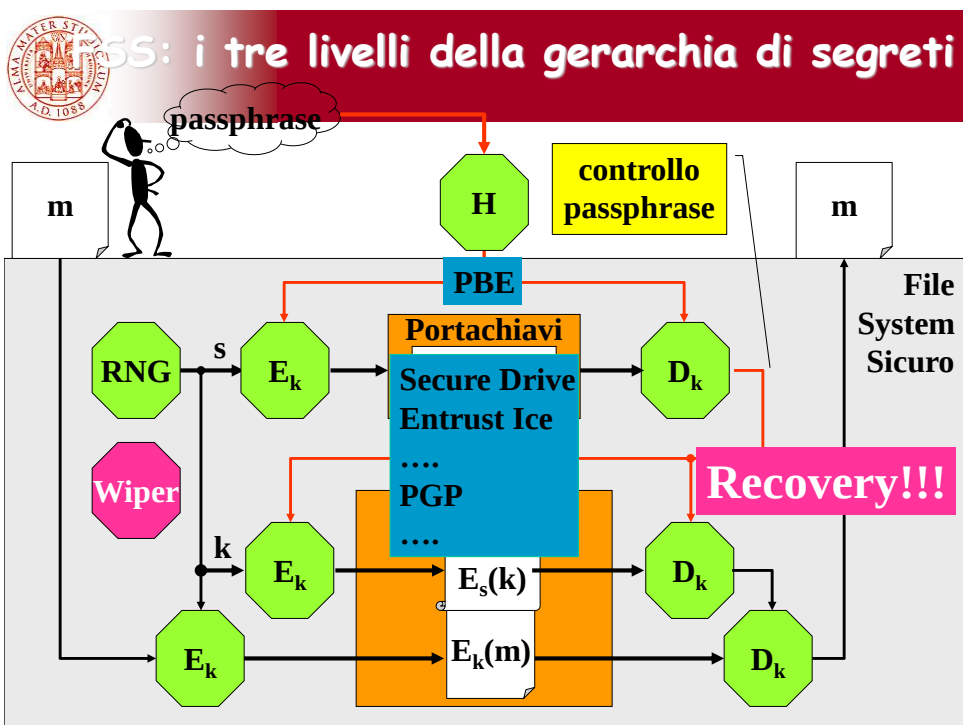
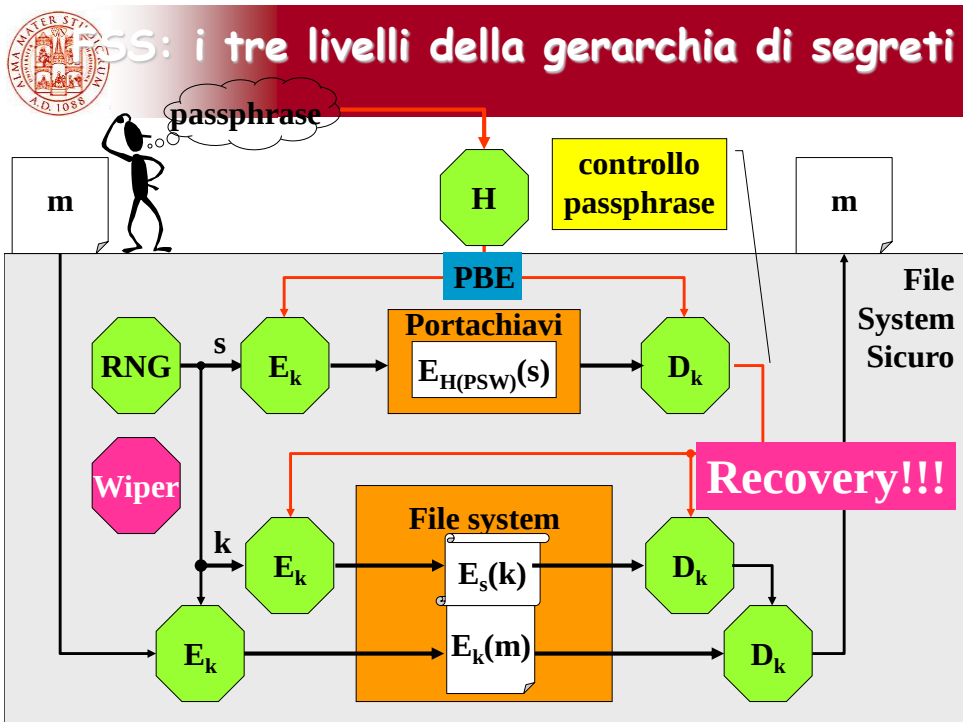


Generazione, memorizzazione ed uso di una chiave segreta



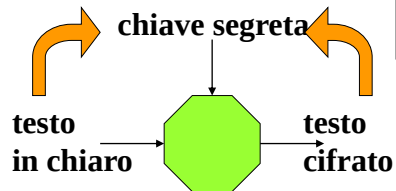
HardDisk < MemoryCard < SmartCard







Deduzione di un segreto dal suo uso



☹️ L'attacco con statistiche

1. Indovinare
2. Intercettare
3. **Dedurre**

ATTACCO	CONOSCENZE DELL'INTRUSO
con solo testo cifrato	linguaggio e probabilità d'occorrenza
con testo in chiaro noto	coppie di testo in chiaro e cifrato
con testo in chiaro scelto	testi cifrati di testi in chiaro scelti
con testo cifrato scelto	testi in chiaro di testi cifrati scelti



Contromisura preventiva

**L'uscita di un algoritmo crittografico
deve apparire come una variabile aleatoria
che assume con eguale probabilità
tutti i suoi possibili valori**



Complessità computazionale di un algoritmo

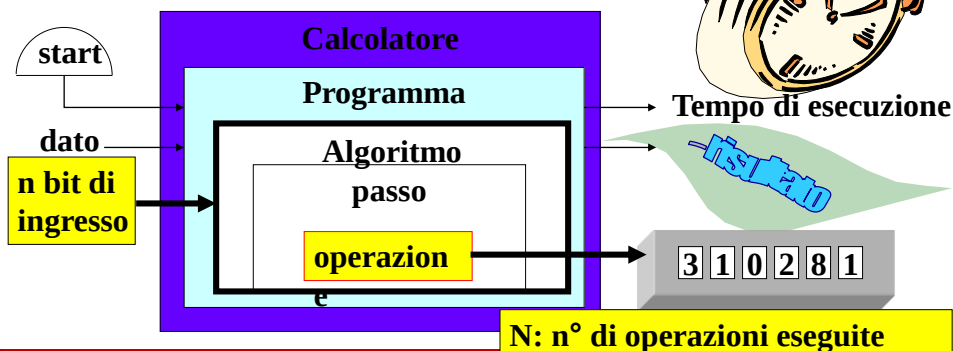


Teoria della complessità computazionale

Calcoli FACILI e Calcoli DIFFICILI

Indicatori di complessità:

- tempo di esecuzione,
- memoria occupata dal programma,
- memoria occupata dai dati





Teoria della Complessità di un Algoritmo

Tempo di esecuzione di un algoritmo: numero di operazioni N che occorre eseguire per terminarlo quando il dato d'ingresso è rappresentato da una stringa di n bit ($n = \log [\text{valore del dato}]$)

$$N = f(n)$$

In generale, a parità di n , si hanno diversi valori di N .

Tempo di esecuzione nel caso peggiore: numero massimo di operazioni

$N_{\max}$ che occorre eseguire per qualsiasi dato d'ingresso di n bit

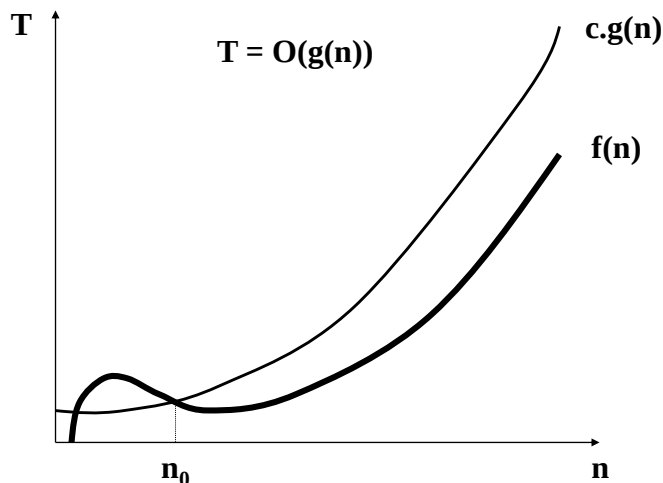
$$N_{\max} = f(n)$$

Si considera la modalità d'incremento di $N_{\max}$ al crescere senza limiti di n

Andamento asintotico del tempo di esecuzione nel caso peggiore detto Ordine di grandezza del tempo di esecuzione: $T = O(g(n))$
ove $g(n)$ è una funzione tale che $0 \leq f(n) \leq c \cdot g(n)$ per $n \geq n_0$ e c cost.



La notazione del “grande O”



Se è nota l'espressione di $f(n)$, si considera come $g(n)$ il termine di $f(n)$ che cresce più rapidamente con n



Classificazioni

Classificazione degli algoritmi

1. con **tempo polinomiale**:

$T = O(n^t)$ con t esponente più grande in $g(n)$,

2. con **tempo esponenziale**:

$T = O(b^n)$, con b costante, o anche $T = O(\exp(n))$

Classificazione dei problemi

1. **facile**, se esiste un algoritmo polinomiale in grado di risolverlo su una macchina di Turing deterministica,

2. **difficile**, se non sono stati fino ad ora individuati (e probabilmente non saranno mai individuati) algoritmi che lo risolvono in tempo polinomiale



Complessità e Sicurezza

- Caso peggiore e istanze facili

- Modalità di incremento e valori di n

R10: “ogni algoritmo che consente di difendere una proprietà critica dell’informazione deve avere tempo polinomiale”

ESEMPI: $O(1)$, $O(n)$, $O(n^3)$

R11: “ogni algoritmo che consente di violare una proprietà critica della informazione deve avere tempo esponenziale”

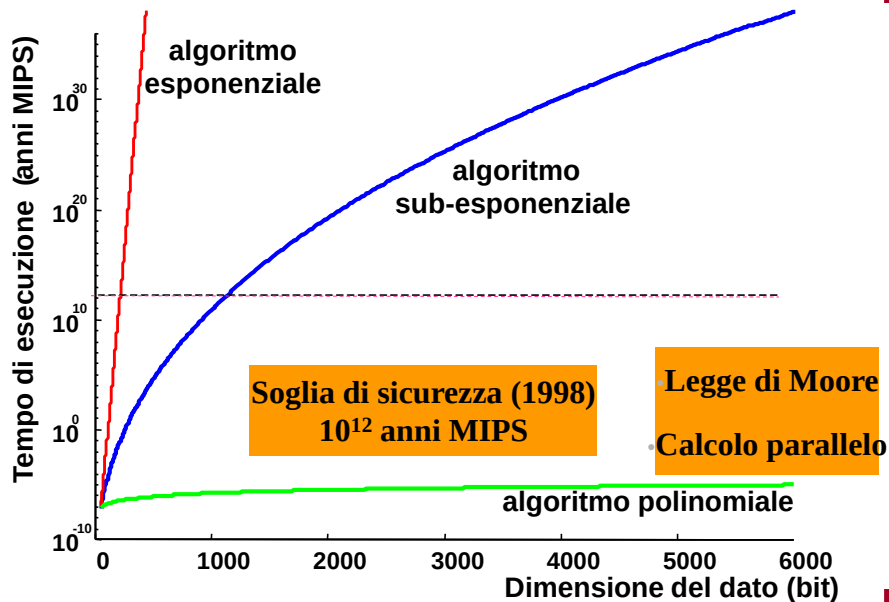
ESEMPI: $O(\exp(n))$, $O(\exp(n)^{1/2})$

- Algoritmi sub-esponenziale: $O(\exp((n)^\alpha (\ln(n)^{1-\alpha})))$ con $0 < \alpha < 1$

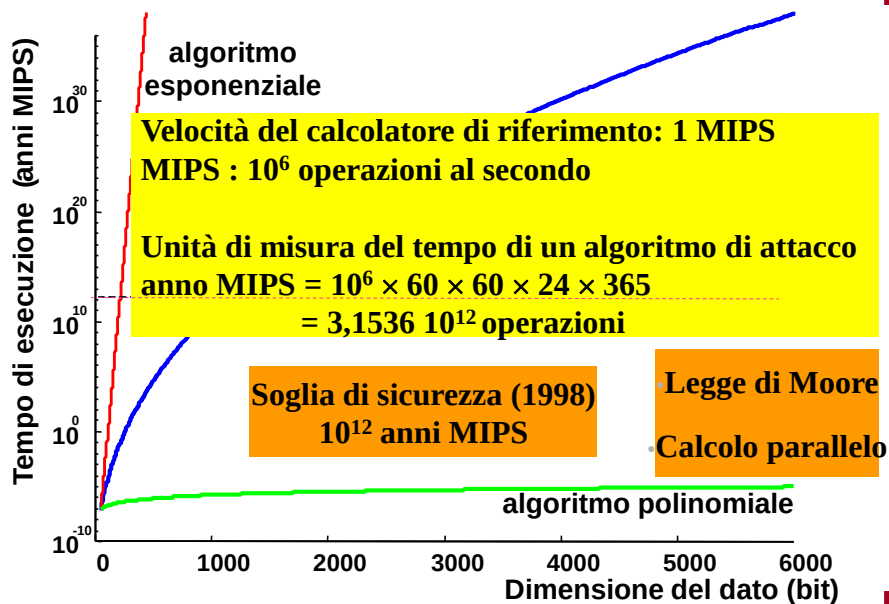




Anni MIPS e dimensione del dato



Anni MIPS e dimensione del dato



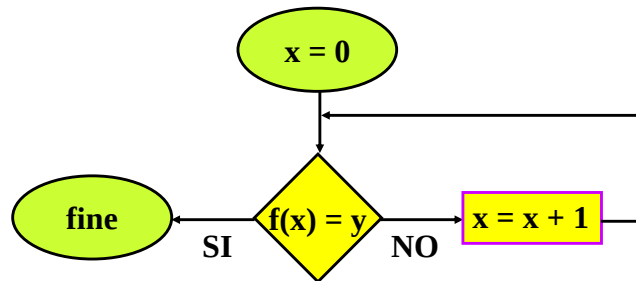


Livello di sicurezza

1-L'algoritmo di ricerca esauriente risolve ogni problema difficile

PROBLEMA: inversione di una funzione

ES. Per quale intero x , con $0 < x < 2^n - 1$, vale la proprietà $f(x) = y$?



Nel caso peggiore occorrono 2^n passi: $O(\exp(n))$



Livello di sicurezza

2-La complessità di ogni algoritmo di attacco può essere misurata facendo riferimento all'attacco con forza bruta

Algoritmo A: N passi per terminare nel caso peggiore

Si individua un n tale che $2^n \leq N < 2^{n+1}$

in altre parole si confronta la complessità di A con quella dell'algoritmo di ricerca esauriente

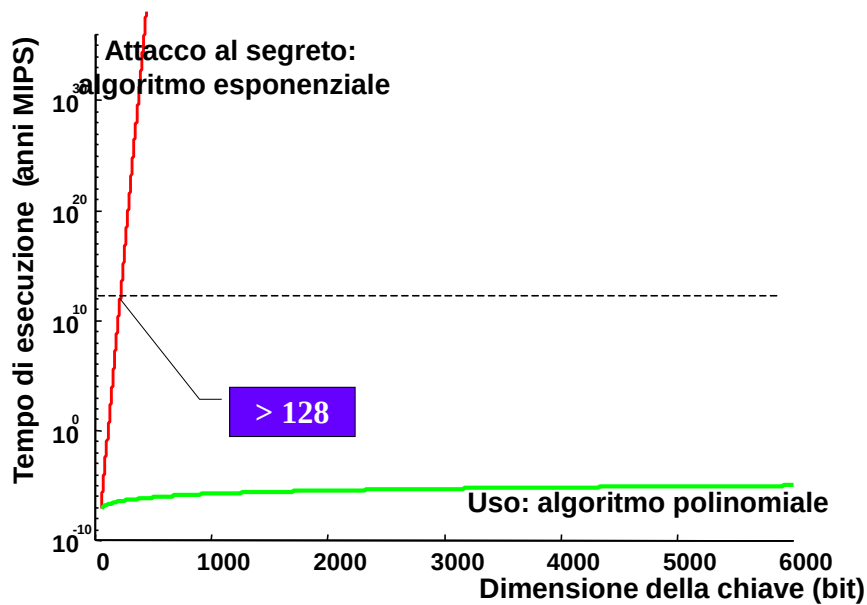
Si assegna all'algoritmo A un livello di sicurezza di n bit

Misura di robustezza indipendente dal calcolatore e dalle operazioni

**10^{12} anni MIPS equivalgono ad un livello di sicurezza di 85 bit.
Attualmente si cerca di conseguire un livello di 128 bit**



Dimensionamento di una chiave simmetrica



Dimensionamento di una chiave asimmetrica

