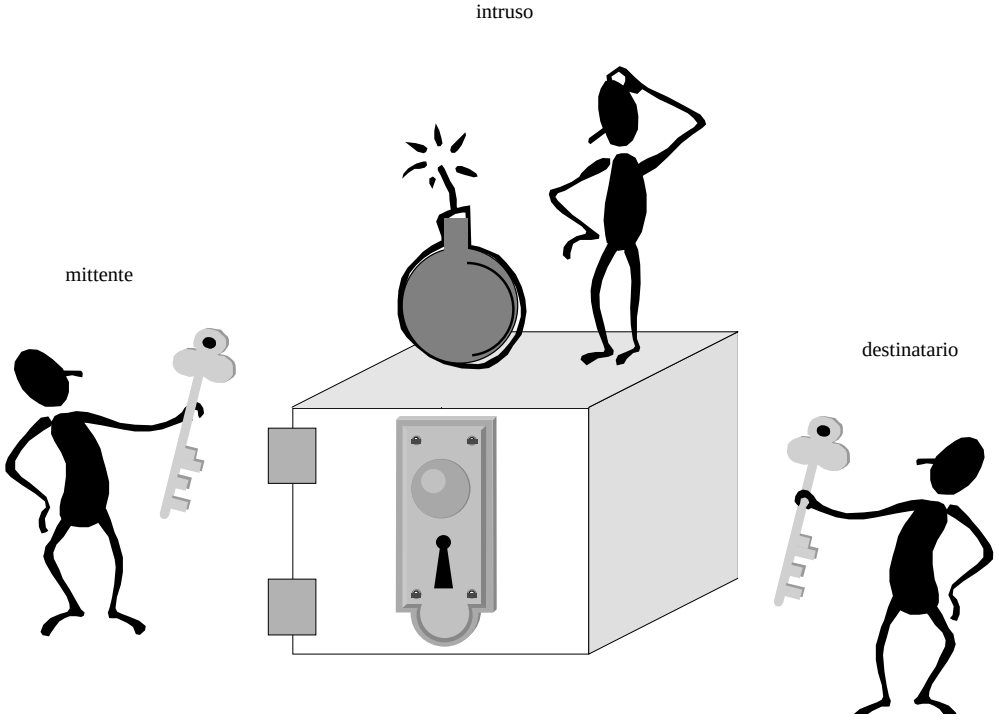
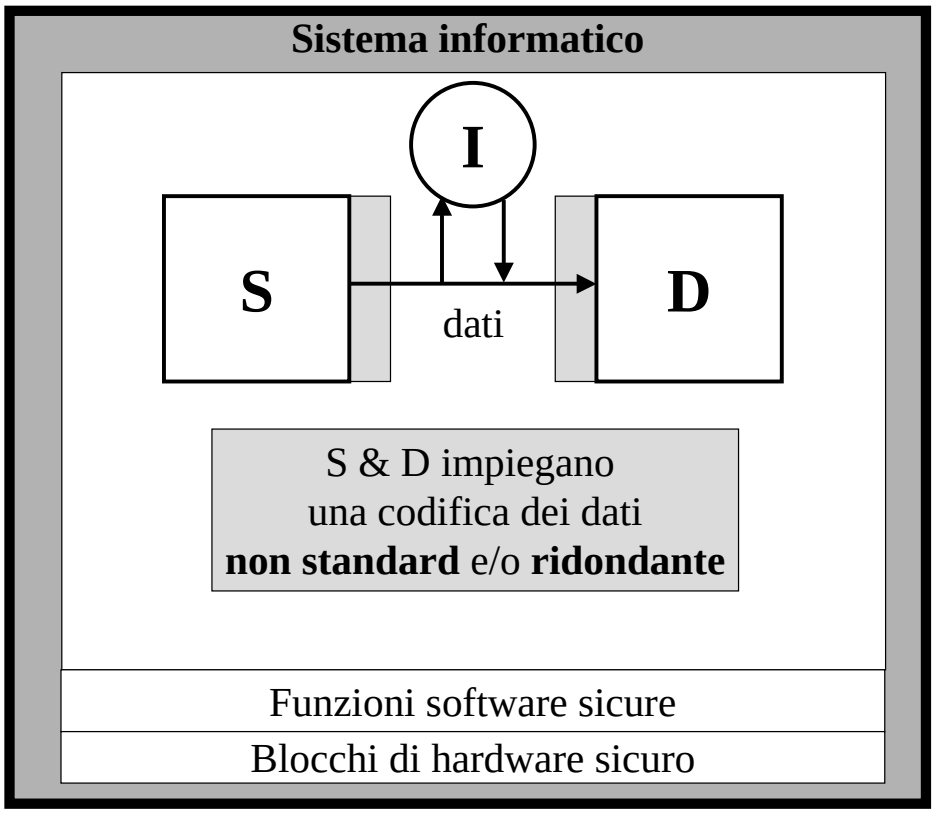
Dati sicuri

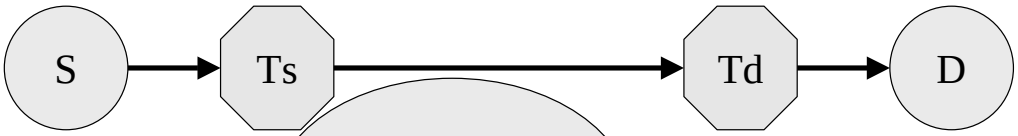


Sicurezza dei dati

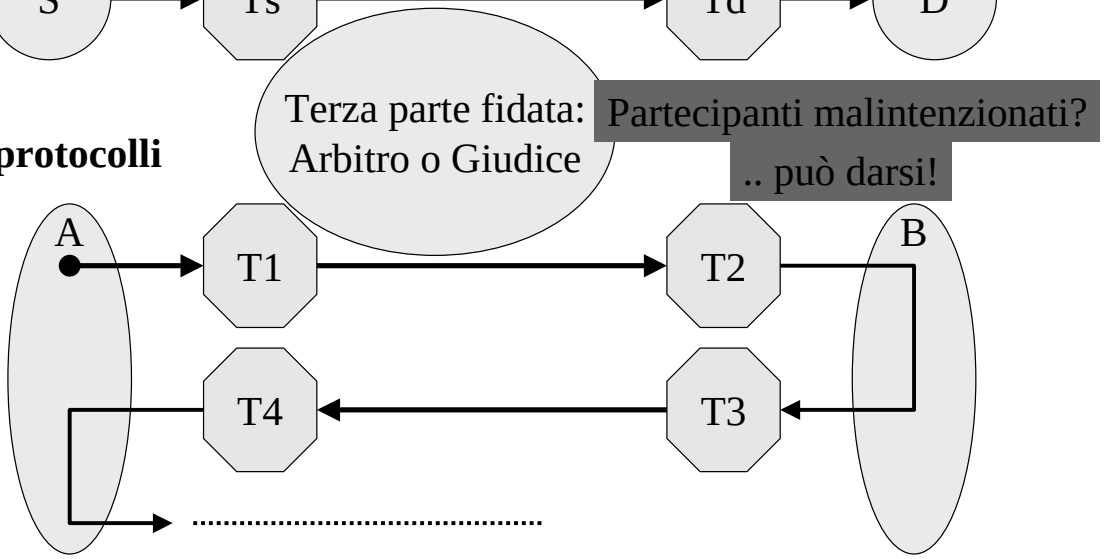


Trasformazioni per la Sicurezza

1: algoritmi

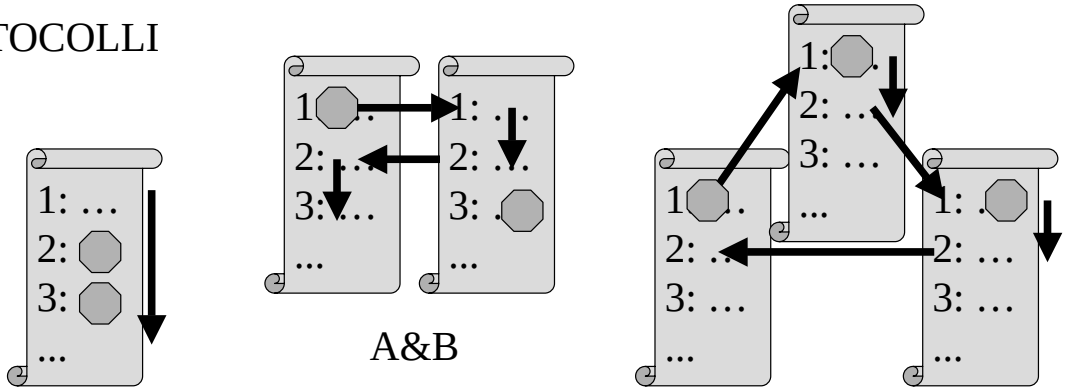


2: protocolli

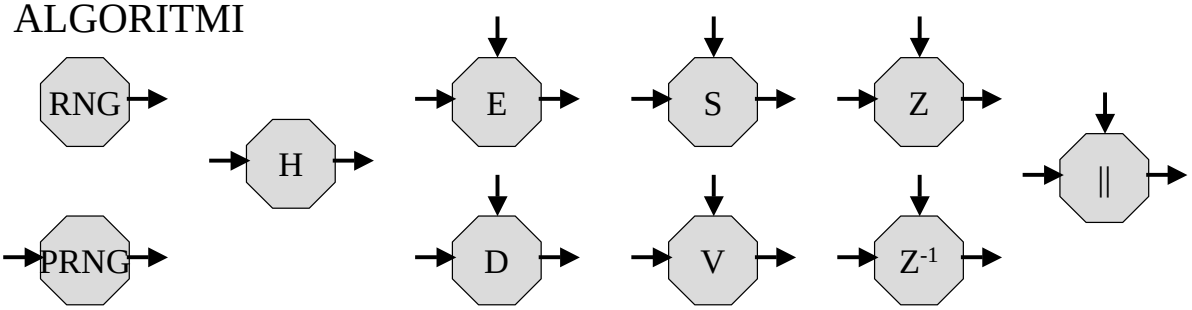


Meccanismi

PROTOCOLLI

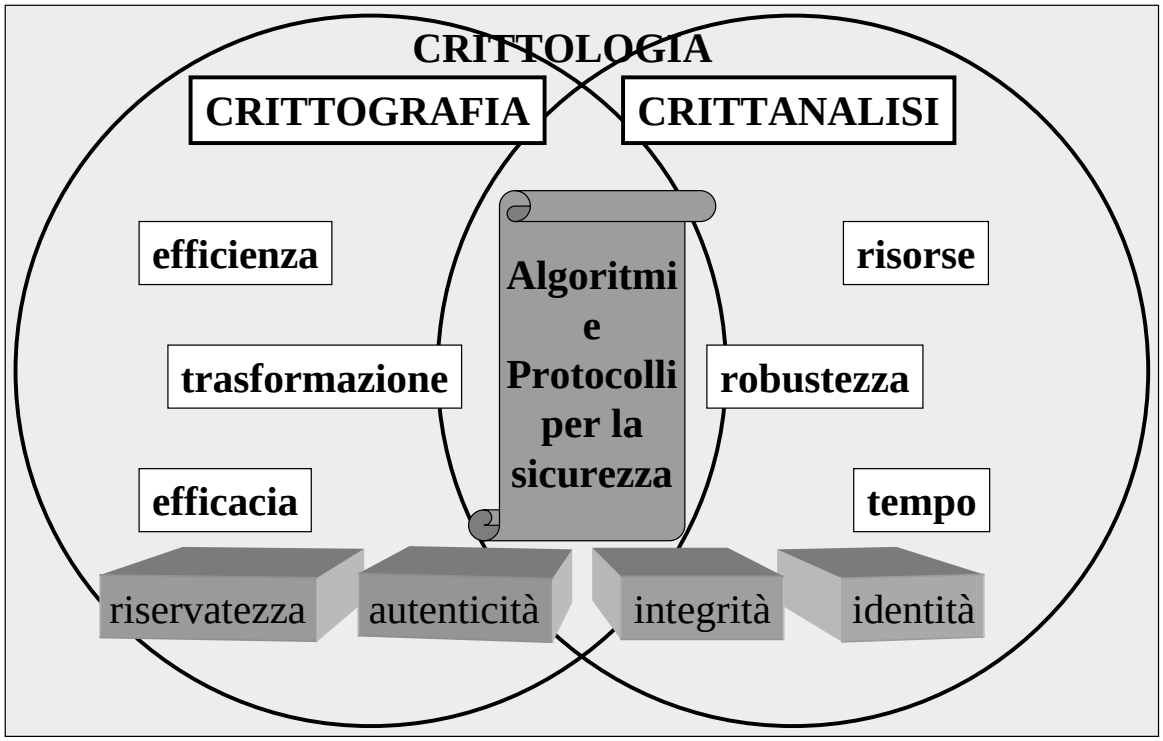


ALGORITMI



2.1 Che cos'è la Crittografia?
2.2 Che cos'è la Crittanalisi?
2.3 Quali sono i principi di difesa?

Crittografia e Crittanalisi



1940-44

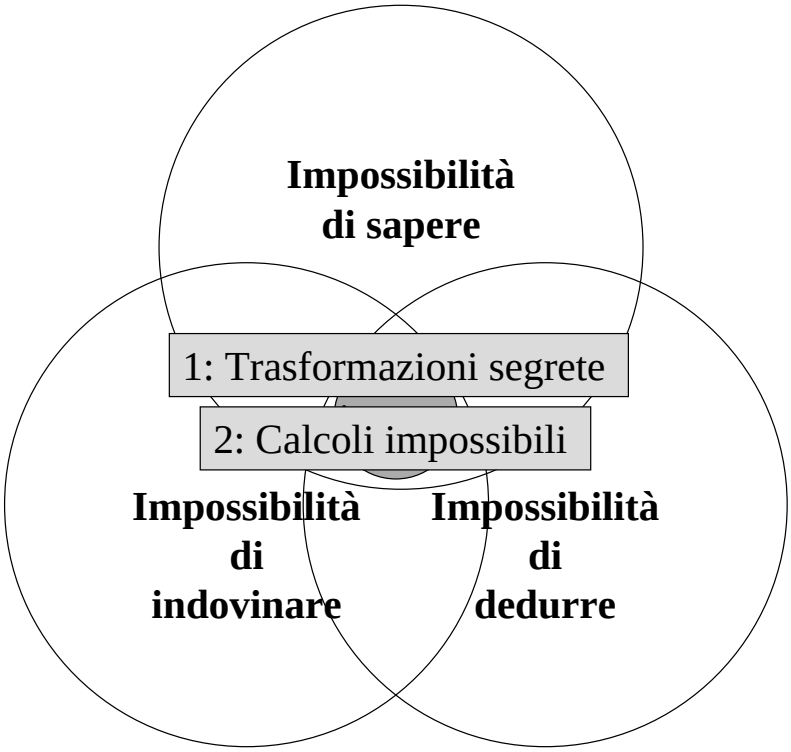
Algoritmi complessi

Crittografia classica

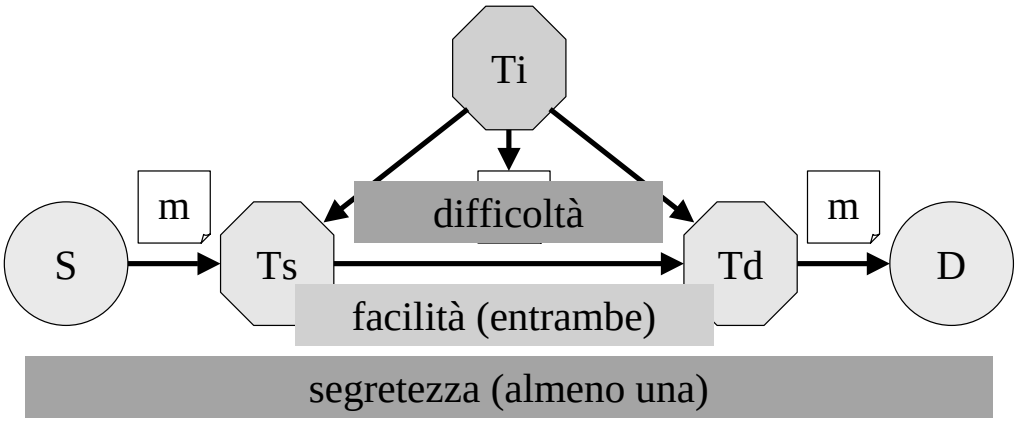
Crittografia moderna

Algoritmi semplici

I Principi della Difesa



Trasformazioni per la Sicurezza



Tutti gli algoritmi corrispondenti ad azioni lecite devono essere facili

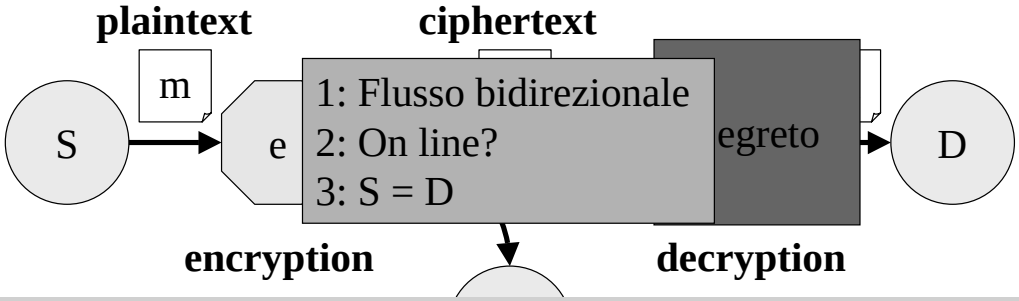
Tutti gli algoritmi corrispondenti ad azioni illecite devono essere difficili

Trasformazioni per la sicurezza

- **riservatezza**

Elaborazioni per la Riservatezza di un Messaggio

R2:"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".



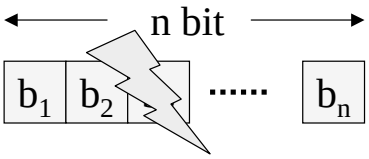
Sicurezza perfetta: da c non si impara nulla di più di quello che si sapeva già

❑R3: "i calcoli per mettere in chiaro un testo cifrato senza conoscere l'algoritmo di decifrazione devono essere difficili"

Trasformazioni per la sicurezza

- **integrità**

Minacce all'integrità



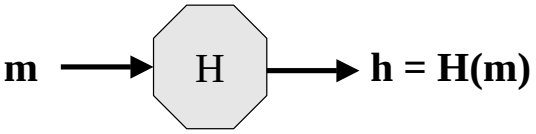
- Inserzione
- Cancellazione
- Spostamento
- Inversione

di uno o più bit

Integrità: rilevazione di attacchi intenzionali

❑R6: "la sorgente deve affiancare al documento un "riassunto" che ne rappresenti in modo univoco il contenuto e l'origine; la destinazione deve poter verificare l'autenticità e la congruenza del riassunto"

Funzione hash crittografica
•facile da eseguire
•risposta imprevedibile

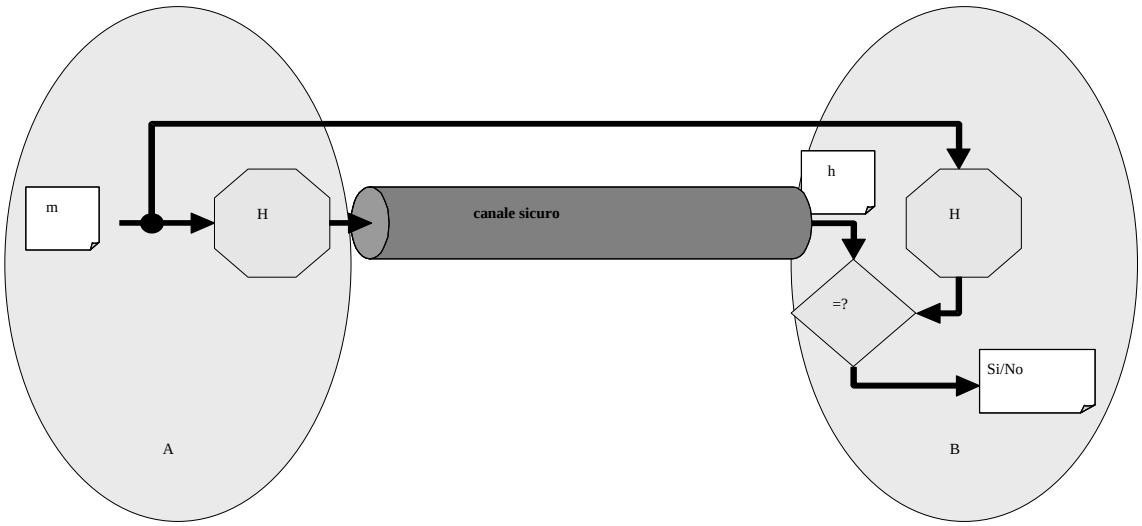


Stringa di lunghezza fissa

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

❑R7: "i calcoli per costruire due messaggi con la stessa impronta devono essere difficili da eseguire"

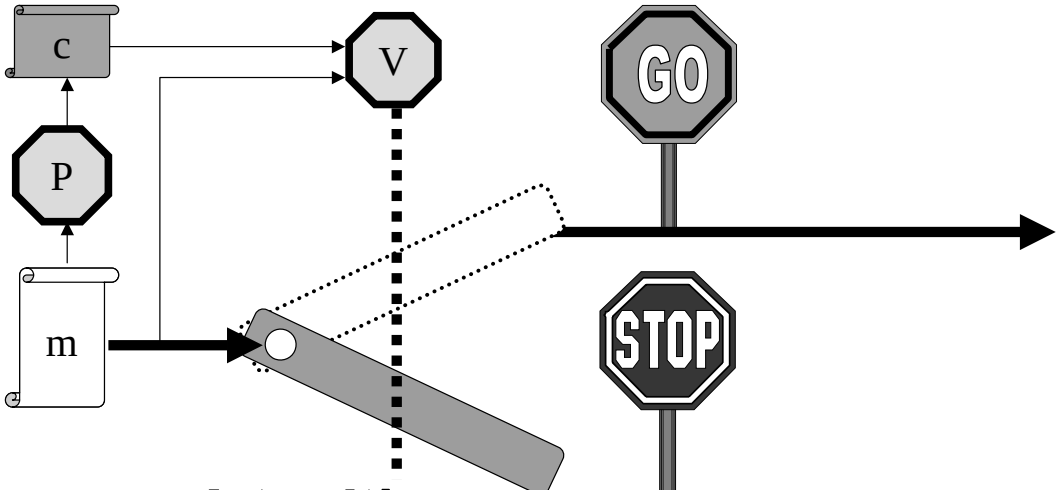
Accertamento dell'integrità



Trasformazioni per la sicurezza

- autenticità

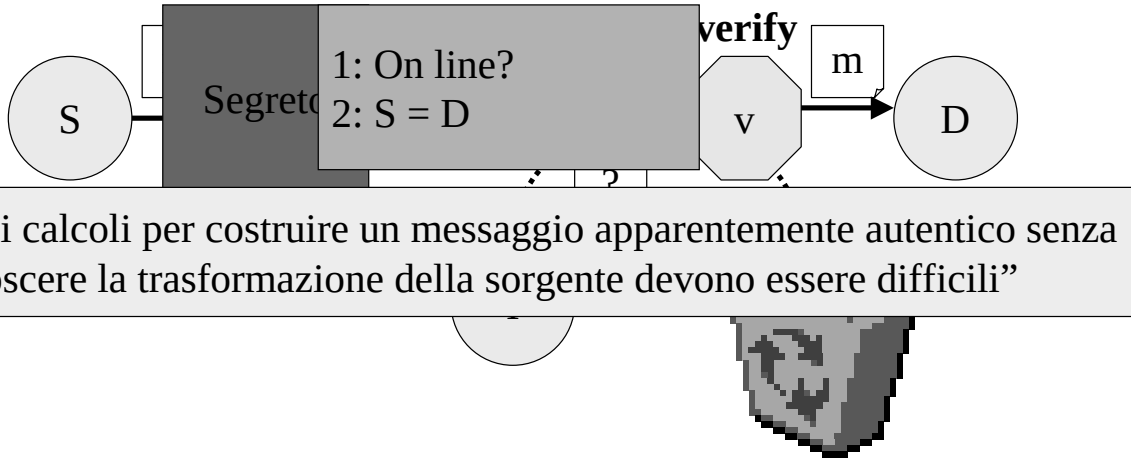
Attacchi attivi: rilevazione e reazione



Integrità
Origine
Tempestività
Ordine

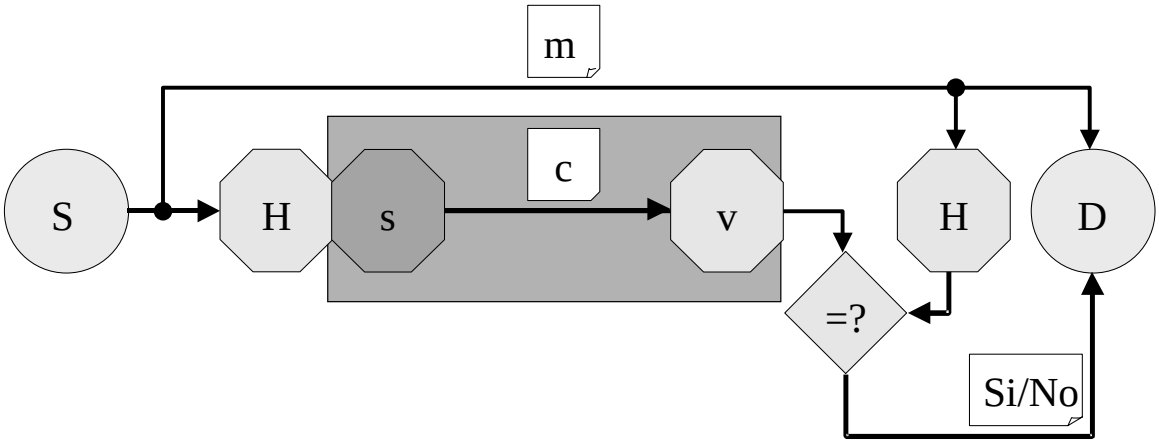
Elaborazioni per l'autenticazione di un messaggio

R4:"la sorgente trasforma il documento, aggiungendogli informazioni atte ad attestare come è fatta la sua rappresentazione originaria e chi l'ha predisposta; la destinazione esegue una trasformazione che ripristina la forma originaria del documento dopo averne verificato gli attestati ".



R5:“i calcoli per costruire un messaggio apparentemente autentico senza conoscere la trasformazione della sorgente devono essere difficili”

Attestazione e verifica dell'integrità .. e dell'origine



a) canale sicuro

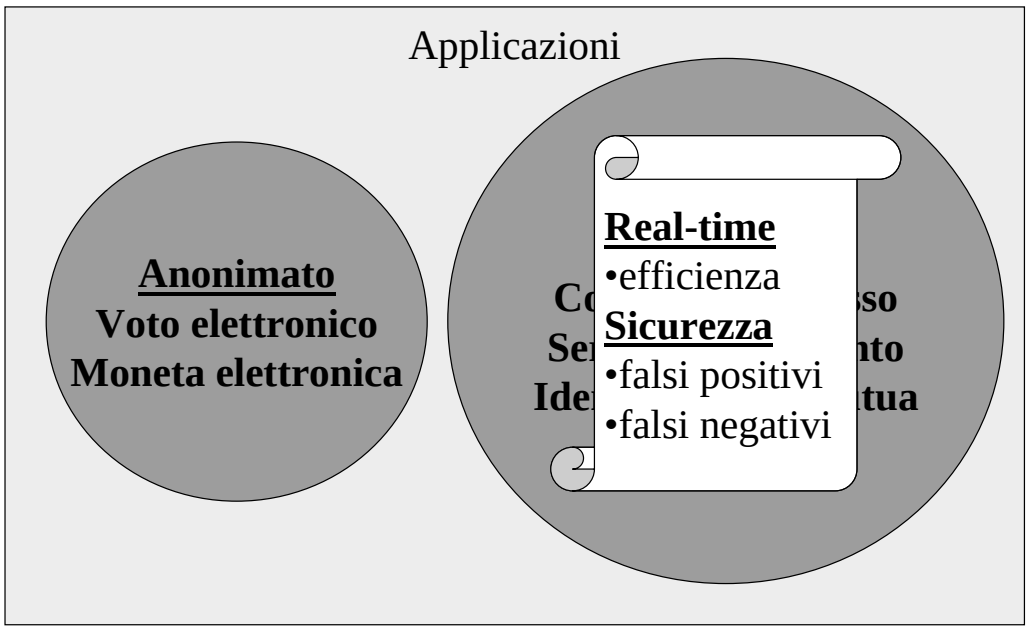
b) canale reso sicuro

Schema di firma digitale (RSA)

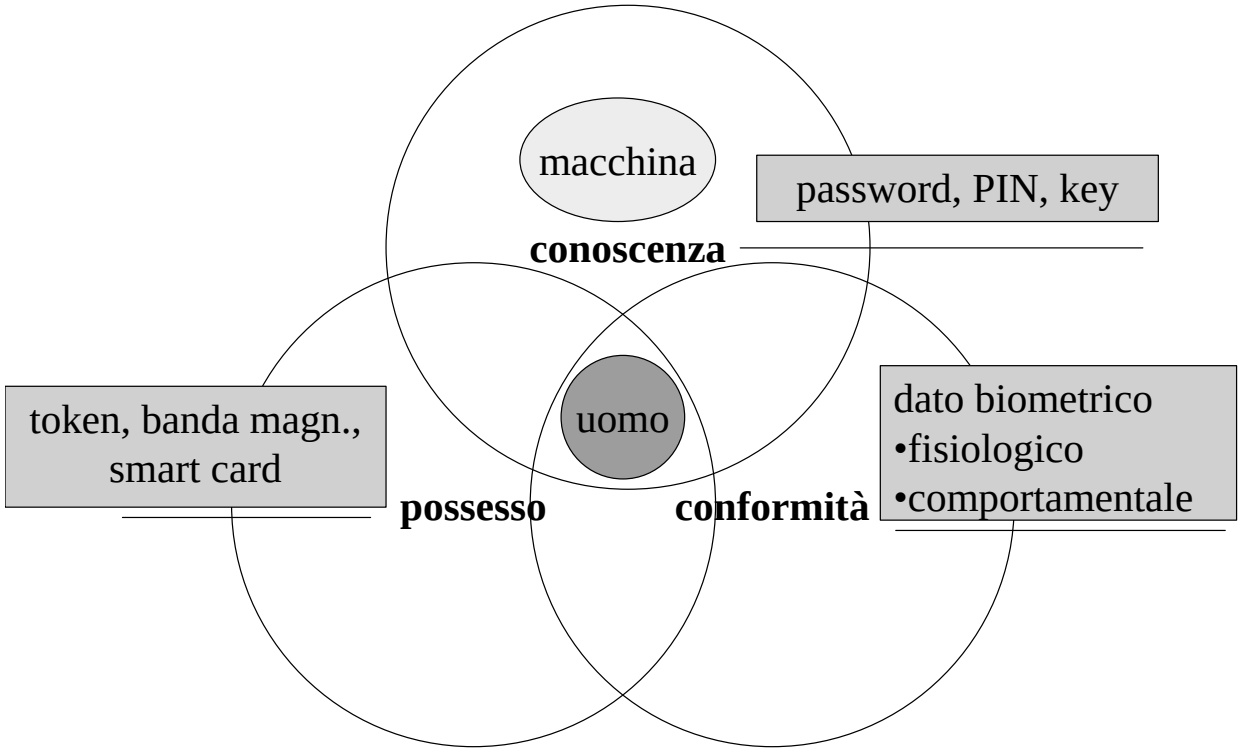
Trasformazioni per la sicurezza

- **identità**

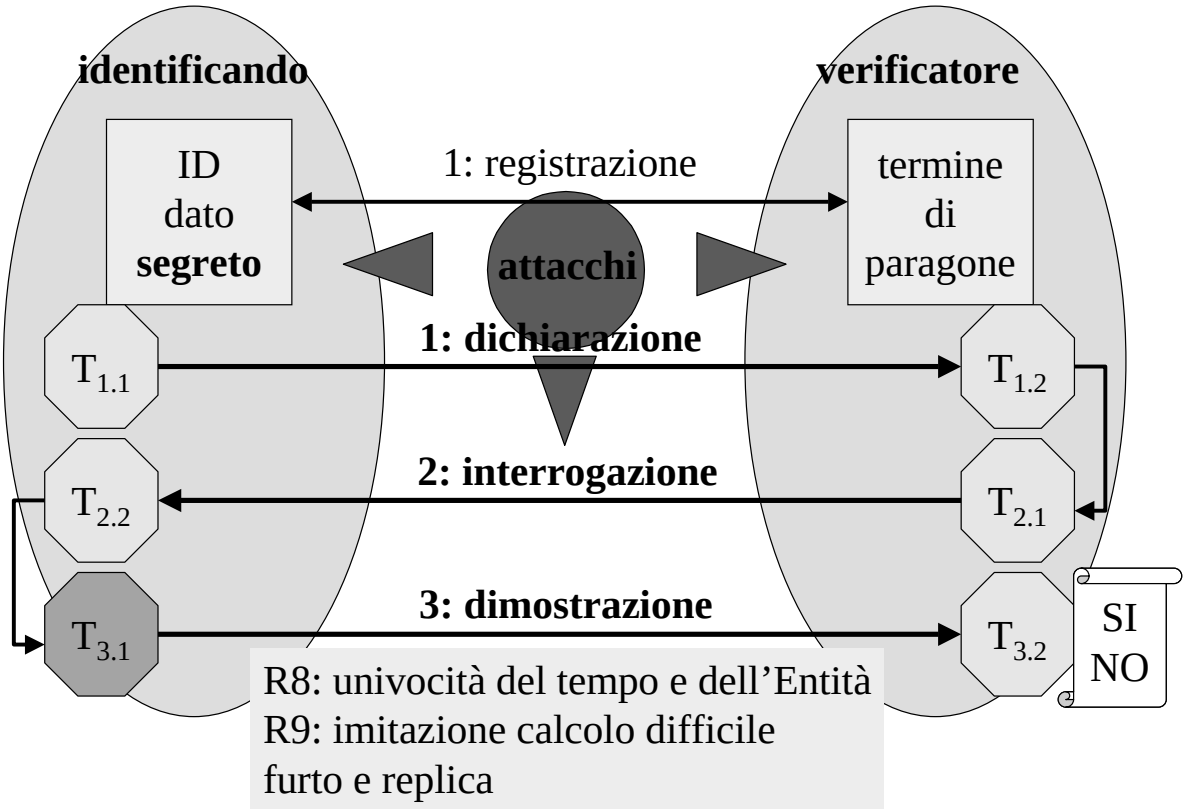
Anonimato/Identificazione



Strumenti di identificazione



Il protocollo d'identificazione



R8: univocità del tempo e dell'Entità
R9: imitazione calcolo difficile
furto e replica

Dimostrazione di conoscenza

- **passiva (o debole)**
  A→B: segreto
  - psw
- **attiva (o forte)**
  A→B: F^i(segreto)
  - one-time psw
  - challenge/response
  - zero knowledge

identificazione	unilaterale	reciproca
passiva	SI	NO
attiva	SI	SI

Trasformazioni per la sicurezza

- **complessità computazionale**

Calcoli **FACILI** e Calcoli **DIFFICILI**

Teoria della Complessità computazionale

Indicatori di complessità:

- tempo di esecuzione,
- memoria occupata dal programma,
- stack, ecc.

Misura 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 [valore del dato])

$$\mathbf{N = f(n)}$$

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

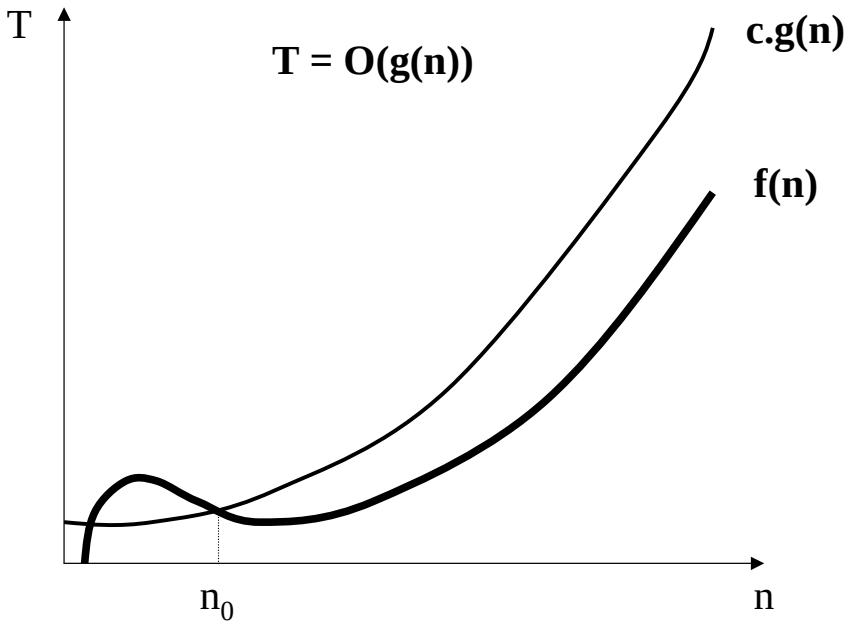
Tempo di esecuzione nel caso peggiore: numero massimo di operazioni $\mathbf{N_{max}}$ che occorre eseguire per qualsiasi dato d'ingresso di **n** bit

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

Si considera la modalità d'incremento di $\mathbf{N_{max}}$ *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 $\mathbf{0 \leq f(n) \leq c.g(n)}$ 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. **tipo P** o **facile** se esiste un algoritmo polinomiale in grado di risolverlo su una macchina di Turing deterministica,
2. **tipo NP** se è possibile, con una macchina di T. non deterministica, verificare in tempo polinomiale che una congettura è una soluzione.

N.B. I problemi P sono anche NP, ma non è vero il contrario.

2.1 **difficile (NP hard),** se non sono stati fino ad ora individuati algoritmi che lo risolvono in tempo polinomiale

2.2 **NP-completo,** se può essere trasformato (o ridotto) in ogni altro problema NP in tempo polinomiale.

Complessità e Sicurezza

• Caso peggiore e istanze facili

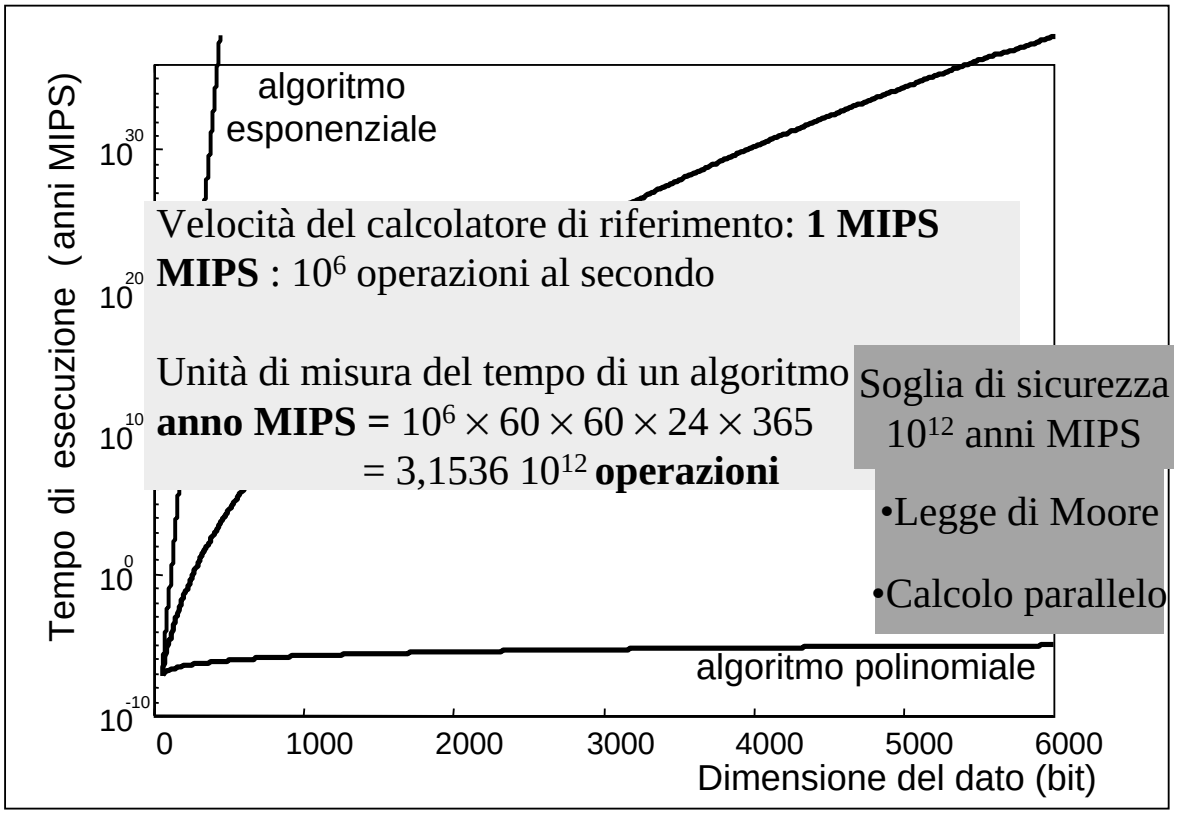
• Modalità di incremento e valori di n

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 anche $O(\exp(n)^{1/2})$

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

Tempo di esecuzione e dimensione del dato



Velocità del calcolatore di riferimento: **1 MIPS**
MIPS : 10^6 operazioni al secondo

Unità di misura del tempo di un algoritmo
anno MIPS = $10^6 \times 60 \times 60 \times 24 \times 365$
= 3,1536 10^{12} **operazioni**

Soglia di sicurezza
10^{12} anni MIPS

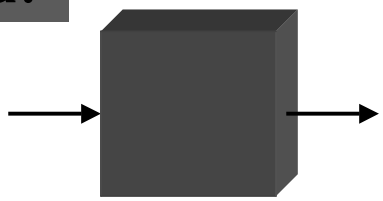
- Legge di Moore
- Calcolo parallelo

Primitive per le
Trasformazioni per la sicurezza

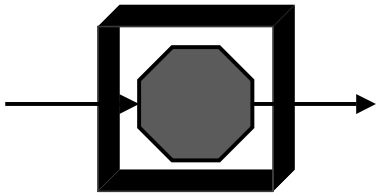
Trasformazioni segrete

Decifrazione Autenticazione
Segretezza si, ma di che cosa?

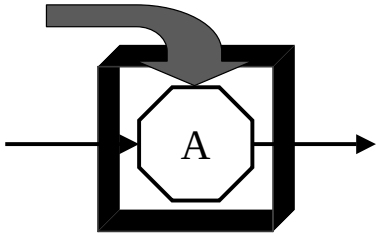
- **Macchina**



- **Algoritmo**



- **Parametro**



Vulnerabilità
delle macchine e degli algoritmi segreti

☹

- Ispezione
- Installazione
- Progetto
- Produzione
- Certificazione

Algoritmo pubblico e parametro segreto

Kerckhoffs : “La criptogràphie militaire”1883

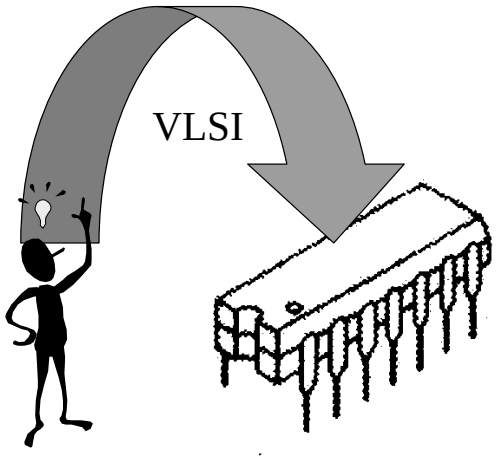
☺

Responsabilità dell’utente — cassaforte

Valutazione pubblica — AES

Software open e free — Crypto++, JCE, JSSE

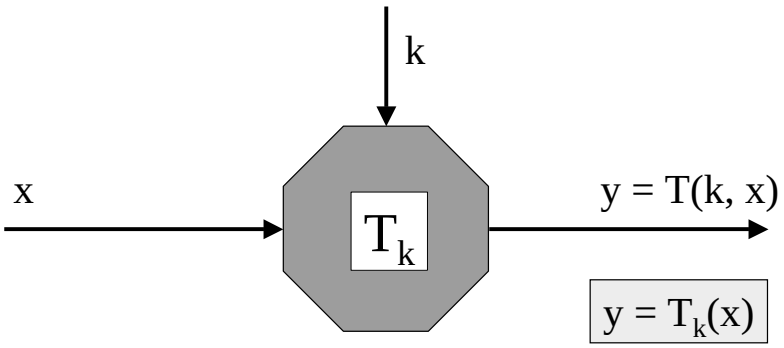
Algoritmi segreti e parametri segreti



Clipper

GSM

Algoritmo con chiave



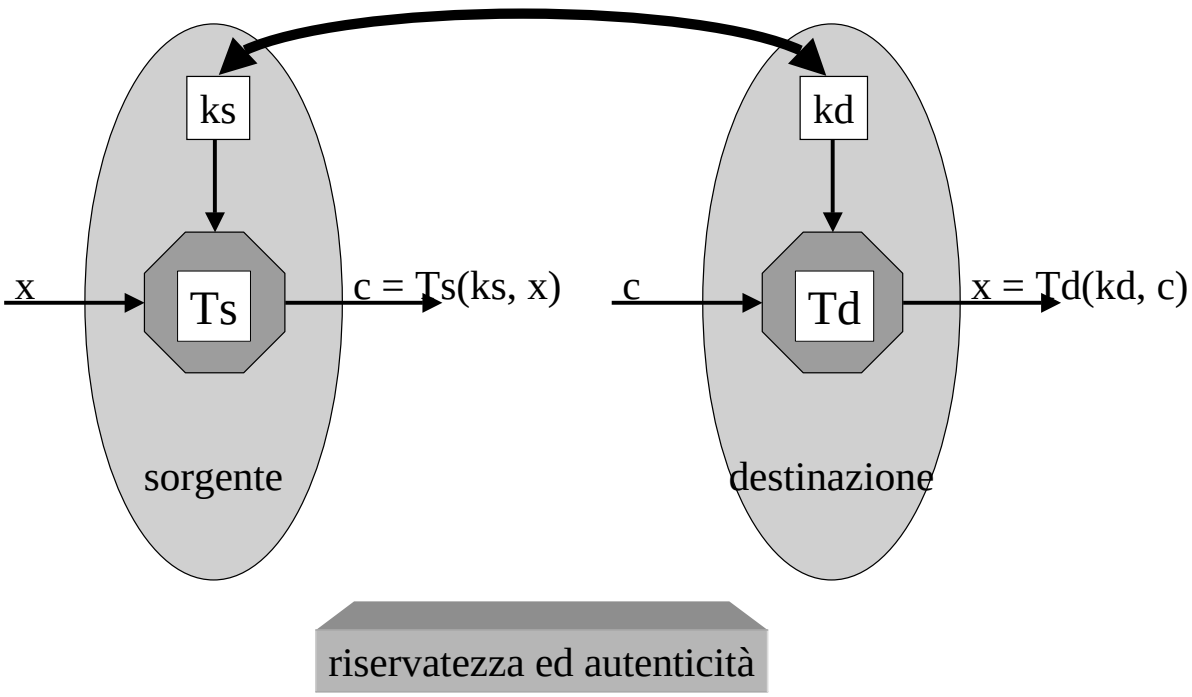
insieme delle trasformazioni: $T = \{t_1, t_2, \ldots, t_N\}$
spazio delle chiavi: $K = \{k_1, k_2, \ldots, k_N\}$

N grandissimo!

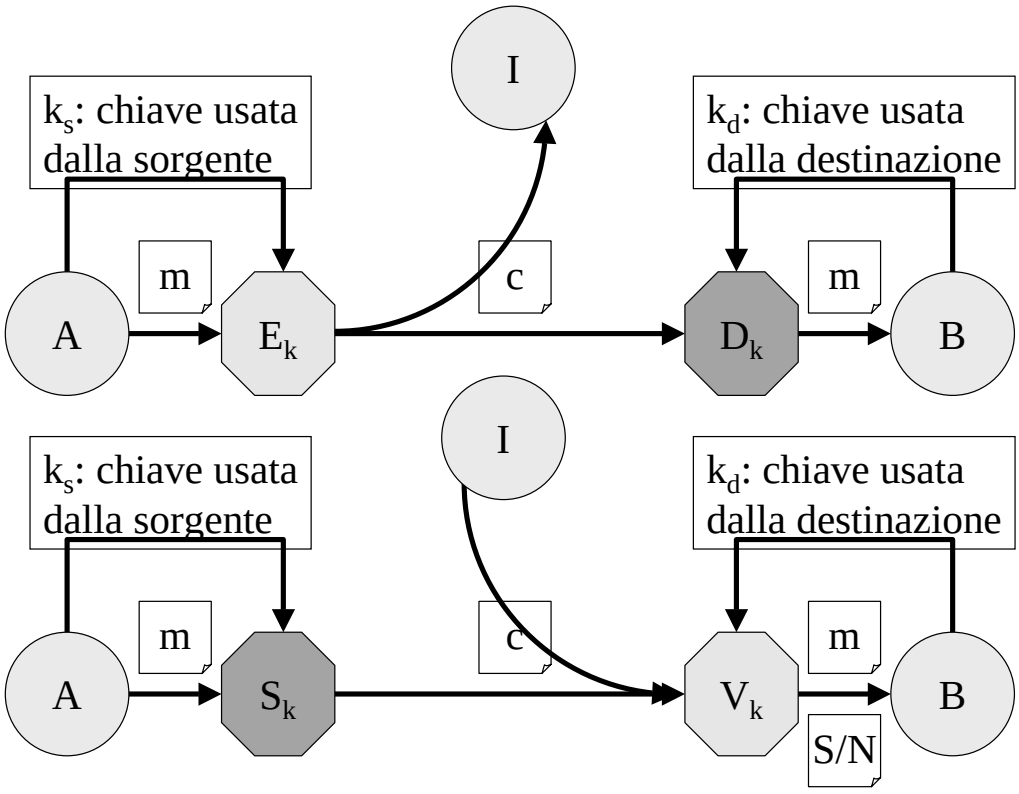
Trasformazioni per la sicurezza

- **chiavi simmetriche**
- **chiavi asimmetriche**

La coppia di algoritmi



riservatezza e autenticazione



La relazioni tra le chiavi

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

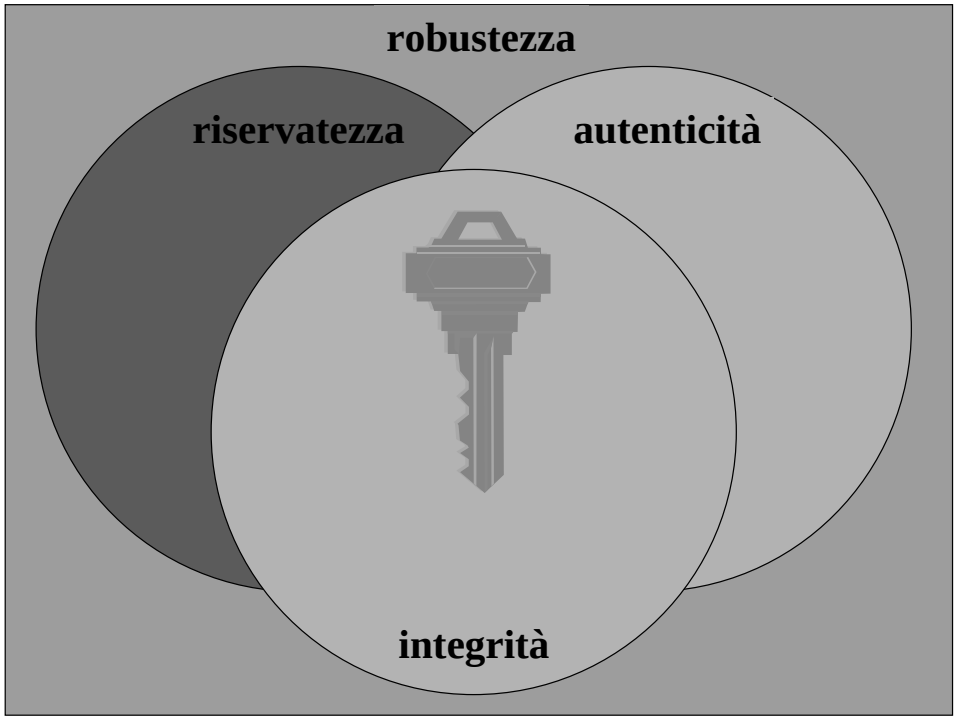
$\mathbf{K \rightarrow K : ks = f(kd)}$.

- **Algoritmo a chiavi simmetriche** o **simmetrico:**
  le chiavi **ks**, **kd** sono o *uguali* o *facilmente calcolabili* una dall'altra; la situazione usuale è
  ks = kd.

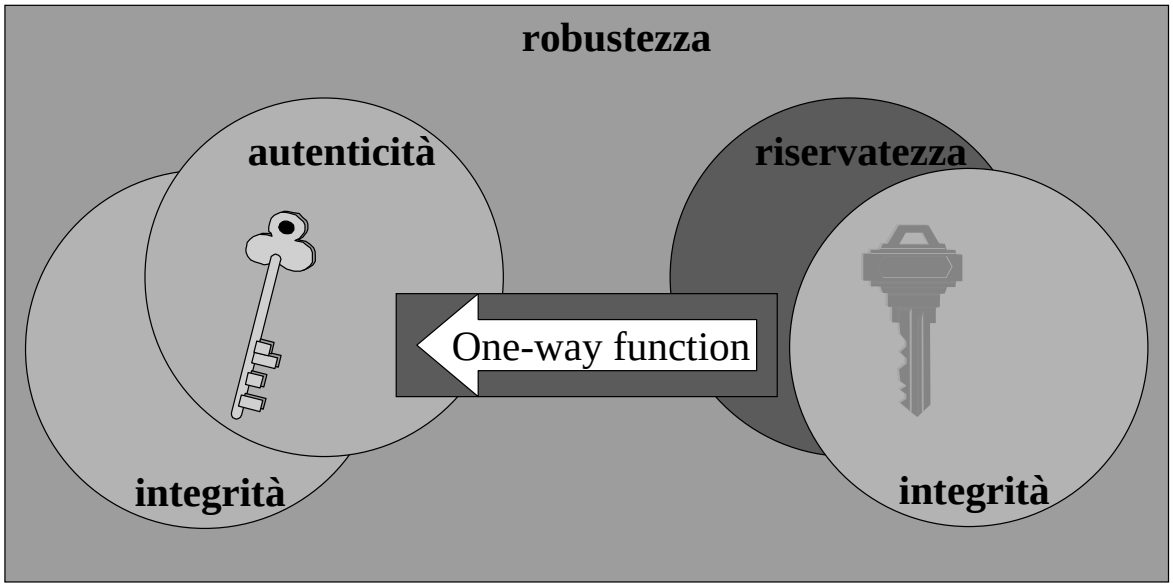
- **Algoritmo a chiavi asimmetriche** o **asimmetrico:**
  le chiavi **ks**, **kd** sono *diverse* ed una delle due è *difficilmente calcolabile* dall'altra

Autenticazione	Riservatezza
kd = f(ks) facile!	**ks = f(kd)** facile!
ks = f^{-1}(kd) difficile!	**kd = f^{-1}(ks)** difficile!

Proprietà delle chiavi simmetriche



Proprietà delle chiavi asimmetriche



Chiave pubblica
PU

Chiave privata
SU

Funzione unidirezionale (one way)

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)**.

Elenco telefonico

N° di X?
Ricerca binaria: O(n)

X di N°?
Ricerca esauriente: $O(2^n)$

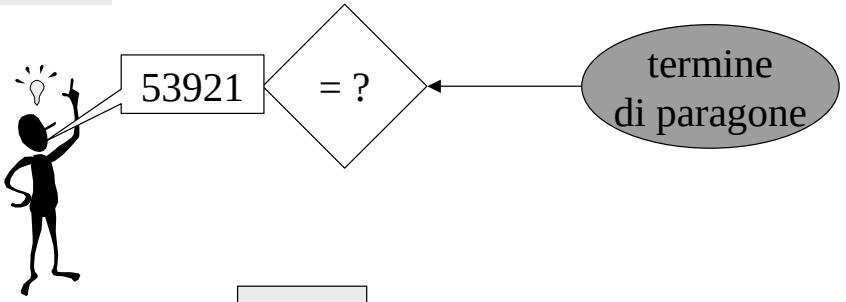
Trasformazioni per la sicurezza

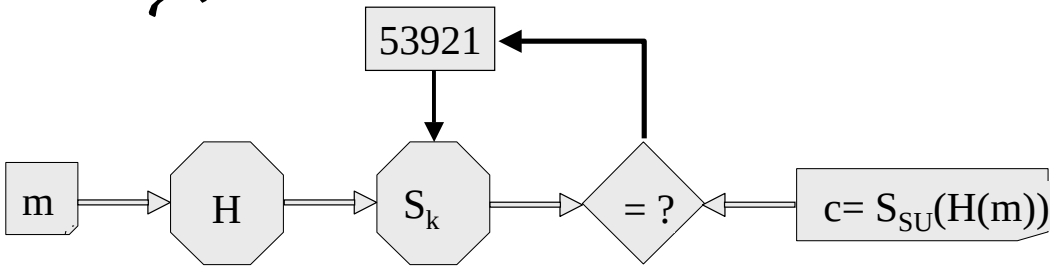
- **attacchi alla chiave segreta**

- Chiave condivisa
- Chiave privata
- Prova d'identità
- …

Dati segreti

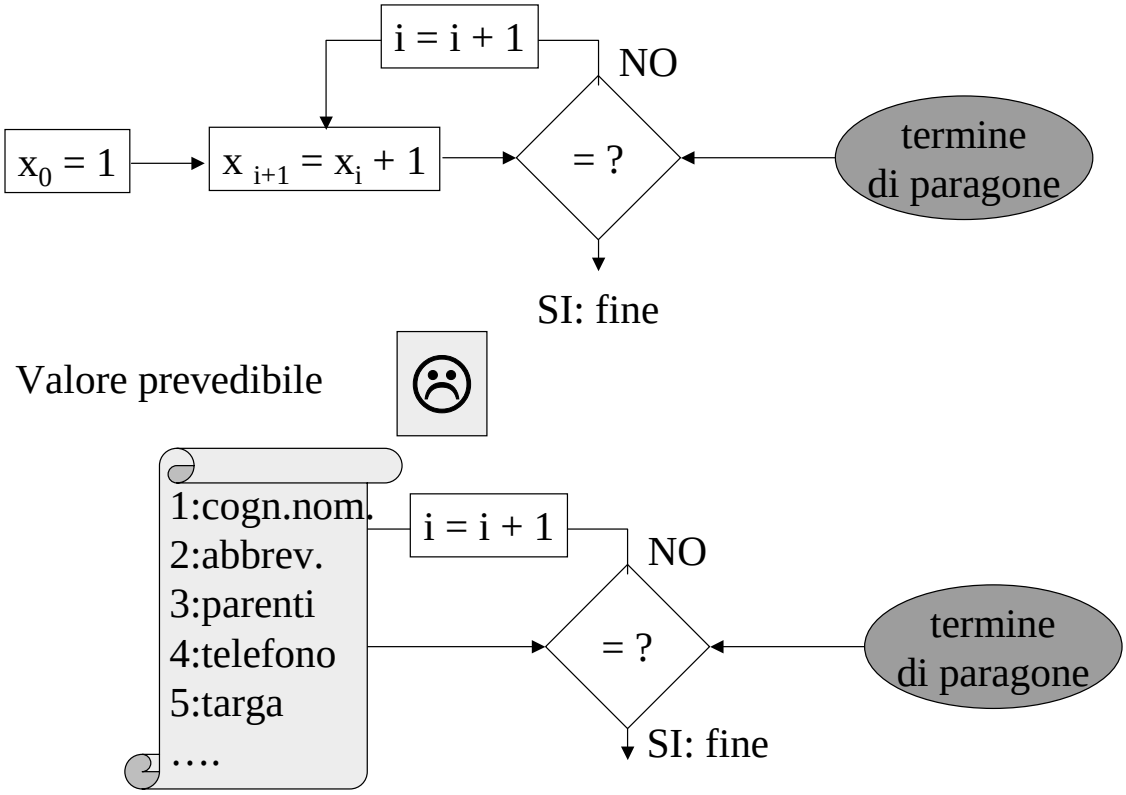
1. Indovinare
2. Intercettare
3. Dedurre





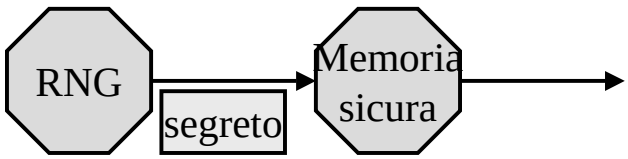
Bancomat: 3 tentativi

L'attacco con forza bruta



Tirare ad indovinare

R12: "*i simboli della stringa che rappresenta un segreto devono essere* ***molti*** *e* ***scelti a caso***"

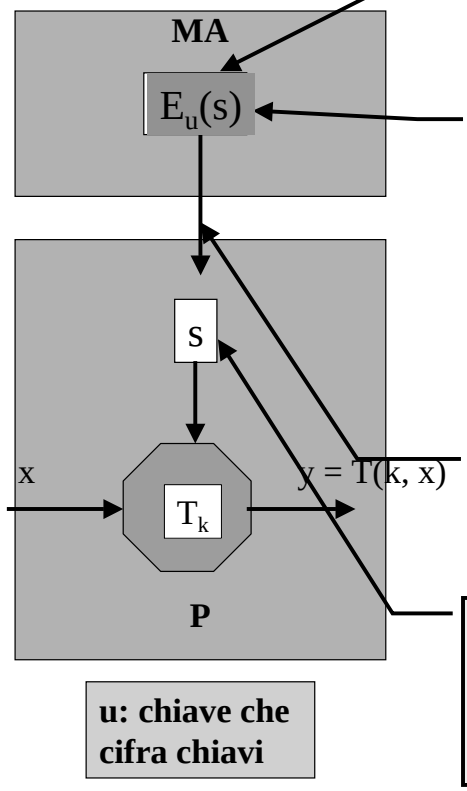


Valutazioni

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

R13: "*un dato segreto deve essere frequentemente modificato*"

Memorizzazione ed uso di una chiave segreta

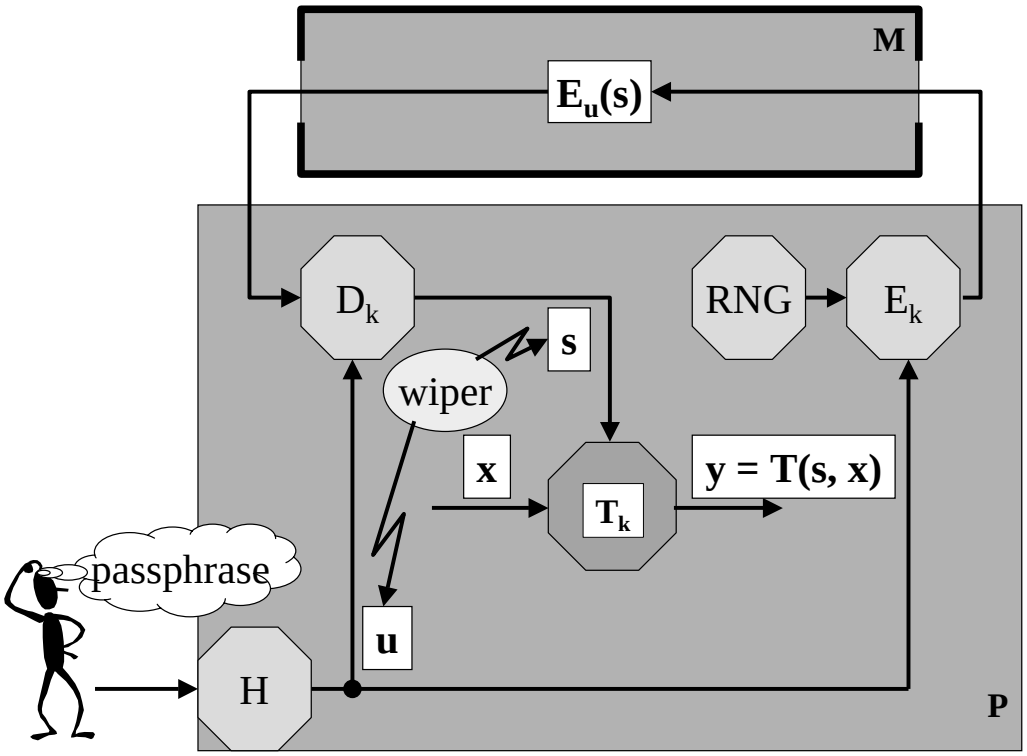


- R14:"*solo il proprietario della chiave segreta deve avere accesso alla cella in cui è memorizzata*".
- R15:"*la chiave segreta deve essere memorizzata in forma cifrata*".

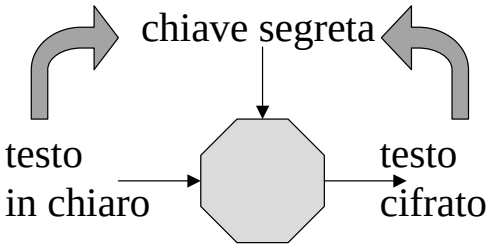
1. ~~Indovinare~~
2. Intercettare
3. Dedurre

- R16:"*la forma cifrata deve essere mantenuta anche nella comunicazione della chiave dalla memoria ausiliaria al processore*".
- R17:"*l'unità di elaborazione deve decifrare la chiave, impiegarla nell'esecuzione dell'algoritmo crittografico e cancellare poi accuratamente il dato dalla sua memoria di lavoro*.

Memorizzazione ed uso della passphrase



Deduzione di un segreto dal suo uso



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