

## CRITTOGRAFIA A CHIAVE PUBBLICA

Permette di scambiare la chiave su un canale non sicuro

### CIFRARI SIMMETRICI

La chiave per cifrare e' uguale a quella per decifrare:

Alice e Bob possono scambiarsi.

### CIFRARI ASSIMETRICI A CHIAVE PUBBLICA

Tutti possono inviare messaggi cifrati, solo il ricevente puo' decifrarli; esistono due chiavi:

cifratura  $K_{pub}$  pubblica, nota a tutti

decifratura  $K_{priv}$  privata, solo Bob (ricevente) la conosce

I sistemi a chiavi pubblica sono simmetrici. Alice e Bob possono scambiarsi; quelli a chiavi private sono assimmetrici.

La funzione di cifratura deve avere una 1-way trapdoor:

$$c = C(m, K_{pub}) \text{ computazionalmente facile}$$

decifrare  $c$  e' difficile a meno di non avere  $K_{priv}$

RIVEST SHAMIR ADLEMAN: Cifraio RSA, e' basato su una 1-way trapdoor

Gli schemi a chiavi pubbliche sono facili per comunicazioni many-to-one, non hanno bisogno di uno scambio di chiavi n utenti  $\longrightarrow$  2n chiavi pubbliche e private

Sono più lenti che i cifrari simmetrici

CIFRARI

I B R I D I

In genere viene utilizzata un cifrario a chiave pubblica per mandare una chiave usata in un cifrario simmetrico (AES).

Vulnerability and attack: chosen-plaintext

# RAPPRESENTAZIONE MATEMATICA I. OGGETTI

$\Gamma := \text{Alfabeto} :=$  insieme finito L-caratteri

Un oggetto e' rappresentato universalmente da una sequenza ordinata di caratteri dell'alfabeto.

$$\# \Gamma = 5$$

$N :=$  oggetto da rappresentare

$\mathcal{J}(s, N) := \text{massimale Lunghezza delle sequenze di}$

rappresenta un oggetto dell'interno (i.e. nome più lungo)

$$d(s, N) := \min_{m \in \mathbb{Z}} d(s, N)$$

Suppongo rappresentazione binaria:

$$\Gamma = \{0,1\}$$

numero totale di sequenze lunghe  $K$ :  $\sum_{i=1}^K 2^i = 2^{K+1} - 2$

N oggetti:  $2^{K+1} - 2 \geq N$

$$K \geq \log_2(N+2) - 1$$

$$d_{\min}(2, N) = \lceil \log_2(N+2) - 1 \rceil$$

$$\Rightarrow \lceil \log_2 N \rceil - 1 \leq d_{\min}(2, N) \leq \lceil \log_2 N \rceil$$

Definisco rappresentazione efficiente una logaritmica nel numero di stringhe (i.e. numero massimo di caratteri di ordine logaritmico)

CENNI SUL CALCOLABILITÀ E COMPLESSITÀ

- Problemi non decidibili

{ calcolabilità }

- Problemi decidibili:

- trattabili (polinomiale)

{ complessità

- non trattabili (esponenziale)

Definisco numerabile un insieme: cui elementi possono essere messi in corrispondenza biunivoca con quelli dei naturali.

gli elementi possono essere elencati

Le stringhe finite di simboli in un insieme finito e numerabile

(si usa l'ordinamento canonico per generare opportunamente ordinate)

Vi problema e' per sua natura una funzione, pero' le funzioni non sono numerabili.

### DIAGONALIZZAZIONE

Si dimostra che  $F := \{ \text{funzioni } f \mid f: \mathbb{N} \rightarrow \{0,1\} \}$

$f \in F$  puo' essere scritta a partire da una regola di costruzione (non sempre) o da una sequenza infinita.

Te  $F$  non e' numerabile

Dimo. Si procede per assurdo

enumero ogni funzione

$x$	0	1	2	.	.	.	.
$f_0$	.	.	.	-	-	-	-
$f_1$	-	-	-	-	-	-	-
$f_2$	-	-	-	.	.	.	.
:	-	-	-	.	.	.	.

Considero  $g \in F$ :

$$g(x) = \begin{cases} 0 & \text{se } f_x(x) = 1 \\ 1 & \text{alt} \end{cases}$$

$g$  non corrisponde ad alcuna  $f_i$  in tabella

differisce da ogn  $f_i$  per il valore sulla diagonale principale

Per assurdo:  $\exists j \mid g(x) = f_j(x)$

$$g(j) = f_j(j) \wedge \{ \text{definizione} \} \Rightarrow F$$

$\Rightarrow F$  non e' numerabile

$\Rightarrow$  gli insiem i funzioni

$\Leftrightarrow \mathbb{Q}^{\mathbb{N}} \subset F$  non sono numerabili

$\Rightarrow$  gli insiem di funzioni non sono numerabili

La conoscenza umana e' finita;

{ def. } gli algoritmi devono una sequenza finita

$\Rightarrow$  gli algoritmi sono numerabili

# Problemi > # Algoritmi

Un esempio di problema non decabile e' il problema dell'arresto: Arresto : { Istanze }  $\rightarrow \{0,1\}$

Prendi un algoritmo A e i dati in input D,  
decidere in tempo finito se  $A(D)$  termina.

Definisco Arresto(A,D) =  $\begin{cases} 1 & \text{se } A(D) \text{ termina} \\ 0 & \text{altr} \end{cases}$   
che determina le risposte in tempo finito.

## PARADOSSO (A)

```
while (Arresto(A,A)) { ; }
```

Dim: Paradosso (A) termina  
 $\Leftrightarrow$

$x = \text{Arresto}(A,A) = 0$   
 $\Leftrightarrow$

$A(A)$  non termina

Paradosso (Paradosso) termina?  
 $\Leftrightarrow$

Arresto (Paradoxa, Paradosso) = 0  
 $\Leftrightarrow$

Paradosso (Paradosso) non termina

ASURDO

Paradosso e Arresto non possono esistere

Per le teorie di Church-Turing tutti i ragionevoli modelli di calcolo risolvono le stesse classi di problemi