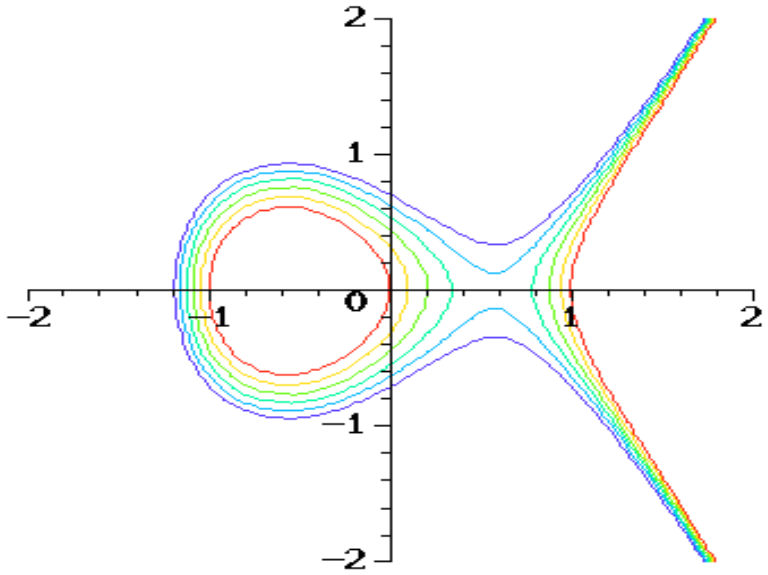


NUMERI E CRITTOGRAFIA

Carlo Toffalori (Camerino)
13 novembre 2014
ITIS Divini, San Severino Marche

Edgar Allan Poe, *Lo scarabeo d'oro*: la caccia al tesoro degli antichi pirati!

Istruzioni incomprensibili…

53‡‡†305))6*;**48**26)4‡.)4‡);806*;**48**†8Ч60))85;1‡
(;:‡*8†83(88)5*†;46(;88*96*?;8)*‡(;**48**5);5*†2 …

Il protagonista (William Legrand):

- 8 *come* e
- ;48 *come* the

e via dicendo.

Lo schema generale

- Due personaggi: **A** e **B** (due degli antichi pirati, oggi ***Alice*** e ***Bob***)
- Il "cattivo" (non necessariamente cattivo): **E** (William Legrand, oggi *Eve*, "the eavesdropper")

L'obiettivo di **A** e **B**: trasmettersi informazioni senza che **E** le capisca; le operazioni sono allora due

- Cifrare
- Decifrare

con una chiave opportuna. In questo consiste la **Crittografia.**

L'obiettivo di **E**: violare il sistema usato da **A** e **B**; questa è la **crittoanalisi**.

Un linguaggio "universale": numeri invece che lettere

A	0	N	13
B	1	O	14
C	2	P	15
D	3	Q	16
E	4	R	17
F	5	S	18
G	6	T	19
H	7	U	20
I	8	V	21
J	9	W	22
K	10	X	23
L	11	Y	24
M	12	Z	25

Procedimento usuale di codifica e decodifica

Permutare numeri o lettere.

Esempio (Giulio Cesare)
Si sostituisce ogni lettera con quella che la segue di 3 passi:
A → D, B → E, C → F, …, W → Z, X → A , Y → B, Z → C

Matematicamente parlando…

$0 \rightarrow 3,\ 1 \rightarrow 4,\ 2 \rightarrow 5, \ldots, 22 \rightarrow 25,\ 23 \rightarrow 0,\ 24 \rightarrow 1, 25 \rightarrow 2$

Dunque

- si codifica tramite $x \rightarrow x+3$ (ma *attenzione a 23, 24 e 25*)
- si decodifica tramite $x \rightarrow x - 3$ (ma *attenzione a 0, 1 e 2*)

Le chiavi ± 3

Essenziale: chiave di codifica / decodifica

- In genere la stessa
- Comunque tra loro direttamente collegate e computazionalmente equivalenti.

Una strana aritmetica:

- arrivati a 26, si ritorna a 0

Come nelle ore del giorno (arrivati a 24 si torna a 0): *l'aritmetica dell'orologio*

- modulo 26,
- modulo 24,
- modulo m per ogni intero positivo m.

Procedimento usuale di crittoanalisi

Analisi di frequenza (Ishaq al-Kindi): si basa sul confronto tra

- le lettere più comuni nell'alfabeto (*e*, *the*, …)
- le lettere più frequenti nel messaggio (8, ;48, …)

Ci sono cifrari inattaccabili e sicuri ?

Esempio: Cifrario di Vernam, *One-Time-Pad* (taccuino monouso), 1917. Porta l'idea di Vigenère alle estreme conseguenze:

- la chiave è tanto lunga quanto il messaggio;

Inoltre

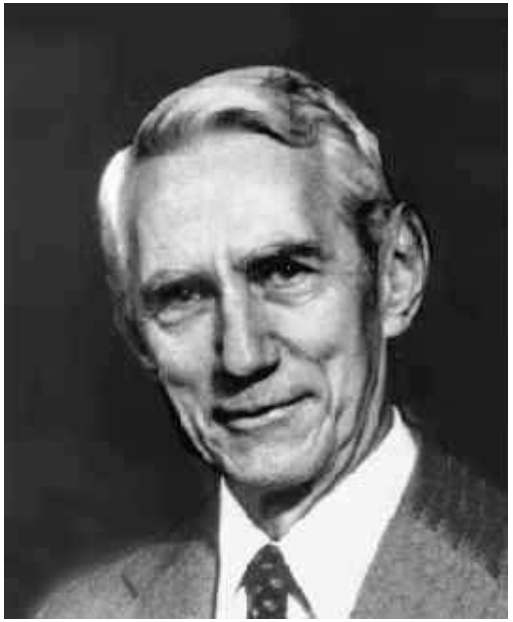
- l'informazione e la chiave sono sequenze di 0 ed 1
- la codifica e la decodifica avvengono sommando con la chiave modulo 2

Messaggio in chiaro	1	0	0	0	1	0	1	0	1	0	0	1	1	1	0
Chiave	1	1	0	1	1	1	1	1	0	1	1	0	0	1	0
Messaggio cifrato	0	1	0	1	0	1	0	1	1	1	1	1	1	0	0

Svantaggi:

- generazione della chiave (sequenza casuale di 0 ed 1)
- trasmissione della chiave

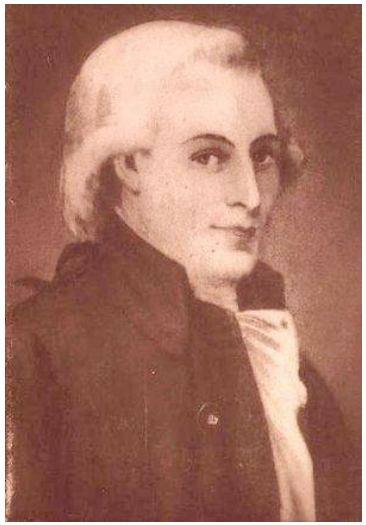
Una chiave per ogni messaggio (“taccuino monouso”): troppo costoso!
Eppure…



La crittografia secondo Claude Shannon*: i crittosistemi alla Vernam sono gli unici garantiti sicuri!*

Ma un cambio di prospettiva: dalla crittografia classica…

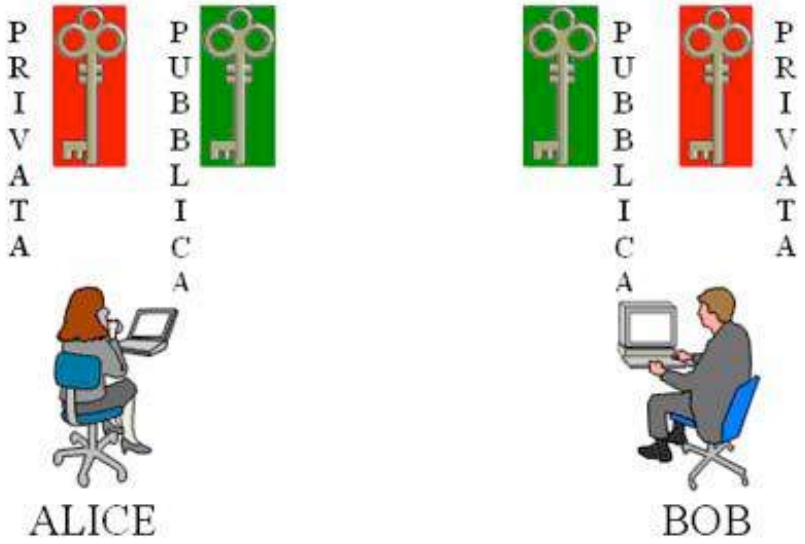
1. pochi utenti
2. chiavi di codifica/decodifica preventivamente concordate e scambiate
3. la codifica computazionalmente equivalente alla decodifica (Casanova e M.me d'Urfé)



... alla crittografia moderna

- trasmissione in rete, voto telematico, ...
- un gran numero di utenti **A**, **B**, **E**, ...

Non più buoni o cattivi, tutti sono chiamati a partecipare.
Ma… ***il segreto di tutti è il segreto di nessuno!***



Come rimediare

a) Non più una sola chiave per codifica e decodifica **ma**, per ogni utente A,
    - una chiave pubblica di codifica (disponibile a chiunque voglia scrivere ad A)
    - una chiave privata di decodifica (conosciuta **solo** da A)

b) Decodificare deve essere enormemente più difficile che codificare (salvo che per A)

Un esempio pratico: il *doppio lucchetto* (Diffie-Hellman 1976)

- B invia un messaggio ad A in una scatola chiusa con la sua chiave (il primo lucchetto)
- A chiude ulteriormente con la sua chiave (i secondo lucchetto) e rispedisce a B
- B apre il suo lucchetto e rispedisce ad A
- A apre il suo lucchetto e la scatola

In teoria: le funzioni a senso unico

- facili da calcolare
- invertibili
- con inversa difficile da calcolare

L'idea

- f codifica
- f^{-1} decodifica

Osservazione

La nozione di funzione a senso unico è

- non rigorosa (che significa: facile/difficile da calcolare?)
- da controllare periodicamente (quello che è difficile oggi può diventare facile domani)

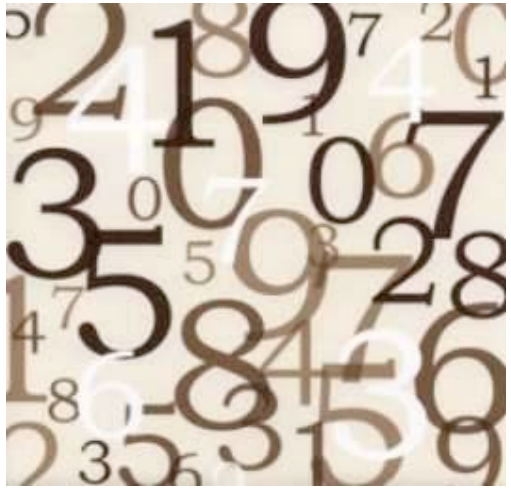
Problema: Dove cercare funzioni a senso unico?

Ma dove trovare esempi concreti di funzioni a senso unico?

- ***Speculazioni teoriche*** (da Leibniz a Turing e a P = NP)

- ***I numeri naturali*** 0, 1, 2, 3, 4, 5, ...

- apparentemente banali
- nascondono i più grossi misteri della matematica



2 citazioni (libere)

- L. Kronecker: *"i soli creati da Dio"*
- A. Weil: *"dimostrano l'esistenza di Dio e anche quella del diavolo"*

I misteri più grandi sui naturali riguardano la divisione

Un naturale $N > 1$ è

- *primo* se gli unici divisori di N sono 1 e N
- *composto* altrimenti

Teorema fondamentale dell'Aritmetica. Ogni naturale $N > 1$ si decompone in uno ed un solo modo (a meno dell'ordine dei fattori) come prodotto di numeri primi.

$$15 = 3 \cdot 5, \quad 18 = 2 \cdot 3^2, \quad 24 = 2^3 \cdot 3, \quad 32 = 2^5 , \ \ldots$$

Due problemi emergono in modo spontaneo

- Un input comune: $N \in \mathbf{N}$, $N > 1$
- Due output collegati ma diversi:

- *(PRIMI)* N è primo o composto?
- *(FATTORIZZAZIONE)* la decomposizione di N nei suoi fattori primi (o almeno, per N composto, d divisore di N, $1<d<N$)

Un algoritmo elementare (conosciuto già dagli antichi Greci): si esplora $1 < d < N$ e si divide N per d

- se la divisione è esatta per qualche d, si ha che N è composto (e si trova un suo divisore $d \neq 1, N$),
- se la divisione non è esatta per nessun d, N è primo.

Possibili scorciatoie:

- basta controllare $1 < d <$ radice quadrata di N
  (se $N = d \cdot q$ con $d, q > 1$, allora d o q è minore o uguale della radice quadrata di N)
- se d non funziona, neppure 2d, 3d, ... funzionano

Gauss, *Disquisitiones Aritmeticae*, 1801 (articolo 329)
"Il problema di separare i primi dai composti e di decomporre i secondi nei loro fattori primi è conosciuto essere uno dei più importanti ed utili in Matematica. La dignità stessa della scienza sembra richiedere di esplorare ogni possibile mezzo per la soluzione di un problema così elegante e famoso."

Perché questa esigenza due millenni dopo i Greci?

Gauss: *"Le tecniche conosciute in precedenza richiederebbero una fatica intollerabile anche per i più instancabili* ***calcolatori****."*

Algoritmo elementare: per controllare N un numero di divisioni fino alla radice quadrata di N, esponenziale rispetto alla lunghezza di N

Possibile separare *PRIMI* da *FATTORIZZAZIONE*!

Teorema di Wilson. Per N naturale > 1, N è primo se e solo se (N – 1)! + 1 è divisibile per N.

Da ricordare: $(N - 1)! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (N - 1)$

- Controlla *PRIMI*, non *FATTORIZZAZIONE*
- Richiede il calcolo di (N – 1)! e di conseguenza N-2 moltiplicazioni (un numero esponenziale rispetto alla lunghezza di N)

Uno dei tanti misteri sui primi: la Congettura di Goldbach
Apostolos Doxiadis, *Zio Petros e la congettura di Goldbach*

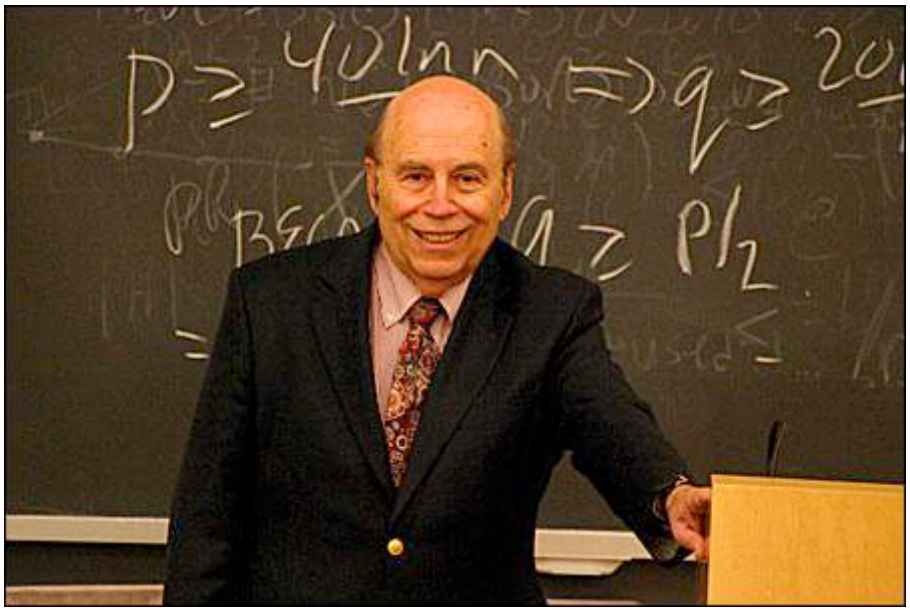
C. Goldbach 1742, lettera a L. Euler: "*Ogni N ≥ 6 è la somma di al più 3 primi (non necessariamente distinti)*"

L. Euler: "*Basta provare che ogni numero pari $N \geq 4$ è la somma di 2 primi*" (perché?)

$4 = 2 + 2$, $6 = 3 + 3$, $8 = 3 + 5$, $10 = 5 + 5 = 3 + 7$, …

Riconoscere i PRIMI

1978: un algoritmo efficiente ma fallibile (*Miller-Rabin*)



L'idea: sacrificare la precisione per aumentare la velocità, si ammettono risposte sbagliate o silenzi, purché la probabilità di errore sia bassa

E. Borel: un evento che ha probabilità < 10^{-50} non accadrà mai, e se anche accade non sarà mai rilevato.

Algoritmi probabilistici veloci

- Montecarlo: risposte probabilmente vere in tempi certamente rapidi (si ricorre a testimoni che possono mentire)
- Las Vegas: risposte certamente vere in tempi probabilmente rapidi (si ricorre a testimoni che possono tacere).

L'algoritmo di Miller-Rabin

- *probabilità di errore* dopo il primo tentativo: è al più 1/4 (perché 3 testimoni a su 4 sono onesti),
- *tempi di lavoro*: un polinomio di grado 5 rispetto alla lunghezza di N.

Ma dopo 100 applicazioni con esito concorde

- *probabilità di errore* si riduce sotto $4^{-100} < 10^{-50}$
- *tempo di lavoro* ancora un polinomio di grado 5 rispetto alla lunghezza di N.

2002: un algoritmo efficiente e infallibile (*AKS Agrawal-Kayal-Saxena*)

Tempi di lavoro (nella ”implementazione” di Lenstra-Pomerance, 2005): approssimativamente un polinomio di grado 6 nella lunghezza dell’input.

FATTORIZZAZIONE

Vari metodi (curve ellittiche, frazioni continue, ...), nessuno polinomiale nella lunghezza di N: ma attenzione ai procedimenti quantistici (P. Shor, 1994) !

L’algoritmo del letamaio (Lenstra jr.): *“Ammettiamo di avere due numeri primi $p \neq q$ e il loro prodotto $N = p \cdot q$ e di perdere p, q in un letamaio, così che ci rimane solo N. Deve essere sentito come una sconfitta della scienza il dover ammettere che l’algoritmo più rapido per recuperare p e q è quello di cercare nel letamaio”*

Ritorno alla crittografia

Il criptosistema RSA (*Rivest-Shamir-Adleman*, 1978)

La funzione a senso unico: la *moltiplicazione*

Per $p \neq q$ primi grandi

- facile calcolare $N = p \cdot q$,
- lento e difficile recuperare p, q dal prodotto N .