crittografia a chiave pubblica

Whitfield Diffie

Martin Hellman

New Directions in Cryptography

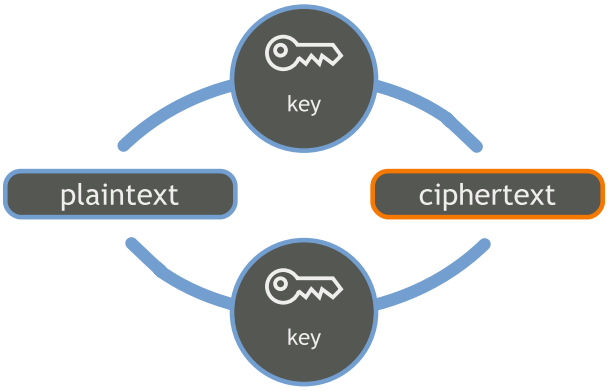
We stand today on the brink of a revolution in cryptography. The development of cheap digital hardware . . . has brought the cost of high grade cryptographic devices down to where it can be used in . . . remote cash dispensers and computer terminals. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

Diffie e Helmann, IEEE IT **22** (1976)

- nel 1976, Diffie e Hellman pubblicano “New directions in cryptography”
- viene considerato l’inizio della crittografia a chiave pubblica (PKC)
- propongono uno schema di scambio della chiave basato sul logaritmo discreto
- introducono l’idea di un crittosistema a chiave pubblica
- la prima realizzazione di un crittosistema di questo tipo si ha nel 1978 con l’RSA

crittografia simmetrica

- Alice e Bob condividono la stessa chiave k – scelta e scambiata fra loro prima di cominciare a comunicare
- la chiave dà luogo a una funzione di cifratura e_k e una funzione di decifratura d_k
- è facile ricavare d_k da e_k
- se si sa cifrare, si sa anche decifrare

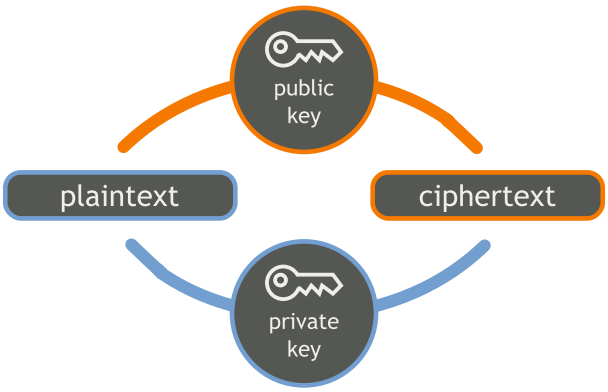


come condividere la chiave?

- prima di cominciare a comunicare Alice e Bob devono scegliere una chiave segreta
- usando un canale sicuro
- se l'avversario intercetta la chiave, la comunicazione è completamente compromessa

idea della crittografia a chiave pubblica

- sviluppare un crittosistema in cui data la funzione di cifratura e_k sia computazionalmente difficile determinare d_k
- Bob rende pubblica la sua funzione di cifratura e_k
- Alice (e chiunque altro) può scrivere a Bob, cifrando il messaggio con la e_k senza bisogno di accordi preliminari
- Bob è l'unico che può decifrare il messaggio
- analogia con un lucchetto, che chiunque può usare, ma di cui solo Bob ha la chiave



funzioni unidirezionali

- bisogna che la funzione di cifratura e sia una funzione unidirezionale (one-way function)
- informalmente, una funzione invertibile $f : \mathcal{P} \to \mathcal{C}$ si dice unidirezionale se
  - dato $x \in \mathcal{P}$, il calcolo di $f(x)$ è facile
  - per quasi tutti gli $y \in \mathcal{C}$ il calcolo di $f^{-1}(y)$ è difficile
  - dato $x \in \mathcal{P}$, il calcolo di $f(x)$ è realizzabile con una complessità polinomiale
  - per quasi tutti gli $y \in \mathcal{C}$ il calcolo di $f^{-1}(y)$ non è realizzabile con una complessità polinomiale
- Esempio una funzione ritenuta unidirezionale: sia $n = pq$, p e q numeri primi "abbastanza grandi", b un intero coprimo con $\phi(n)$; sia $f : \mathbb{Z}_n \to \mathbb{Z}_n$ t.c.

$$f(x) = x^b \pmod{n}$$

trapdoor

- se la funzione è unidirezionale, anche per Bob è impossibile decifrare il messaggio
- una trapdoor (botola) one-way function è una funzione unidirezionale che diventa facile da invertire, se si conosce un'informazione supplementare
- l'informazione supplementare viene tenuta segreta da Bob, e usata per decifrare
  - ci sarà una chiave pubblica nota a tutti – che serve per cifrare
  - e una chiave privata nota solo a Bob – che serve per decifrare

cenni sulla complessità computazionale

- come si misura la complessità computazionale di un algoritmo?
- si parla di algoritmi che operano su numeri interi
  - ex: dati due interi trovare la somma, il prodotto, il MCD
- il "tempo" di esecuzione dipende dalla grandezza dell'input
- la grandezza di un input N è pari al numero di bit nella sua rappresentazione binaria ($\approx \log_2 N$)
- la complessità di un algoritmo si misura in termini di operazioni bit
- sono operazioni bit:
  - addizione fra due cifre binarie (es. $0 + 1$);
  - sottrazione fra due cifre binarie (es. $1 - 0$);
  - moltiplicazione fra due cifre binarie (es. $1 \cdot 1$);
  - divisione di un intero a *due* cifre binarie per *una* cifra binaria (es. 10 diviso 1);
  - traslazione a *sinistra* di un posto, cioè moltiplicazione per 2, e traslazione a *destra* di un posto, cioè divisione per 2.

complessità polinomiale e esponenziale

Definizione

La complessità computazionale di un algoritmo che opera sugli interi è data dal numero di operazioni bit occorrenti per eseguirlo.

è una funzione (della lunghezza dell'input)

Definizione

Un algoritmo $\mathcal{A}$ per eseguire un calcolo su numeri interi si dice polinomiale se esiste un intero positivo d tale che il numero di operazioni bit necessarie per eseguire l'algoritmo su interi di lunghezza binaria al più k è $\mathcal{O}(k^d)$.

Definizione

Un algoritmo si dice esponenziale se il numero di operazioni bit necessarie per eseguire l'algoritmo su interi di lunghezza binaria al più k è dello stesso ordine di 2^{ck}, per una costante $c > 0$

- polinomiale $\leftrightarrow$ (computazionalmente) facile
- esponenziale $\leftrightarrow$ (computazionalmente) difficile
- mostare che esiste un algoritmo polinomiale che risolve un dato problema è semplice - basta descrivere l'algortimo
- ma come si fa a mostare che non esiste un algoritmo polinomiale che risolve un dato problema?