IL METODO CRITTOGRAFICO RSA

Lezione 2h

Attività 2h

In crittografia la sigla **RSA** indica un algoritmo di crittografia asimmetrica, inventato nel 1977 da Ronald Rivest, Adi Shamir e Leonard Adleman utilizzabile per cifrare o firmare informazioni.

Il sistema di crittografia si basa sull'esistenza di due chiavi distinte, che vengono usate per cifrare e decifrare. Se la prima chiave viene usata per la cifratura, la seconda deve necessariamente essere utilizzata per la decifratura e viceversa. La questione fondamentale è che, nonostante le due chiavi siano fra loro dipendenti, non è possibile risalire dall'una all'altra, in modo che se anche si è a conoscenza di una delle due chiavi, non si possa risalire all'altra, garantendo in questo modo l'integrità della crittografia.

Per realizzare un sistema crittografico pubblico con il cifrario asimmetrico è importante che un utente si crei autonomamente entrambe le chiavi, denominate "diretta" e "inversa", e ne renda pubblica una soltanto. Così facendo si viene a creare una sorta di "elenco telefonico" a disposizione di tutti gli utenti, che raggruppa tutte le chiavi dirette, mentre quelle inverse saranno tenute segrete dagli utenti che le hanno create e da questi utilizzate solo quando ricevono un messaggio cifrato con la rispettiva chiave pubblica dell'"elenco" da parte di un certo mittente, ottenendo in questo modo i presupposti necessari alla sicurezza del sistema.

Per ottenere una discreta sicurezza è necessario utilizzare chiavi binarie di almeno 2048 bit. Quelle a 512 bit sono ricavabili in poche ore. Le chiavi a 1024 bit, ancora oggi largamente utilizzate, non sono più consigliabili. La fattorizzazione di interi grandi, infatti, è progredita rapidamente mediante l'utilizzo di hardware dedicati, al punto che potrebbe essere possibile fattorizzare un intero di 1024 bit in un solo anno di tempo, al costo di un milione di dollari (un costo sostenibile per qualunque grande organizzazione, agenzia o intelligence).

Vediamo come funziona RSA con due persone, A e B, che vogliono scambiarsi i messaggi. A è l'utente che genera due chiavi pubbliche (utilizziamo per l'esempio numeri primi piccoli).

1. A genera **due numeri primi distinti p** e **q** e li moltiplica tra di loro ottenendo il numero **N** che viene reso pubblico, mentre **p** e **q** devono restare segreti.
   - **Esempio:**

p = 5; q = 11 p.q = 5.11 = **55**
N = 55, prima chiave pubblica

2. A calcola **b** (funzione di Eulero): **b** = Φ(n) = (p-1)*(q-1). Il numero **b** deve restare segreto.
   - **Esempio:**

Φ(55) = (5 - 1)·(11 - 1) = 4·10 = **40**
b = 40

3. A calcola il primo intero **e** che sia primo con **b** (non abbia divisori in comune, ovvero MCD(e, b) = 1). Il numero **e** è la seconda chiave pubblica.
    - **Esempio:**

e = 3 perchè MCD(3, 40) = 1
e = 3 seconda chiave pubblica

4. A calcola il **numero d inverso di e nella classe di congruenza modulo b,** che è il più piccolo x per cui sia **e·d** mod **b** = 1; il numero **d é la chiave per decifrare e deve** r**estare segreta, chiave privata**. Per il calcolo di d è efficiente un'estensione del classico algoritmo di Euclide per l'MCD (detto anche teorema cinese del resto); noi invece useremo semplice metodo a tentativi:
    - **Esempio:**

d = 2 -> 2.3 mod 40 = 6 NO d = 3 -> 3.3 mod 40 = 9 NO d = 4 -> 4.3 mod 40 = 12 NO ... d = 26 -> 26.3 mod 40 = 78 mod 40 = 38 NO d = 27 -> 27.3 mod 40 = 81 mod 40 = 1 SI
d = 27

B adesso vuole trasmettere un messaggio ad A, B lo scompone inizialmente in una sequenza di numeri (in precedenza ci si è accordati riguardo alla modalità di "traduzione"; potrebbero essere **p.es. i codici ASCII**, oppure l'ordine delle lettere dell'alfabeto traslate di una chiave,ma ci sono anche altri sistemi di traduzione.

Quindi B legge le chiavi pubbliche di A (**N** e **e)** e trasmette i numeri **m** uno alla volta cifrandoli con la formula $\mathbf{c \equiv m^e \bmod N}$.

- **Esempio:** per trasmettere il numero **7**, B **calcola** $\mathbf{c \equiv m^e \bmod N = 7^3 \bmod 55 = 343 \bmod 55 = 13}$; il numero da trasmettere è quindi 13

A usa per questo la chiave di decifrazione **d**, segreta, che permette di recuperare m grazie alla formula $\mathbf{m \equiv c^d \bmod N}$, questa uguaglianza è un teorema molto importante della teoria dei numeri.

- **Esempio:** $\mathbf{m = c^d \bmod n = 13^{27} \bmod 55 = 7}$ **(**provare per credere!)
- Il **codice RSA** viene considerato sicuro perché non è ancora stato trovato il modo per fattorizzare numeri primi molto grandi, che nel nostro caso significa riuscire a trovare **p** e **q** conoscendo **N.** . Nel corso degli anni l'algoritmo RSA ha più volte dimostrato la sua robustezza: in un esperimento del 1994, coordinato da Arjen

Lenstra dei laboratori Bellcore, per “rompere” una chiave RSA di 129 cifre, svelando il meccanismo con cui quella chiave generava messaggi crittografati, sono stati necessari 8 mesi di lavoro coordinato effettuato da 600 gruppi di ricerca sparsi in 25 paesi, che hanno messo a disposizione 1600 macchine da calcolo, facendole lavorare in parallelo collegate tra loro attraverso Internet.

Data la mole delle risorse necessarie per rompere la barriera di sicurezza dell’algoritmo RSA, è chiaro come un **attacco alla privacy** di un sistema a doppia chiave non sia praticamente realizzabile. Inoltre, nell’esperimento era stata utilizzata una chiave di 129 cifre mentre i programmi di crittografia attualmente a disposizione prevedono chiavi private con una “robustezza” che raggiunge e supera i 2048 bit, risultando quindi praticamente inattaccabili, visto anche che l’ordine di grandezza dei tempi necessari alla rottura di chiavi di questo tipo è esponenziale e passa in fretta da qualche giorno a qualche centinai di anni.

Attività : Creare le due chiavi utilizzando come p e q rispettivamente 13 e 19, comunicarle al compagno di banco che ti deve inviare un messaggio crittato, decritta quindi il messaggio, seguendo i passaggi descritti sopra.