Introduzione alla crittografia. Il crittosistema RSA e la sua sicurezza

Prof. Massimiliano Sala

MINICORSI 2011. Crittografia a chiave pubblica: oltre RSA

Università degli Studi di Trento, Lab di Matematica Industriale e Crittografia

24 marzo 2011

1.INTRODUZIONE

Introduzione

CRITTOGRAFIA = scienza/arte delle scritture segrete

Obiettivo:

consentire a due utenti di comunicare su un canale potenzialmente insicuro, senza permettere a una terza persona di comprendere il contenuto dei messaggi

Alice

Bob

Alcune definizioni...

testo in chiaro

– – – – – – – – – – >

Sistema di cifratura:

- $\mathcal{M}$ = insieme di messaggi;
- $\mathcal{C}$ = insieme di testi cifrati;
- $\mathcal{K}$ = insieme di chiavi;
- ad ogni chiave k sono associate due funzioni: una che cifra (e_k) e una che decifra (d_k).

Due categorie di cifrari

- A CHIAVE PRIVATA (o SIMMETRICA)
- A CHIAVE PUBBLICA (o ASIMMETRICA)

Due categorie di cifrari

Cifrari a CHIAVE PRIVATA

- *un'unica* chiave privata k, sia per l'operazione di cifratura che per quella di decifratura
- solo e conoscono k ($\Rightarrow$ e_k e d_k)
- è necessario che prima Alice e Bob scelgano k, utilizzando un canale sicuro
- è difficile calcolare d_k da e_k
- ci sono 2 tipi di cifrari a chiave simmetrica: i *Block Ciphers*, generalmente più sicuri ma pesanti, e gli *Stream Ciphers*, meno pesanti ma più insicuri

Due categorie di cifrari

Cifrari a CHIAVE PUBBLICA

- Alice sceglie una chiave pubblica k_{p_a}
- Bob sceglie una chiave pubblica k_{p_b}
- Associata a ciascuna chiave pubblica, ci sono due chiavi private k_{s_a} e k_{s_b}
- con le chiavi pubbliche si cifra il messaggio, $e_{k_{p_a}}$ e $e_{k_{p_b}}$; con quelle private invece si decifra $d_{k_{s_a}}$ e $d_{k_{s_b}}$

Due categorie di cifrari

Cifrari a CHIAVE PUBBLICA (scambio di chiavi)

Il meccanismo delle chiavi pubbliche e private si può usare per scambiare o negoziare una chiave segreta da usare nei cifrari simmetrici

Confronto fra Cifrari

Cifrari a Chiave Privata:

- richiedono un canale sicuro per la comunicazione della chiave
- sono computazionalmente più leggeri anche quando si tiene alto il livello di sicurezza

Cifrari a Chiave Pubblica:

- non serve un canale sicuro
- sono computazionalmente più pesanti

2. L'ALGORITMO RSA

L'algoritmo RSA

L'RSA è un cifrario a chiave pubblica che permette di cifrare un messaggio sfruttando alcune proprietà elementari dei numeri primi.

Questo cifrario prende il nome dalle iniziali dei matematici che nel 1976 lo crearono: Rivest, Shamir e Adleman.

La loro idea fu quella di sfruttare la difficoltà di fattorizzare un numero intero. Di fatti la chiave pubblica è un numero N ottenuto moltiplicando due numeri primi molto grandi che restano segreti.

Descrizione di RSA

Per capire come funziona l'RSA abbiamo bisogno inanzitutto di due risultati dovuti a Eulero e Fermat.

Definizione (Funzione di Eulero φ)

La funzione di Eulero associa a un intero N il numero degli interi primi con N e minori di N (compreso 1).

$$\Phi(N) = |\{m \in \mathbf{N} | gcd(1, m) = 1,\ 1 \leq m < N\}|$$

Descrizione di RSA

Teorema (Teorema di Fermat-Eulero)

Dati due interi qualsiasi N e m, con $gcd(N, m) = 1$:

$$m^{\varphi(N)} \equiv 1 \mod (N)$$

Corollario (Piccolo Teorema di Fermat)

Se N è primo e m è coprimo con N:

$$m^{N-1} \equiv 1 \mod (N)$$

Descrizione di RSA

Possiamo così passare a descrivere il cifrario.

Alice vuole trasmettere il messagio m a Bob .

- Bob deve creare una chiave pubblica $k_{p_b} = (N, e)$ ed una privata $k_{s_b} = (N, d)$ stando attento a non divulgare d
- Bob trasmette (N, e) ad Alice

Descrizione di RSA

Come crea Bob la chiave pubblica e privata?

- Bob sceglie due numeri primi p e q, molto grandi e li moltiplica $N = p \cdot q$
- Bob sceglie un numero e, coprimo con N e più piccolo di $\varphi(N) = (p-1) \cdot (q-1)$
- Bob calcola d tale che
  $e \cdot d \equiv 1 \mod (p-1) \cdot (q-1)$
  d è l'inverso moltiplicativo di e
- solo Bob conosce la chiave segreta (N, d) mentre (N, e) è pubblica

Descrizione di RSA

Come avviene la cifratura e decifratura?

- Alice calcola $c = m^e \mod N$
- Alice trasmette c a Bob
- Bob riceve c e lo decripta calcolando:

$$c^d \equiv m \mod N$$

Infatti: $c^d \equiv m^{e \cdot d} \equiv m^{1+\varphi(N)} \equiv m^1 \cdot m^{\varphi(N)} \equiv m \mod N$

Descrizione di RSA

La forza di questo algoritmo sta nel fatto che calcolare d, conoscendo la chiave (N, e), è un problema difficile.

Questo non esclude che ci sia un modo semplice di calcolarlo, ma al momento non si conosce.

Nella sezione relativa alla sicurezza di RSA approfondiremo quest'aspetto.

Descrizione di RSA

Esempio

Bob sceglie $p = 3$ *e* $q = 11$. *Allora* $N = p \cdot q = 33$ *e* $\varphi(N)$.
Bob sceglie ad esempio $e = 7$ *che è coprimo con* 33 *e calcola* d *tale che* $d \cdot e \equiv 1 \mod \varphi(N)$.

La chiave pubblica è $(33, 7)$, *quella privata* $(33, 3)$.
Se Alice vuole trasmettere il messaggio $m = 17$:

$$\xrightarrow{c \equiv m^e \equiv 17^7}$$

A questo punto Bob calcola:

$$c^d \equiv 17^{7^3} \equiv 17^{21} \equiv 17^{1+20} \equiv 17 \equiv m \mod 33$$

3.ATTACCHI AD RSA

Attacchi ad RSA

Ci sono 2 strategie generali per attaccare l'algoritmo RSA:

1. trovare un algoritmo che dato un messaggio cifrato c e la chiave pubblica (N, e) restituisce il messaggio decifrato m senza però recuperare la chiave privata (N, d).
   Nessuno ne ha mai trovato un'istanza significativa
2. ricavare la chiave privata (N, d) da quella pubblica (N, e)

Attacchi ad RSA

Come si può ricavare la chiave privata da quella pubblica?

Apparentemente ci sono 3 alternative:

1. fattorizzare N
2. Calcolare $\varphi(N)$
3. ricavare il valore d direttamente dalla chiave pubblica (N, e)

Si può facilmente dimostrare che queste tre metodi sono equivalenti:

$$(1) \Leftrightarrow (2) \Leftrightarrow (3)$$

Attacchi ad RSA - Chiavi Deboli

In alcuni casi particolari è facile rompere il sistema, per via del modo in cui la chiave è stata malamente scelta (CHIAVI DEBOLI).

Questo succede ad esempio:

- se m ed e sono così piccoli che $m^e < N$; allora risulta facile trovare la radice e-esima di c, poichè $c = m^e$
- se $p - q < N^{\frac{1}{4}}$, ci sono algoritmi veloci per trovare p e q
- se $p - 1$ o $q - 1$ hanno solo fattori piccoli, N si fattorizza velocemente con l'algoritmo di Pollard $p - 1$

Purtroppo molte implementazioni dell'algoritmo RSA non controllano l'eventualità di avere una chiave debole.

Attacchi ad RSA - Fattorizzare N

Come possiamo fattorizzare un intero N?

Sono svariati gli algoritmi per fattorizzare un numero N, più o meno costosi computazionalmente.

Per interi con una *struttura* particolare, esistono algoritmi relativamente veloci.

Attacchi ad RSA - Fattorizzare N

Riportiamo un elenco dei più efficienti algoritmi per fattorizzare un generico numero intero:

- Fattorizzazione di Dixon
- Fattorizzazione delle Frazioni Continue (CFRAC)
- Crivello Quadratico - Quadratic sieve
- Shanks' square forms factorization (SQUFOF)
- General Number Field Sieve (GNFS)

Attacchi ad RSA - GNFS

Questo algoritmo riduce il problema di fattorizzare N a risolvere un enorme sistema di equazioni lineari.

Sono quindi due i passi necessari all'algoritmo:

- costruire il sistema partendo da N
- risolvere il sistema lineare, ad esempio con il metodo di *Eliminazione di Gauss* (ma Strassen è il più efficace se la matrice è molto grande)

Nota: Non è in grando di fattorizzare numeri che sono potenza di primi, che può però essere fatto velocemente in altri modi.

Attacchi ad RSA - GNFS

Questo algoritmo è il più efficace al giorno d'oggi per fattorizzare numeri con più di 100 cifre.

La sua complessità è:

$$e^{(\log(N))^{\frac{1}{3}}(\log\log(N))^{\frac{2}{3}}}$$

dove $\log(N)$ corrisponde al numero di cifre binarie necessarie per rappresentare N.
La complessità è quindi subesponenziale rispetto ad N.

Attacchi ad RSA - GNFS

Notiamo che nonostante quello che si possa pensare, la fattorizzazione di N non è un problema NP-hard.

Ovvero, trovare un algoritmo polinomiale che fattorizza gli interi non porta sconvolgimenti nella teoria della complessità.

4. SICUREZZA DI RSA

Sicurezza di RSA

Usando l'algoritmo GNFS si riescono a fattorizzare numeri N dell'ordine di $2^{768} \sim 10^{231}$.

Spendendo un milione di euro per dotarsi di un cluster potente, si potrebbe rompere una chiave RSA-768 (cioè N è dell'ordine di 2^{768}) ogni 6 mesi (in media).

Per cui chiavi RSA-1024 o superiori sono al momento fuori dalla portata di un attacco pratico.

Sicurezza di RSA

Un altro problema legato alla sicurezza è il modo in cui i parametri del crittosistema vengono generati.

- Bob deve poter scegliere due numeri primi p e q, almeno *pseudo-primi*, molto grandi
- serve la casualità dei parametri, quindi una buona sorgente di randomicità
- cercare di evitare possibili chiavi deboli