Elementi di crittografia: La crittografia asimmetrica

Nell'articolo precedente abbiamo visto come la crittografia simmetrica abbia occupato gran parte della storia di questa disciplina. Nell'epoca moderna si é fatta però sempre più pressante la necessità di sviluppare un sistema che permettesse di andare oltre all'unica chiave per codificare e decodificare il messaggio.
Viene così sviluppata la crittografia RSA (dal nome dei suoi tre inventori: Ronald **R**ivest, Adi **S**hamir e Leonard **A**dleman).

RSA: due chiavi ma diverse

Immagina che il nostro baule nel quale metteremo il messaggio abbia una serratura e due chiavi, la chiave A e la chiave B.
Se utilizzo la chiave A per chiudere la serratura, dovrò utilizzare la chiave B per aprirla, mentre se utilizzo la chiave B per chiuderla dovrò utilizzare la chiave A per aprirla.
A questo punto, se stabiliamo che la chiave A é privata (solo io posso utilizzarla) e la chiave B é pubblica (chiunque può utilizzarla) siamo molto vicini al concetto della cifratura RSA.
Prima di vederla nella pratica, farò un breve (ed estremamente semplificato) accenno al suo funzionamento.

I numeri primi, certo, proprio loro

Credo tutti sappiano che i numeri primi hanno una curiosa caratteristica che é quella di essere divisibili unicamente per 1 o per se stessi. Il loro studio é importantissimo nella matematica avanzata e nella fisica. A noi ora interessano in quanto sono alla base della crittografia RSA, il cui

funzionamento semplificato é il seguente.

Si prendono due numeri primi molto grandi (diciamo un centinaio di cifre) e si moltiplicano tra loro. **Il numero che ne risulterà sarà la chiave pubblica**. Potrà quindi esserti inviata “in chiaro” al momento che esegui una transazione con la carta di credito ad esempio. I dati della tua transazione saranno codificati utilizzando questo numero.

Ma come, tutti lo possono vedere!

Già, ma questo non é un problema, in quanto **per la decodifica dovremo disporre dei due numeri primi che moltiplicati tra loro originano quel numero** (e questa sarà la chiave privata).

Ma é così difficile trovare questi due numeri? Ad esempio per tentativi non é possibile?

E’ teoricamente possibile, ma la sicurezza di questa tecnica é data dal fatto che fattorializzare un numero enorme in numeri primi é un problema computazionalmente improponibile.
Anche disponendo di algoritmi di ricerca molto avanzati e di un mainframe potentissimo, **ci vorrebbero molti anni per risolvere il problema** (inoltre le chiavi vengono cambiate periodicamente).
Dunque RSA é sicuro in quanto la violazione della chiave é una procedura **troppo complessa e troppo lunga per poter essere attuata e quindi possibile solo in teoria.**

RSA per tutti

Chiaramente per il singolo utente questo sistema può sembrare un po’ fuori portata, e invece no. Esistono molti sistemi che implementano la teconologia RSA per gli utenti finali, il più diffuso dei quali é PGP.
Grazie al software PGP si possono creare la chiave pubblica e la chiave privata e codificare e decodificare i messaggi.

Vediamo dunque di capire come funziona. Poniamo di voler inviare un messaggio confidenziale a Luigi.
Non dovrò fare altro che mettere il messaggio nel solito baule e prendere la **chiave pubblica di Luigi** per chiuderlo. Ora il baule potrà essere aperto unicamente con la chiave privata di Luigi, quindi Luigi sarà l’unico che può accedere al contenuto del baule. La **confidenzialità** é garantita ed il gioco é fatto; o quasi...
Il problema è che chiunque può **chiudere il baule con la chiave pubblica di Luigi,** e dunque non é garantita l’**autenticità**. Luigi non può essere certo che sia stato veramente io ad inviargli il messaggio.

Facciamo in questo modo allora.

Prendo il messaggio, lo metto nel baule, e chiudo il baule con la **mia chiave privata**. Luigi aprirà il baule con la **mia chiave pubblica**. Se il baule si apre, significa che é stato chiuso con la mia chiave privata dunque posso essere solo io il mittente. L'**autenticità** é garantita ed il gioco é fatto; o quasi...
Il problema è che **chiunque** può aprire il baule con la mia chiave pubblica, quindi il messaggio non sarà più **confidenziale.**

Ma come si fa a garantire sia la confidenzialità che l'autenticità?

I più svegli lo avranno già capito.

- Metto il messaggio nel baule e lo chiudo con la **chiave pubblica di Luigi.**
- Metto il baule in un altro baule e lo chiudo con la **mia chiave privata.**
- Luigi aprirà il primo baule con la **mia chiave pubblica.**
- Siccome l'ho chiuso con la mia chiave privata si aprirà (l'**autenticità** é garantita).
- All'interno troverà l'altro baule che aprirà con la **sua chiave privata.**
- Siccome l'ho chiuso con la sua chiave pubblica si aprirà (la **confidenzialità** é garantita).

Naturalmente é un software che gestisce tutte queste operazioni, ci mancherebbe!

Ed ora il cerchio si chiude

Se vogliamo andare fino in fondo, ti svelo un segreto. La codifica con RSA è molto dispendiosa dal punto di vista computazionale ed aumenta esponenzialmente con l'aumentare delle dimensioni del messaggio.
Per ovviare a questo problema il messaggio in realtà è **cifrato con un algoritmo di tipo simmetrico**, ed RSA viene utilizzato **unicamente per cifrare la chiave di questo algoritmo**; geniale vero?
E che dire dell'integrità?
E' garantita da un controllo di integrità realizzato con un algoritmo di hashing (come ad esempio per il download di file).
Ho scritto una marea di cose per arrivare alla conclusione che la crittografia è come il maiale: non si butta nulla!
E tu? Utilizzi dei sistemi di crittografia asimmetrica? In che circostanza?