DOI:10.1478/C1C0601001

Atti dell'Accademia Peloritana dei Pericolanti
Classe di Scienze Fisiche, Matematiche e Naturali
Vol. LXXXIV, C1C0601001 (2006)

RSA: DAGLI ANNI '70 AL GIORNO D'OGGI

SILVIA PELLEGRINI

(Conferenza del 9 febbraio 2006)

ABSTRACT. In this note, after a brief historical excursus, we describe the revolution in cryptography of the 70es, due to the introduction of public key cryptography and the RSA algorithm. We also consider some contemporary applications of this algorithm. Modern cryptography has indeed become a commodity technology, which is currently used for applications ranging from e–commerce to digital television.

1. Introduzione

La **crittologia** è l'arte di registrare i pensieri di qualcuno in modo da renderli illeggibili a chiunque altro. Consente a due persone di entrare in contatto con la garanzia di assoluta segretezza, almeno in teoria. La parola **crittologia** deriva dal greco $\kappa\rho\upsilon\pi\tau\grave{o}\varsigma$ (nascosto) e $\lambda\acute{o}\gamma o\varsigma$ (parola) e comprende la **crittografia**, che riguarda i metodi che assicurano sicurezza, segretezza, autenticità e la **crittoanalisi**, che si occupa dei metodi per rompere i messaggi crittati o per manometterli prima che giungano a destinazione. Il problema di crittare e decrittare un messaggio può essere considerato parallelo al procedimento di codifica e decodifica. Anche in questo caso c'è chi vuole trasmettere un messaggio a qualcuno attraverso un canale di comunicazione non affidabile, ma qui l'inaffidabilità del canale è causata da un intercettatore non autorizzato che vuole conoscere il messaggio trasmesso oppure che vuole soltanto alterarlo. Mentre l'obiettivo della teoria dei codici è salvaguardare l'esattezza del messaggio trasmesso, correggendo gli errori che possono essere entrati casualmente durante la trasmissione per un *disturbo* del canale, l'obiettivo della crittologia è mascherare il messaggio originale in modo tale che un intercettatore non autorizzato non possa venirne a conoscenza. Un sistema crittografico funziona quando garantisce che il messaggio inviato resterà segreto e non verrà alterato, cioè quando garantisce *segretezza* e *autenticità*. I sistemi utilizzati per crittare un messaggio sono sostanzialmente due: **sistemi simmetrici** e **sistemi asimmetrici** detti anche **sistemi a chiave pubblica**. I primi sono a chiave singola, nel senso che la chiave usata per crittare è la stessa usata per decrittare, in quelli asimmetrici, invece, la chiave usata per crittare non coincide con quella usata per decrittare.

2. Un pò di storia...

La storia della crittologia è strettamente legata alla storia dell'uomo: nel corso dei secoli re e generali compresero che per governare i propri sudditi o comandare i propri eserciti sarebbero state necessarie comunicazioni sicure. Incentivarono, quindi, lo sviluppo di tecniche che potessero alterare i messaggi in modo da renderli comprensibili solamente a persone autorizzate e, nello stesso tempo, equipe di specialisti tentavano di far breccia nei messaggi criptati intercettati per comprenderne i contenuti. Questa battaglia tra inventori e solutori di codici influenzò profondamente la storia dell'umanità e produsse notevoli progressi scientifici. È del V secolo a.C. il primo esempio di crittografia della storia. Narra Plutarco che il governo spartano comunicasse con i propri generali mediante uno stratagemma. Su un pezzo di legno cilindrico (*scitale*) veniva avvolta una striscia di papiro. Il messaggio veniva scritto trasversalmente, poi il papiro veniva inviato e solamente chi era in possesso di un cilindretto di uguale diametro era in grado di comprendere quanto era scritto sul papiro. La crittologia nasce quindi in ambito militare. Lo storico greco Polibio dà notizia di un crittosistema che ha, tra le sue caratteristiche, la conversione delle lettere in numeri. Da Svetonio sappiamo che anche Giulio Cesare scriveva crittato: semplicemente sciftando di 3 lettere le lettere del messaggio. Alla A sostituiva la D, alla B la E e così via. La *chiave* utilizzata era quindi $k : m \longrightarrow 3 + m$. Per decrittare veniva utilizzata la stessa chiave, ovviamente retrocedendo di tre lettere. Quello di Cesare è quindi un sistema crittografico simmetrico.
In generale, se A è l'alfabeto del testo in chiaro e B è l'alfabeto del testo cifrato *una corrispondenza biunivoca (o almeno iniettiva)* $f : A \longrightarrow B$ *è detta* **chiave**.
La chiave è quindi una particolare scelta per legare tra loro l'alfabeto del testo in chiaro e l'alfabeto crittante. Il procedimento generale, di tipo matematico, architettato per crittare prende il nome di **algoritmo**. Ciò che è importante è che resti segreta la chiave, mentre l'algoritmo può anche essere reso pubblico. La separazione concettuale tra *chiave* e *algoritmo* è uno dei più saldi principi della crittologia, formulato in modo definitivo nel 1883 dal linguista olandese Auguste Kerckhoffs von Nieuwenhof nel trattato "*La Cryptographie Militaire*". La **Legge di Kerckhoffs** afferma infatti che *la sicureza di un crittosistema non deve dipendere dal tener celato il critto-algoritmo. La sicurezza dipenderà solo dal tener celata la chiave.*
Il mondo dei crittografi è un mondo affascinante. La storia della crittologia, raccontata in modo semplice e intrigante, si può per esempio reperire in [1].

3. Anni '70: DES

Per poter descrivere come nacque RSA, il più famoso crittosistema a chiave pubblica, cercherò, sia pure brevemente, di delineare l'ambiente in cui si trovarono a lavorare, nei primi anni '70, gli scienziati che, in quel periodo, si occupavano di crittologia.
Negli anni '60 il ministero della Difesa americano aveva iniziato a finanziare un'organizzazione chiamata ARPA (Advanced Research Projects Agency) che aveva tra i propri obiettivi quello di trovare il modo di collegare tra loro computer militari a grande distanza tra loro. L'ARPAnet, cioè il network di comunicazioni dell'ARPA fu inaugurato nel 1969, continuò a crescere e nel 1982 generò Internet. Gli anni '60 furono gli anni in cui i calcolatori divennero sempre meno costosi e sempre più potenti. Sempre più numerose

erano le aziende che potevano permettersi di utilizzare metodi crittografici per trasmettere dati in tutta segretezza. Tutto ciò creava un nuovo problema. Se un’azienda doveva inviare informazioni crittate a un’altra azienda doveva essere certa che il destinatario utilizzasse lo stesso sistema di crittaggio. Occorreva quindi implementare un software *standard*, accessibile a tutti. Il 15 Maggio 1973 il National Bureau of Standards americano invitò a presentare proposte di un sistema crittografico che potesse venire ufficialmente e universalmente adottato. Il più apprezzato algoritmo era allora Lucifer, sviluppato agli inizi degli anni ’70 da Horst Feistel che, fuggito dalla Germania nel 1934, lavorava all’IBM. La NSA (National Security Agency), l’ente che si occupava della sicurezza delle comunicazioni del governo e dei comandi militari, decise di adottare la cifratura Lucifer sia pure con qualche variazione. L’agenzia pretese che il numero di chiavi venisse limitato, che non si superasse il numero di cento milioni di miliardi. DES (Data Encryption Standard) è il nome dato alla versione a 56 bit della cifratura Lucifer. 56 era infatti il numero di 0 e 1 necessari per esprimere la chiave in forma binaria. In questo modo l’Agenzia riteneva che questa quantità di chiavi consentisse una trasmissione sicura in ambito civile, ove i calcolatori non erano di straodinaria potenza e, nello stesso tempo, era un numero di chiavi da esaminare non proibitivo per i calcolatori di cui essa stessa disponeva. Un numero superiore di chiavi avrebbe finito con impedirle il controllo delle informazioni trasmesse. Il DES venne pubblicato nel 1975. Questo “*algoritmo di codifica per la protezione dei dati elettronici*” opera su blocchi di dati di 64 bits tramite una chiave di 56 bits. La descrizione di DES, peraltro alquanto complessa, esula da questo contesto ed è comunque reperibile in [2].

4. Distribuzione delle chiavi

Il DES, nonostante la sua grande forza non risolveva però il problema noto come il problema della *distribuzione delle chiavi*. Supponiamo di voler trasmettere un messaggio. Si può criptare con DES ma, al destinatario, oltre al software necessario come possiamo trasmettere la chiave? Non certo per telefono, o per posta. Il metodo più sicuro sembrerebbe la consegna *brevi manu* o tramite corriere. Fu infatti questo il metodo che negli anni ’70 gli Istituti di Credito adottarono per distribuire le chiavi alla clientela. Migliaia di funzionari viaggiavano in continuazione con valigette blindate e la crescente espansione del mercato fece diventare questo metodo, dai costi spropositati, un vero incubo. Il problema della distribuzione delle chiavi non era certamente nuovo. Durante la seconda guerra mondiale costituì uno dei grossi problemi con cui si dovettero confrontare i comandanti tedeschi. Giornalmente dovevano comunicare a tutti gli operatori Enigma le chiavi utilizzate. Gli U-boot che restavano lontani dalle basi per lunghi periodi erano costretti a imbarcare le chiavi in quantità necessaria e, se venivano catturati, finivano con il consentire al nemico una facile decodifica di messaggi di importanza vitale. Il problema era però anche economico. Se i costi della distribuzione delle chiavi potevano essere sostenuti e giustificati in ambito militare, doveveno essere drasticamente abbattuti da qualunque compagnia civile. Tre matematici americani della Stanford University compresero l’importanza di determinare un sistema agevole, che salvaguardasse la privacy di tutti coloro che avrebbero voluto comunicare per via elettronica e, a metà degli anni ’70, trovarono una brillante soluzione, una tecnica in grado di risolvere il problema alla base. Whitfield Diffie, Martin Hellman e Ralph Merkle certamente conoscevano una storiella che all’epoca

circolava. Supponiamo che Alice voglia inviare un messaggio personale a Bob. Lo colloca in una scatola e la chiude con un lucchetto. Quando Bob riceve la scatola applica un altro lucchetto e rinvia la scatola ad Alice. Alice rimuove il proprio lucchetto e rinvia la scatola a Bob che può facilmente aprire la scatola e leggere il messaggio. Questo racconto, letto in chiave crittografica, sembrerebbe risolvere il problema della distribuzione delle chiavi. Alice critta il messaggio, Bob lo critta a sua volta, Alice decritta e Bob decritta a sua volta e legge il messaggio. Ma i sistemi di crittaggio non funzionano affatto come i lucchetti sulle scatole. In crittografia, la cifratura eseguita per ultima deve sempre essere decifrata per prima. Non si può prescindere dall'ordine nemmeno in una cifratura monoalfabetica. Consideriamo il seguente esempio:

chiave di Alice

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
f m r z d j u p e w v a q g x b s y k i t c n l o h
```

chiave di Bob

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
g t y d o p v e x z a u f k q l b w h r m c j s n i
```

CI VEDIAMO DA LUCA

Cifrato da Alice: RE CDZEFQX ZF ATRF
Cifrato da Bob: WO YDIOPBS IP GRWP
Decifrato da Alice: JY RETYHPQ TH NCJH
Decifrato da Bob: WC THBCSFO BS YVWS

Il risultato è senza senso, mentre si può facilmente verificare che, se il messaggio venisse prima decifrato da Bob e poi da Alice, sarebbe il messaggio originale. La situazione proposta dai lucchetti è quindi profondamente diversa da quanto accade nei sistemi di crittaggio e, in generale, in matematica. Ciò che quindi i tre matematici cominciarono a cercare era una funzione che si comportasse come i lucchetti. L'idea venne ad Hellman ed è di una semplicità disarmante. Si basa sull'aritmetica modulare. Alice e Bob concordano una funzione del tipo $n^x (mod\, p)$ ove p è un numero primo e $n < p$. Qui p è un grande numero primo, ma noi, per descrivere questo algoritmo, fissiamo per esempio

$$6^x \ mod\ 13$$

I passi qui di seguito consentono di verificare che, con questo metodo, Bob e Alice possono scambiarsi la chiave in tutta segretezza.

(i) Alice sceglie un numero A e lo tiene segreto: per esempio $A = 4$
(ii) Alice calcola $\alpha = 6^A = 6^4 \equiv_{13} 9$
(iii) Alice invia $\alpha = 9$ a Bob
(i') Bob sceglie un numero B e lo tiene segreto: per esempio $B = 5$
(ii') Bob calcola $\beta = 6^B = 6^5 \equiv_{13} 2$
(iii') Bob invia $\beta = 2$ ad Alice
(iv) Alice riceve $\beta = 2$ da Bob, eleva β al suo numero segreto $A = 4$ e ottiene $\beta^A = 2^4 \equiv_{13} \mathbf{3}$
(iv') Bob riceve $\alpha = 9$ da Alice, eleva α al suo numero segreto $B = 5$ e ottiene $\alpha^B = 9^5 \equiv_{13} \mathbf{3}$

Bob e Alice hanno ottenuto lo stesso numero: il 3, e questo perchè, molto semplicemente $(n^B)^A$ (mod p) coincide con $(n^A)^B$ (mod p).
Questo numero, il **3**, così ottenuto, è la chiave. Chiunque fosse stato in ascolto quando Bob e Alice si scambiavano i dati iniziali $n = 6$ e $p = 13$, oppure i numeri $\alpha = 9$ e $\beta = 2$, non sarebbe stato in grado di calcolare la chiave perché i numeri $A = 4$ e $B = 5$ Bob e Alice non se li sono scambiati e risalire ad A o a B conoscendo il resto della divisione per p non è possibile. Ora che senza incontri segreti Bob e Alice sono in possesso di una chiave possono pensare di crittare il loro messaggio, per esempio, in DES. Diffie, Hellman e Merkle presentarono questi risultati alla National Computer Conference nel 1976 e iniziarono le pratiche per il brevetto. Fu ancora Diffie che intuì la possibilità di una cifratura asimmetrica. L'idea era la seguente: *ogni utente deve possedere una chiave privata e una chiave pubblica, nota a tutti e reperibile per esempio in un registro. Chi vorrà inviare un messaggio lo cripterà con la chiave pubblica del destinatario e soltanto lui, utilizzando la sua chiave privata, potrà decriptarlo.* Si trattava di un'idea generale che, qualora fosse stata effettivamente realizzata, avrebbe avuto enormi conseguenze. Diffie la rese pubblica nel 1975 e da quel momento i crittologi si misero all'opera per trovare una funzione che si adattasse a questo scopo. Diffie, Hellman e Merkle entro il 1976 avevano rivoluzionato il mondo della crittografia: avevano risolto il problema della distribuzione delle chiavi e avevano introdotto il concetto di cifratura asimmetrica o a chiave pubblica. Ma non furono loro a realizzare un efficace sistema di crittaggio asimmetrico.

5. L'algoritmo RSA

Certamente stimolati dai matematici di Stanford, Ron Rivest, Adi Shamir e Leonard Adleman, ricercatori del MIT trovarono una funzione basata sul concetto di modulo che risolveva il problema della cifratura asimmetrica proposto. Mentre nelle tecniche di scrittura *simmetriche*, decifrare significa semplicemente eseguire al contrario il procedimento di cifratura, in un sistema *asimmetrico*, la chiave usata per crittare non coincide con la chiave usata per decrittare. La stessa cifratura DES, per quanto complessa, è simmetrica. I 16 passaggi che consentono di crittare il messaggio, se eseguiti al contrario lo decrittano. Il procedimento ideato da Rivest, Shamir e Adleman si basa sulla funzione e sul Teorema che Eulero dimostrò nel diciottesimo secolo, secondo il quale se a è primo con N

$$a^{\phi(N)} \equiv_N 1$$

Qui ϕ è la funzione di Eulero che associa ad ogni numero naturale N diverso da 1 il numero dei numeri primi con N e minori di N

$$\begin{array}{rcccc} \phi & : & \mathbb{N} \setminus \{1\} & \longrightarrow & \mathbb{N} \\ & & N & \longrightarrow & \phi(N) \end{array}$$

Inoltre, se p e q sono numeri primi, ovviamente, $\phi(p) = p - 1$ e, si dimostra agevolmente che $\phi(pq) = \phi(p)\phi(q) = (p-1)(q-1)$.
Descrizione dell'algoritmo
(i) Alice sceglie due grandi numeri primi p e q e ne calcola il prodotto $N = pq$
(ii) Alice sceglie, random, un numero e tale che $MCD(e, \phi(N)) = 1$

la chiave pubblica di Alice è $\{N, e\}$

(iii) solamente Alice, che conosce $\phi(N) = (p-1)(q-1)$ è in grado di risolvere l'equazione congruenziale $ex \equiv_{\phi(N)} 1$

la chiave privata di Alice è l'unica soluzione d, di questa equazione

Ora supponiamo che Bob voglia inviare un messaggio ad Alice.
(i') Bob trasforma in un numero M il messaggio. (Utizza per esempio i caratteri ASCII)
(ii') Bob si procura le chiavi pubbliche di Alice $\{N, e\}$ e invia $C = M^e(modN)$
(iii') Alice riceve C e lo eleva a d, la chiave privata che solamente lei conosce, ritrovando (modulo N) il messaggio originale M. Infatti

$$C^d \equiv_N M$$

Non ci resta che dimostrare che

Date le chiavi pubbliche $\{e, N\}$ e la chiave privata $\{d\}$, in un crittosistema RSA, ogni messaggio M è tale che $M^{ed} \equiv_N M$

Infatti $ed \equiv_{\phi(N)} 1$ e quindi $M^{ed} = M^{1+k\phi(N)} = M(M^{\phi(N)})^k$. Il teorema di Eulero afferma che, quando a è primo con N, $a^{\phi(N)} \equiv_N 1$ e, in particolare, quando p è un numero primo, $a^{(p-1)} \equiv_p 1$. Ora, $M^{ed} = M(M^{\phi(N)})^k = M(M^{(p-1)(q-1)})^k$, e ovviamente, $M(M^{(p-1)(q-1)})^k$ è congruo ad M sia modulo p che modulo q e quindi

$$M^{ed} \equiv_N M$$

Se il messaggio di Bob dovesse essere intercettato, per poterlo decifrare bisognerebbe poter risalire ai numeri primi p e q conoscendone il prodotto pq. La forza di RSA è determinata dal fatto che, quando i numeri p e q sono molto grandi è difficile riottenerli in tempi contenuti conoscendone il loro prodotto N. Allo stato attuale della conoscenza i procedimenti che consentono la scomposizione di un numero in fattori primi comportano che il numero in questione venga diviso successivamente per una serie di numeri primi, controllando se tali divisioni danno un resto. Questi metodi richiedono ovviamente tempi molto lunghi quando il numero da esaminare è molto grande. Il livello di sicurezza di RSA dipende dalla grandezza dei numeri primi scelti: per esempio, per certe operazioni bancarie, si tende a scegliere numeri N dell'ordine di 10^{308}. Se in futuro la matematica dovesse offrire un metodo di scomposizione in fattori primi più veloce di quelli ora noti è ovvio che questo potrà incrinare la sicurezza di RSA, ma la possibilità di decomporre rapidamente un numero in fattori primi viene ricercata da duemila anni e non sono stati fatti passi significativi.
L'algoritmo RSA venne immediatamente brevettato, e, come vedremo, anche oggi, questo crittosistema è largamente utilizzato.
Per comprendere meglio come opera RSA, costruiamo un esempio scegliendo numeri primi piccoli.
Supponiamo che Alice abbia scelto $p = 11$ e $q = 17$. Rende noto il loro prodotto $N = pq = 187$ e sceglie e rende noto un numero e che deve essere primo con $(p-1)(q-1) = 160 = \phi(N)$. Per esempio sceglie $e = 9$. Le chiavi pubbliche di Alice sono quindi

{**187, 9**}. Solamente Alice, che conosce $\phi(N) = 160$, è in grado di trovare, attraverso l'algoritmo di Euclide, l'unico numero d che risolve l'equazione congruenziale $9x \equiv_{160} 1$. La soluzione **89** è la sua chiave privata.
Ora Bob vuole inviare ad Alice un messaggio, quindi anzitutto lo trasforma in numeri. Nella seguente tabella sono riportati i numeri binari ASCII delle lettere maiuscole, la notazione binaria e la conseguente notazione decimale.

A	1	0	0	0	0	0	1	65	N	1	0	0	1	1	1	0	78
B	1	0	0	0	0	0	1	0	66	O	1	0	0	1	1	1	1	79
C	1	0	0	0	0	0	1	1	67	P	1	0	1	0	0	0	0	80
D	1	0	0	0	0	1	0	0	68	Q	1	0	1	0	0	0	1	81
E	1	0	0	0	0	1	0	1	69	R	1	0	1	0	0	1	0	82
F	1	0	0	0	0	1	1	0	70	S	1	0	1	0	0	1	1	83
G	1	0	0	0	0	1	1	1	71	T	1	0	1	0	1	0	0	84
H	1	0	0	0	1	0	0	0	72	U	1	0	1	0	1	0	1	85
I	1	0	0	0	1	0	0	1	73	V	1	0	1	0	1	1	0	86
J	1	0	0	0	1	0	1	1	74	W	1	0	1	0	1	1	1	87
K	1	0	0	0	1	0	1	1	75	X	1	0	1	1	0	0	0	88
L	1	0	0	0	1	1	0	0	76	Y	1	0	1	1	0	0	1	89
M	1	0	0	0	1	1	0	1	77	Z	1	0	1	1	0	1	0	90

Per esempio $1\,0\,1\,1\,0\,1\,0 = 2^6 + 0 + 2^4 + 2^3 + 0 + 2^1 + 0 = 64 + 16 + 8 + 2 = 90$

Se quindi Bob vuole inviare ad Alice la parola "KISS" la trasforma nella sequenza di numeri $75\ 73\ 83\ 83$, cripta elevando ciascuna cifra a 9 e riduce il risultato (mod 187).

$$75^9 \equiv_{187} 27,\ 73^9 \equiv_{187} 63,\ 83^9 \equiv_{187} 134$$

Quindi Bob invia $27\ 63\ 134\ 134$

Alice eleva la sequenza ricevuta alla sua chiave privata, calcola 27^{89} che, modulo 187, è 75 cioè la lettera K, 63^{89} che, modulo 187, è 73 cioè la lettera I e infine 134^{89} che, modulo 187, è 83 cioè la lettera S. Alice ritrova così il messaggio originale.

6. PGP

L'RSA venne commercializzato nel 1977 ma comportava una potenza di calcolo che solamente i militari, il governo e società di grandi dimensioni si potevano permettere. In quegli anni si era già entrati nell'era dell'informazione e se nel passato la crittografia era stata utilizzata per effettuare comunicazioni sicure in ambito prevalentemente militare, si rendeva necessario utilizzarla per garantire autenticità e sicurezza in ambito civile. Il primo che comprese che la crittografia avrebbe potuto realizzare il diritto alla privacy e alla libertà di parola fu Phil Zimmermann. Iniziò così a mettere a punto programmi che richiedessero risorse reperibili in un normale personal computer. Chiamò il suo progetto Pretty Good Privacy (riservatezza niente male): PGP. Dato che l'RSA comportava un notevole lavoro per il calcolatore, Zimmermann combinò nel PGP le due tecniche: quella simmetrica e quella asimmetrica. Decise di crittare il messaggio con un sistema simmetrico, simile al

DES, chiamato IDEA e utilizzò l'RSA, asimmetrico e quindi più lento per trasmettere soltanto la chiave. Il software ideato da Zimmermann prevedeva tra l'altro che la scelta dei grandi numeri primi necessari per crittare la chiave, fossero facilmente trovati muovendo casualmente il mouse e proprio questa casualità garantiva l'unicità delle chiavi pubblica e privata. Inoltre, il PGP introduceva la firma digitale, risolvendo in questo modo il problema dell'autenticità del messaggio. Immaginiamo che un cliente voglia inviare alla propria banca un ordine di vendita/acquisto o di trasferimento di capitali. Il cliente critta il messaggio con la propria chiave privata: in questo modo è come se vi apponesse la propria firma. Soltanto lui infatti è in possesso di questa chiave. Quando la banca riceverà il messaggio sarà certa della provenienza se riuscirà a renderlo in chiaro utilizzando la chiave pubblica di quel cliente. Si tratta evidentemente di una rilettura del principio che è alla base dell'invenzione di Diffie ed Hellmann sulla distribuzione delle chiavi. Il PGP così congeniato non rese facile la vita di Zimmermann: utilizzava infatti la cifratura RSA, che era stata brevettata, ma soprattutto il Senato degli Stati Uniti aveva approvato una legge che rendeva illegali i sistemi crittografici troppo sicuri, sistemi, cioè, che impedivano al Governo di volgere in chiaro in qualunque momento i messaggi crittati intercettati. Nel Giugno 1991 Zimmermann forzò la mano: mise il PGP in Internet. Da quel momento chiunque potè scaricarlo gratuitamente e ciò gli creò non pochi problemi. Venne considerato dalla RSA Data Security Inc., un pirata informatico, ma, quel che è peggio, venne accusato dal Governo di esportazione di materiale bellico, tale infatti era considerato il software crittografico, al pari dei cannoni e delle mitragliatrici. L'appoggio incondizionato della comunità scientifica evitò a Zimmermann guai peggiori e dopo alcuni anni di battaglie legali vide finalmente riconosciuta l'importanza di PGP.

7. RSA al giorno d'oggi

Vediamo ora come, inconsapevolmente, ciascuno di noi, ogni giorno, utilizza questi sistemi crittografici, in particolare RSA.

7.1. Connessioni protette: SSL.

Tutte le volte che effettuiamo una richiesta di tipo `https://` (secure hyper–text transfer protocol) stabiliamo una connessione *ssl* (Secure Socket Layer-strato di connessione sicura) [3]. Si tratta del protocollo crittografico più utilizzato al momento. Lo usiamo, infatti, quando effettuiamo operazioni bancarie oppure acquisti in rete quando, cioè, trsmettiamo via Internet documenti privati. Compito principale di *ssl* è assicurare l'identità del server, ci deve garantire di aver effettuato il collegamento con l'interlocutore voluto. Questa sicurezza viene ottenuta attraverso un sistema che impiega diversi metodi di autenticazione e di cifratura. Per quanto riguarda l'autenticazione viene utilizzata una rete di fiducia. Autorità centrali firmano i certificati da installare sui server e un browser, quando accede ad una pagina cifrata, verifica che il certificato del server sia legittimo. Se non lo è invia un messaggio di segnalazione all'utente. Per quanto riguarda la cifratura, *ssl* prevede che le pagine vengano inviate cifrate dal server al browser. Gli algoritmi di cifratura sono 3DES (DES applicato 3 volte) [4], AES (Advanced Encryption Standard) [5], RC4 e RC5 crittosistemi messi a punto recentemente da Rivest [6]. Si tratta di sistemi tutti simmetrici, la chiave di sessione, negoziata in modo casuale, è invece asimmetrica e ottenuta mediante RSA o Diffie-Hellmann.

7.2. Riservatezza: gpg.

Se vogliamo effettuare una conversazione protetta, in particolare se vogliamo cifrare la posta elettronica, possiamo utilizzare *gpg*. Si tratta della versione di PGP del progetto GNU (GNU privacy guard) [7]. Il Progetto GNU venne lanciato nel 1984 per sviluppare un sistema operativo Unix-compatibile completo che fosse software libero. Rispetto a *ssl* la rete di fiducia è decentralizzata, non esiste cioè un server centrale che garantisca univocamente l'identità di un soggetto, ma semplicemente dei "punteggi di fiducia" associati a diversi individui. In una conversazione protetta con *gpg* entrambe le parti si autenticano a vicenda ed entrambe utilizzano una chiave permanente. *gpg* non solo è usato per cifrare la posta elettronica e garantire riservatezza, ma vi sono applicazioni anche alla telefonia su rete (VOIP)(voice over internet protocol). Dal punto di vista tecnico/implementativo i protocolli utilizzati da *gpg* e quelli utilizzati da open*ssl* [8] sono simili anche se, solitamente, una chiave *gpg* è molto più lunga di una chiave *ssl* e questo per il fatto che una connessione *ssl* deve essere stabilita in tempo reale, mentre solitamente *gpg* è utilizzato per messaggi che possono essere decodificati con calma come appunto la posta elettronica.

7.3. Anonimità: tor e mixmaster.

Se invece vogliamo inviare messaggi senza essere individuati, abbiamo a disposizione sistemi di tipo *tor/mixmaster* [9, 10], che sono finalizzati a garantire l'anonimità della comunicazione, piuttosto che l'identità delle parti o la segretezza. Ciò viene implementato non solo usando una qualche forma di cifratura ma soprattutto costringendo i dati a passare per una serie di server distinti.
Se un messaggio deve andare $A \longrightarrow B$, l'idea è quella di farlo passare per un numero elevato di server intermedi E_1, E_2, E_3,... facendo in modo che ogni macchina nella catena conosca solamente quelle immediatamente precedente e successiva. Pertanto se, per esempio, il percorso del messaggio è

$$A \longrightarrow E_1 \longrightarrow E_2 \longrightarrow E_3 \longrightarrow B$$

il server E_1 conosce solamente il mittente A e la destinazione immediata E_2; E_2 conosce E_1 ed E_3 (ma non conosce né A né B); E_3 conosce E_2 e B (ma non conosce né A né E_1). L'operazione di occultamento viene effettuata utilizzando una cascata di codifiche a chiave pubblica da parte del mittente originario.
Se m è il messaggio, quanto effettivamente trasmesso da A al server E_1 è

$$c(c(c(c(m,b)+[perB],k_{e_3})+[perE_3],k_{e_2})+[perE_2],k_{e_1})$$

ove $k_{e_1}, k_{e_2}, k_{e_3}$ sono le chiavi pubbliche dei server E_1, E_2, E_3, mentre b è la chiave pubblica del destinatario B. Quando il messaggio arriva al server E_1, E_1 lo può decrittare con la propria chiave privata e inviare ad E_2 che, a sua volta, decritta con la propria chiave privata, e così via finché il messaggio arriva al destinatario B che lo rende in chiaro con la propria chiave privata. In questo tipo di architetture, il percorso che il messaggio deve seguire è fissato a priori dal mittente.
Nei dettagli: *Mixmaster* è una implementazione di questo tipo di algoritmo per la posta elettronica; dato che i messaggi di posta elettronica non sono necessariamente in tempo reale, i vari nodi attraversati dai messaggi possono anche alterare il percorso previsto. Per esempio, possono decidere di allungarlo aggiungendo un altro passaggio, oppure attendere

un tempo casuale prima di ritrasmettere un messaggio al fine di evitare che una analisi dei tempi di ingresso e uscita dei dati consenta di correlare A con B.
Tor, al contrario, è una implementazione del medesimo tipo di architettura per comunicazioni in tempo reale, quali per esempio il traffico web. È chiamata "onion routing", in quanto ogni server rimuove uno strato della cipolla crittografica. Il fatto che il sistema debba poter rispondere in tempo breve rende impossibile le operazioni di rimescolamento presenti nel caso di mixmaster e quindi, se da un lato *tor* è maggiormente funzionale rispetto a mixmaster, risulta più sensibile ad attacchi che si basino sull'analisi dei tempi di ingresso/uscita dei dati.

7.4. Uso diverso della crittografia: DVD e DVB.

Vediamo, per concludere, un differente uso della crittografia. Si tratta di quello utilizzato per i DVD (Digital Versatil Disc) [11] oppure per le trasmissioni video digitali DVB (Digital Video Broadcast) [12]. In questo caso la crittografia viene impiegata per limitare l'accesso ai dati ai soli sottoscrittori di un servizio. La differenza principale dallo scenario classico è che, in questo caso, il fruitore finale del prodotto, cioè l'utente, può aver interesse a divulgare i dati da lui recepiti ad altri, mentre i fornitori del servizio desiderano ovviamente evitare questo tipo di situazione. La crittografia viene quindi impiegata per limitare la ridistribuzione delle informazioni.
Il caso dei DVD è emblematico: il pianeta è diviso in 8 regioni
1. Stati Uniti e Canada
2. Giappone, Europa, Sud Africa, Medio Oriente, Gronelandia
3. Corea del Sud, Taiwan, Hong Kong
4. Australia, Nuova Zelanda, America Latina e Messico
5. Europa dell'est, Russia, India, Africa
6. Cina
7. Non specificato (perché riservata ad un cerchia particolare di persone. Per esempio i componenti la giuria dei premi Oscar).
8. Aerei e Navi

I dati vengono codificati con una chiave che ne limita l'utilizzo ad un sottoinsieme di queste regioni. Il motivo è quello di limitare l'accesso a prime visioni fuori dalle aree di origine e di instaurare delle possibili gabbie di prezzo. L'algoritmo dei DVD è basato su due chiavi private: una chiave *di disco* e una chiave *di titolo*. Ogni disco protetto è cifrato con una chiave *di disco* che, a sua volta, è immagazzinata sul disco stesso e cifrata mediante la chiave *di regione* detta *master key*. Un lettore, nel momento in cui un disco è inserito, cerca la chiave *di disco* in forma codificata e prova a decrittarla con la sua chiave *di regione*; se ci riesce procede poi a decrittare le chiavi *di titolo* corrispondenti ai vari spezzoni e visualizza tutto il film.
Diverso è il problema della TV digitale: in questo caso il problema è quello di controllare l'accesso al contenuto da parte di gruppi di numerosi utenti, che hanno, tra l'altro, una composizione variabile nel tempo. I protocolli utilizzati impiegano un misto di chiavi simmetriche e asimmetriche. Utenti e provider sono dotati di chiavi asimmetriche, mentre è simmetrica la chiave di sessione, cioè quella usata per crittare il film. Ad ogni utente viene distribuita una *Smart Card* che contiene una chiave asimmetrica privata. Questa chiave identifica univocamente l'utente e, per impedire che essa venga duplicata, viene

codificata con la chiave pubblica del provider. Il provider, a questo punto, trasmette il film codificato con una chiave di sessione (simmetrica). Quest’ultima viene codificata con tutte le chiavi pubbliche degli utenti abilitati a fruire della trasmissione. Il provider trasmette altresì la sua chiave privata per la decodifica della *Smart Card*. Ora che la *Smart Card* è decodificata, può comunicare al decoder la sua chiave privata e il decoder può risalire alla chiave di sessione. Una volta estratta la chiave di sessione questa viene usata per decodificare la trasmissione.

Riferimenti bibliografici

[1] Simon Singh *Codici e Segreti* (Rizzoli, Milano, 1999).
[2] http://www.itl.nist.gov/fipspubs/fip46-2.htm
[3] http://www.webopedia.com/TERM/S/SSL.html
[4] http://dnclab.berkeley.edu/ kenneth/courses/sims250/des.html
[5] http://en.wikipedia.org/wiki/Advanced Encryption Standard
[6] http://www.rsasecurity.com/rsalabs/node.asp?id=2250
[7] http://www.gnupg.org/
[8] http://www.openssl.org/
[9] http://www.tor.eff.org/
[10] http://www.mixmaster.sourceforge.net/
[11] http://www.dvdforum.org/forum.shtml
[12] http://www.dvb.org/

Silvia Pellegrini
Dipartimento di Matematica
Facoltà di Ingegneria
Università degli Studi di Brescia
Via Valotti, 9
25133 Brescia (IT)
E-mail: silvia.pellegrini@ing.unibs.it

Presented: February 9, 2006
Published on line on May 15, 2006