SICUREZZA E RISERVATEZZA GARANTITA DAL PROGETTO “PGP” (Pretty Good Privacy)

a cura di Anghinoni Ugo

Università di Pavia
Sede di Mantova
Facoltà di Ingegneria
Corso di Ingegneria Informatica

Note ed avvertenze

Tutte le informazioni della presente relazione sono state reperite in Internet e solo in piccola parte modificate per ragioni di sintassi o di impaginazione.

Non se garantisce pertanto la veridicità assoluta seppure siti diversi riportino, sostanzialmente, le medesime informazioni.

Non si garantisce neppure l'assenza totale di errori ed anzi si ringrazia anticipatamente chiunque, nello spirito di correggere e migliorare testo e contenuti, voglia indirizzare suggerimenti alla casella di mail ugoantin@tin.it ... magari cifrando il messaggio con PGP!

Indice generale

Capitolo 1 – La Crittografia

1.1 Introduzione alla Crittografia

La prima persona che usò la crittografia fu Giulio Cesare! Egli doveva inviare messaggi ma non si fidava dei messaggeri, cosi' inventò un metodo per codificare quei messaggi.
Solo il destinatario prestabilito - che conosceva il metodo per decodificare il messaggio - poteva leggerli.
Perciò, cos'è la crittografia?
La crittografia è l'arte che crea ed usa i **sistemi di crittografia**. Un sistema di crittografia e' un metodo per rendere illeggibili i messaggi, in modo da renderli decodificabili solo dal destinatario prestabilito. I sistemi di crittografia sono chiamati anche **sistemi di cifratura**. L' arte di scardinare i sistemi di cifratura e' chiamata **crittoanalisi.** Bene, la scienza che studia la crittografia e la crittoanalisi e' chiamata **crittologia** (dal greco *kryptos* , che significa 'nascosto' e logos, che significa 'discorso, parola'). Il messaggio originale e' chiamato **testo in chiaro**, ed il messaggio codificato e' chiamato **testo cifrato**. Quando codifichi un messaggio, usi una procedura che lo converte in testo cifrato. Questa procedura e' chiamata **cifratura**. Viceversa, quando vuoi rendere leggibile un messaggio, usi il procedimento opposto, chiamato **decifratura**.
La crittografia dunque è l'arte di progettare algoritmi (o cifrari) per crittografare un messaggio rendendolo incomprensibile a tutti tranne al suo destinatario che con un algoritmo simile deve essere in grado di codificarlo, attraverso un parametro segreto detto chiave, usato in precedenza anche dal mittente per la cifratura. La sicurezza di un sistema di crittografia risiede solo ed esclusivamente nella segretezza della chiave e non dell'algoritmo che è opportuno far conoscere alla pubblica analisi, in modo che se ne possano scoprire eventuali punti deboli in tempo.
Ci sono molte ragioni per crittografare i messaggi che viaggiano sulla rete.
Ad esempio, dovendo inviare informazioni personali a qualcuno, diciamo il numero della carta di credito, e, ovviamente, volendo evitare che qualcuno - che non è il destinatario prestabilito - possa leggere tali informazioni.
Forse ci si chiede come qualcuno possa leggere la posta di altri?! Basta guardare l'area in testa a qualsiasi messaggio di posta elettronica. Come si può vedere, quel messaggio è passato attraverso vari host lungo la strada che ha fatto da mittente a destinatario.
Si pensi a quel messaggio come se fosse una cartolina.
Si consideri solamente che qualsiasi persona che lavora in quegli host riesce tecnicamente a leggere potenzialmente tutte le “cartoline”. Incredibile: la posta ordinaria è più sicura di quella elettronica! Almeno si può mettere le lettere all'interno di buste di carta per impedire che vengano lette occasionalmente.
Ecco quindi l' importanza della crittografia sulla rete.

1.2 La crittografia del passato: alcuni esempi

1.2.1 Il cifrato di Cesare

Per comunicare con i sui generali, Giulio Cesare sostituiva ad ogni lettera del messaggio un'altra lettera un certo numero di posizioni più avanti nell'alfabeto. Per l'esattezza utilizzava la chiave "3", tutte le lettere venivano scalate di tre cifre: la A diventava D, la B diventava E, la C diventava F e così via. Un metodo semplicissimo ma per quell'epoca più che rivoluzionario. Supponiamo di dover decifrare con questo metodo la frase:

PROVA DI CIFRATURA

E di utilizzare la chiave 3. Il testo cifrato (utilizzando il moderno alfabeto) sarà:

SURYD GL FLIUDWXUD

La chiave utilizzata per la cifratura è la stessa che viene utilizzata per la decifratura e per questo deve essere scambiata tra le due parti che devono comunicare. La debolezza di questo cifrato sta bel fatto che come avrete capito si possono utilizzare solo 25 chiavi (tante quante le lettere dell'alfabeto meno una) e basta qualche tentativo sulle prime parole del testo cifrato per capire quale chiave è in grado si decifrare il messaggio. Questo cifrato rimane comunque molto importante per il fatto che ha doto il via a molte altre varianti, alcune delle quali molto valide.
Un primo miglioramento è quello di avere ogni simbolo del testo in chiaro (le 26 lettere) sostituito con qualche altra lettera in modo autonomo e senza una legge fissa. Un sistema di questo tipo è detto a sostituzione monoalfabetica e la chiave è la stinga di 26 lettere corrispondente all'intero alfabeto.

Se per esempio decidiamo di utilizzare la seguente chiave:

QAZWSXEDCRFVTGBYHNUJMIKOLP

Significa che per costruire il nostro cifrato dobbiamo affidarci alle corrispondenze tra il nostro alfabeto e quello generato dalla chiave:

ABCDEFGHIJKLMNOPQRSTUVWXYZ QAZWSXEDCRFVTGBYHNUJMIKOL

Il solito nostro testo in chiaro verrà cifrato così:

PROVA DI CIFRATURA YNBIQ WC ZCXNQJMNQ

Questo tipo di cifrato può sembrare sicuro visto che le possibili chiavi sono 26! (fattoriale), parecchi milioni di miliardi ed andare a tentativi come per il cifrato di Cesare è improponibile.

Ma l'analisi dei cifrati monoalfabetici, dove ad ogni lettera corrisponde un solo carattere segreto, è relativamente facile, e richiede la conoscenza delle varie frequenze delle lettere nelle varie lingue. Si basa infatti sull'esame delle frequenze delle sequenze dei crittogrammi e sulla ricostruzione dei bigrammi (la, il, lo, se ...) e dei trigrammi (per, con, del ...) più frequenti nella lingua. Dato un testo cifrato, si procede eseguendo una statistica sulla frequenza delle lettere presenti e si costruisce un grafico disponendo i caratteri in ordine decrescente. I caratteri più frequenti nel crittogramma, saranno le lettere più frequenti nella lingua.

1.2.2 Il cifrario di Leon Battista Alberti

Nel 1466 Leon Battista Alberti pubblicò un suo libro, scritto qualche anno prima, in cui descriveva i principali metodi di cifratura conosciuti all'epoca e introduceva una nuova tecnica inventata personalmente che consisteva in una sostituzione simile a quella di Cesare con sostituzione periodica della chiave.

Se utilizziamo il nostro solito esempio:

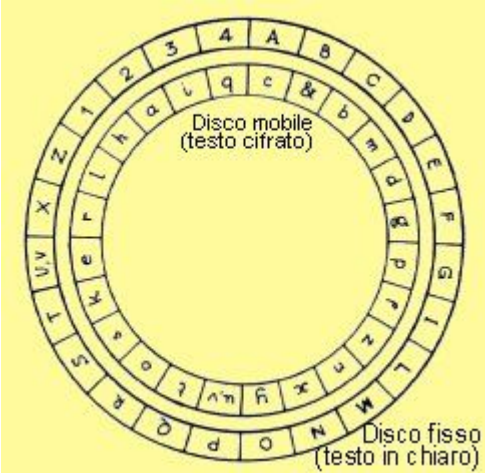
PROVA DI CIFRATURA

E di utilizzare la chiave 4 per la prima parola, la chiave 6 per la seconda e la chiave 5 per la terza. Il risultato della cifratura sarà quindi:

TVSZE JO HNKWFYZVF

La chiave era quindi costituita dalla concatenazione delle varie chiavi usate per ogni parola in modo ciclico. Successivamente Alberti elaborò un sistema che permetteva di inserire all'interno del messaggio l'informazione per il cambiamento della chiave. Il particolare che fa ricordare le idee di Alberti è però un semplicissimo dispositivo "meccanico" di cifratura composto da due dischi concentrici sovrapposti, con quello superiore (il più piccolo) in grado di ruotare che permetteva di trovare, impostando la chiave, tutte le corrispondenze in modo molto rapido.

Figura 1. Cifrario di Leon Battista Alberti



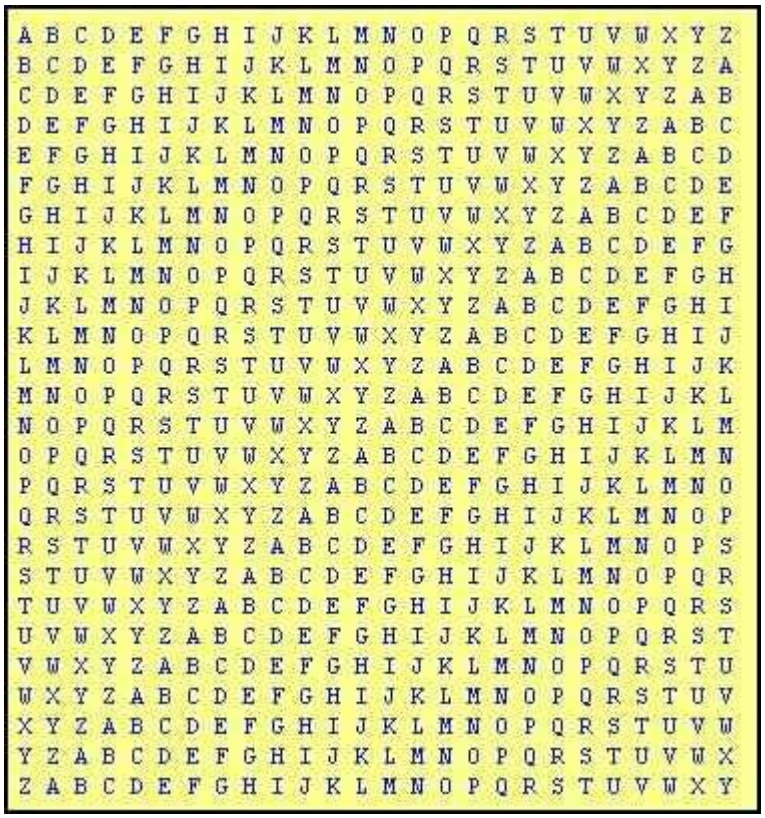
Per quanto riguarda il su cifrario polialfabetico, non riuscì ad ottenere il successo che meritava soprattutto per la decisione dell'autore di tenerla segreta per parecchi anno e quando fu pubblicato il suo trattato la tavola di Vigenère era diventata ormai troppo conosciuta.

1.2.3 Il codice di Vigènere

Il codice di Vigenère si basa un'operazione che viene chiamata sostituzione polialfabetica, molto più sicura di una semplice sostituzione monoalfabetica.
Il Vigenère propose l'uso della tavola quadrata, composta da alfabeti ordinati spostati di una lettera. Il metodo Vigenère ebbe una fortuna immediata, era così efficace che per molti anni fu chiamto "il cifrario indecifrabile" e fu molto usato nell'ambito militare anche dopo che gli analisti ne scoprirono il metodo di decrittazione.

La tavola è composta dalla lista decifrante scritta orizzontalmente in testa; le liste cifranti sono solo le 26 sottostanti ciascuna individuata dalla loro prima lettera che fa parte della chiave per cifrare e decifrare. Ecco la tavola:

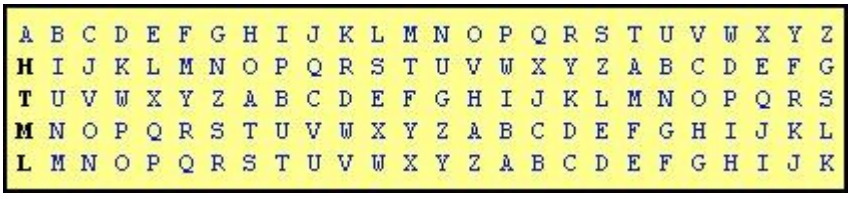
Figura 2. Tavola di Vigenère



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
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	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	S
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	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

Per utilizzare la tavola e cifrare un primo messaggio è necessario dunque scegliere una chiave, ad esempio HTML. A questo punto la tavola appena vista si riduce di qualche riga, cinque per essere esatti, costituite dalla prima e dalle quattro righe che iniziano con le lettere della chiave. La tavola che segue sarà quindi sufficiente per eseguire la cifratura:

Figura 3. Le chiavi



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
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K

Per cifrare si dovrà prima di tutto scrivere le lettere del verme sotto a quelle del testo chiaro e cercare, per ogni lettera del chiaro, la corrispondente cifrata nell'incrocio tra la colonna individuata dalla lettura chiara (quelle della prima riga) e la linea individuata dalla lettera chiave.

```
PROVA DI CIFRATURA
HTMLHTMLHTMLHTML
```

Nel nostro caso quindi alla prima lettera (**P**) corrisponde la lettera **W**, trovata dall'incrocio tra la colonna identificata dalla lettera **P** e la riga identificata dalla lettera **H**. Alla seconda lettera (**R**) corrisponde la lettera **K**, trovata dall'incrocio tra la colonna **R** e la riga **T**. Si continua di questo passo utilizzando la chiave in un ciclo continuo fino a cifrare tutto il testo.

Il risultato finale della cifratura è:

```
WKAGHWUNPYDLANDL
```

La forza di questo cifrato sta nel fatto che il numero di chiavi è enorme, quindi gli attacchi praticati con la forza (ad esempio provare tutte le possibili combinazioni) non sono praticabili. Per quasi trecento anni rimase infatti inviolato ma poi cadde di fronte all'analisi del colonnello prussiano Friedrich Kasiski, che nel 1863 pubblicò un libro che conteneva un metodo di decrittazione della tavola di Vigenère e del suo cifrario polialfabetico con chiave ripetuta. Considerando infatti l'esempio precedente:

```
PROVADICIFRATURA - testo in chiaro
HTMLHTMLHTMLHTML - chiave
WKAGHWUNPYDLANDL - testo cifrato
```

Le due **R** del testo in chiaro vengono cifrate la prima con una **T** la seconda con una **M** come deve essere in un cifrario polialfabetico. Ma le ultime due **A** vengono invece cifrate con la stessa lettera, la **L**. Il motivo è evidente: le due **A** si trovano a quattro caratteri di distanza l'una dall'altra e quattro è proprio la lunghezza della chiave. Di fatto il codice di Vigenère si riduce qui a cinque codici di Cesare intercalati.

L'attacco di Kasiski si basa quindi sull'osservazione che in un crittogramma alla Vigenère si trovano spesso sequenze identiche di caratteri a una certa distanza l'una dell'altra. Questo avviene evidentemente per il motivo esposto sopra, il fatto cioè di utilizzare ciclicamente la stessa chiave. Se per esempio usando una chiave di quattro lettere come sopra, e si scrive due volte la stessa parola a 20 caratteri di distanza questa sarà cifrata in modo identico essendo 20 un multiplo della lunghezza della chiave che è cinque.

Se allora si individuano tutte le sequenze ripetute (e in un testo lungo o in più testi se ne troveranno molte) allora è pressoché certo che il massimo comune divisore tra le distanze tra sequenze identiche è la lunghezza della chiave, o tutt'al più un suo multiplo. Una volta individuata la lunghezza della chiave (supponiamo sia quattro), il messaggio si riduce a quattro messaggi intercalati, tutti cifrati con un codice di Cesare ed è allora molto facile completarne la decifratura.

La conclusione è che la cifra di Vigenère è affidabile solo quando la chiave è di lunghezza comparabile a quella del testo e viene cambiata molto spesso, cosa che comporta problemi pratici non indifferenti (trasmissione e cambiamento della chiave richiedono un canale di comunicazione assolutamente sicuro).

1.2.4 Metodi a trasposizione di colonne e macchina Enigma

1.2.4.1 Cifrario a trasposizione per colonne

Nel cifrario a trasposizione per colonne è una parola o una frase che non contiene alcuna lettera ripetuta. Lo scopo della chiave è numerare le colonne di una tabella che contiene il testo in chiaro scritto orizzontalmente per righe della lunghezza della chiave. Il testo cifrato deve invece essere letto per colonne, seguendo l'ordine alfabetico proposto dalla chiave. Facciamo l'esempio di dover cifrare il testo

PROVA DI CIFRATURA A TRASPOSIZIONE PER HTML.IT

E di utilizzare la chiave HTML. La tabella che ci permette di creare il nostro testo cifrato è la seguente.

Figura 4. Prova di cifratura

1	4	3	2
H	**T**	**M**	**L**
P	R	O	V
A	D	I	C
I	F	R	A
T	U	R	A
A	T	R	A
S	P	O	S
I	Z	I	O
N	E	P	E
R	H	T	M
L	.	I	T

è da sottolineare che in quasi tutti gli algoritmi di crittografia è bene evitare di utilizzare gli spazi perché facilitano di molto il lavoro del crittoanalista.

Tornando al nostro cifrato, il testo sarà quindi il risultato della lettura per colonne seguendo l'ordine alfabetico della chiave (sottolineato da i numeri in rosso):

PAITASINRLVCAAASOEMTOIRRROIPTIRDFUTPZEH

Per analizzare un testo cifrato in questo modo prima di tutto è necessario assicurasi che si tratti di un algoritmo a trasposizione. Per fare ciò quindi si procede controllando che la frequenza delle lettere non corrispondano alla normale frequenza per il testo in chiaro. Il passo successivo consiste nell'ipotizzare il numero di colonne: spesso è facile ipotizzare una parola probabile nel contesto del messaggio e in base ai diagrammi riscontrati di questa parola si può determinare la lunghezza della chiave. A questo punto si tratta solo di determinare l'ordine delle colonne esaminandole singolarmente e procedendo a tentativi.

1.2.4.2 La macchina Enigma

L'ultimo passo prima della così detta crittografia moderna è costituito dalla costruzione dalla macchina elettromeccanica tedesca ENIGMA usata nella seconda guerra mondiale. Essa era composta da ruote con i caratteri incisi sul bordo, e con contatti elettrici in corrispondenza delle lettere in entrambi i lati. Il testo in chiaro, digitato su una tastiera, veniva riprodotto utilizzando i caratteri della prima ruota, la quale a sua volta costruiva un nuovo alfabeto utilizzando i caratteri della seconda, e poi della terza, e cosi' via ... Tutte le ruote, e potevano essere parecchie, venivano "scalate", in modo che la sostituzione delle lettere fosse ogni volta diversa. La chiave consisteva nel settaggio iniziale delle ruote, che potevano essere posizionate in una quantità di posizioni diverse tanto alta quante più erano le ruote utilizzate. Questo meccanismo è facile da costruire via software e abbastanza sicuro, può tuttavia essere infranto. Fu brillantemente attaccato dal matematico polacco Marin Rejewsky che con il suo lavoro permise di decifrare numerosi messaggi militari tedeschi, un fattore che probabilmente contribuì alla vittoria finale degli alleati.

1.3 Crittografia a chiave segreta e a chiave pubblica

La crittografia tradizionale è basata su una **chiave segreta.** Un mittente che vuole inviare un messaggio cifrato a qualcuno, lo cifra usando una chiave segreta ed il destinatario lo decifra usando la stessa chiave segreta. Ovviamente, sia il mittente che il ricevente di quel messaggio devono conoscere la stessa chiave segreta.
Questo metodo e' conosciuto come crittografia a c**hiave segreta** o **crittografia simmetrica.**
Il problema principale è: il mittente ed il ricevente devono accordarsi su una chiave segreta comune, e devono usare un canale sicuro per scambiarsi questa informazione. Cosi' potrebbero usare un corriere fidato, il telefono o...il mittente potrebbe cifrare la chiave segreta!
Ma come puo' il mittente cifrare la chiave? Il ricevente non potrebbe decifrarla, perchè non conosce la chiave stessa! Così, si potrebbe usare il sistema telefonico, certo, ma qualcuno potrebbe intercettare la telefonata...Allora si potrebbe usare un corriere fidato, certo, ma il corriere potrebbe essere corrotto...Per queste ragioni venne inventato un altro sistema di crittografia: il sistema a **chiave pubblica** (chiamato anche sistema **crittografico asimmetrico**).
Il concetto di crittografia a chiave pubblica fu introdotto nel 1976 da Whitfield Diffie e Martin Hellman.
Come funziona questo sistema? Semplice: ogni persona ha una coppia di chiavi, una pubblica ed una privata (chiave privata e chiave segreta sono sinonimi qui). La chiave pubblica di ciascuna persona è pubblicata ed accessibile a tutti - in modo che chiunque la voglia usare lo possa fare - mentre **la chiave privata è tenuta segreta**.
Nessuna informazione segreta deve viaggiare dal mittente al ricevente. Così, se si vuole comunicare con qualcuno crittografando il messaggio, tutto quello che bisogna fare è usare la sua chiave pubblica. Il ricevente di tale messaggio, può poi decifrarlo usando la sua chiave privata. In altre parole: c'è un legame tra la chiave pubblica e la sua corrispondente chiave privata.
Nessuno può recuperare la chiave privata dalla sua corrispondente chiave pubblica! Così, solo il destinatario prestabilito può decifrare un messaggio a lui indirizzato tramite la sua chiave privata.
Purtroppo c'è un problema; cioè, la chiave pubblica è associata al reale destinatario a cui si vuole inviare un messaggio crittografato?!
Per esempio: voglio rendere pubblica la mia chiave '*Topolino*'. La metto sulla rete e dico: 'Salve!

Questa è la mia chiave pubblica: Topolino. Mandami un messaggio!'. Poi qualche tizio briccone, cambia la mia chiave pubblica con la sua chiave pubblica '*chiave-briccone*'. Così il mio messaggio diventa: 'Salve! Questa e' la mia chiave pubblica: chiave-briccone. Mandami un messaggio!'.
Ora supponi che tu voglia inviarmi un messaggio cifrato. Tu non conosci la mia chiave pubblica reale ('Topolino') e credi che la mia chiave sia 'chiave-briccone'.
Cosa succede allora? Semplice: un tizio 'simpaticone' intercetta il messaggio, lo decifra tramite la sua chiave privata (può farlo, perchè tu hai cifrato il messaggio tramite la sua chiave pubblica 'chiave-briccone'!), lo legge, e poi lo cifra nuovamente usando la mia chiave pubblica questa volta. Così io ricevo il tuo messaggio e lo decifro tramite la mia chiave privata (che nessun altro tranne me conosce!), non mi accorgo che una terza persona lo ha intercettato e nessuno si accorge di niente. Perciò, bisogna essere sicuri di una chiave pubblica prima di usarla. In altre parole, quella chiave pubblica deve essere **autenticata**.

1.4 Tecniche e Algoritmi

Di seguito si fornisce un **elenco schematico** delle tecniche e degli algoritmi comunemente usati nella crittografia.
Stante la numerosità questo vuole essere un sintetico panorama delle tecniche e delle sigle ricorrenti senza procedere ad una dettagliata spiegazione di ognuna.
Si rimanda al sito evidenziato in ogni sezione per maggiori dettagli.
Solo per le tecniche o gli algoritmi sottolineati, particolarmente importanti anche per il prosieguo della relazione, si rimanda alle pagine immediatamente successive per una illustrazione approfondita.

1.4.1 Tecniche

- Funzione Senso-Unico
- Problema della fattorizzazione
- Problema dei Logaritmi Discreti
- Cifratori a Blocchi
- Cifratore a Flussi
- Funzioni Hash (vedasi cap. 1.8)
- MAC

Vedasi: http://www.wowarea.com/italiano/aiuto/crytecit.htm

1.4.2 Algoritmi a chiave privata

In un sistema crittografico a chiave segreta (a differenza dei sistemi a chiave pubblica), sia il mittente che il ricevente di quel messaggio devono conoscere la stessa chiave segreta. Ci sono vari sistemi a chiave segreta (o privata):

- DES (vedasi cap. 1.5)
- IDEA (vedasi cap. 1.6)
- SAFER
- RC2
- RC4
- RC5
- FEAL
- SKIPJACK
- BLOWFISH
- SEAL

Vedasi: http://www.wowarea.com/italiano/aiuto/algpri.htm

1.4.3 Algoritmi a chiave pubblica

In un sistema crittografico a chiave pubblica (a differenza di quelli a chiave privata), il mittente ed il ricevente del messaggio non devono conoscere la stessa chiave. Infatti usano 2 coppie di chiavi ciascuno: una chiave pubblica ed una privata. Ci sono vari sistemi a chiave pubblica:

- RSA (vedasi cap. 1.7)
- ELGAMAL
- Elliptic curves
- KNAPSACK
- LUC
- McEliece
- Probabilistic encryption

Vedasi: http://www.wowarea.com/italiano/aiuto/algpubit.htm

1.5 DES

1.5.1 DES

Il DES (Data Encryption Standard) è un cifrario composto sviluppato dall'IBM, modificato dalla National Security Agency (NSA) e adottato dal governo statunitense nel 1977 ufficialmente per la protezione di dati riservati ma non classificati come "segreti militari" o di "stato" e che tuttora è usato da tutte le agenzie federali (fatta eccezione per quegli atti che richiedevano un livello più alto di sicurezza).

Il DES è un codice cifrato a blocchi. La chiave usata per cifrare è un blocco di 64 bit suddivisa in 8 sottoblocchi di 8 bit ciascuno; l'ultimo bit di ogni sottoblocco è di controllo, di conseguenza i bit

liberi che costituiscono in pratica la chiave sono 56.

Il testo da cifrare viene suddiviso in blocchi di 64 bit ciascuno e vengono cifrati uno dopo l'altro in successione con uguale procedimento.

Se un blocco non raggiunge la lunghezza desiderata di 64 bit si utilizza un procedimento detto "pad", che può essere implementato in diversi modi: un metodo aggiunge zeri fino alla lunghezza stabilita mentre un altro, se i dati sono binari, integra il blocco con bit che sono l'opposto degli ultimi bit del messaggio. Nel caso di dati ASCII si usano invece byte generati in modo casuale specificando nell'ultimo byte il carattere ASCII corrispondente al numero di byte aggiunti. Infine un'ultima tecnica, in parte equivalente alla precedente, usa sempre bit casuali ma fornisce, negli ultimi tre bit, il numero di byte originali, cioè quelli che costituiscono il messaggio senza riempimento

Durante la cifratura un blocco di testo normale viene per prima cosa trasposto, cioè, come già ampiamente spiegato in precedenza, cambia posizione con un altro. Poi il blocco di 64 bit viene diviso in una metà destra e una metà sinistra di 32 bit. A questo punto vengono applicati 16 passi tramite una funzione che opera delle trasposizioni e delle sostituzioni ad ogni metà mediante delle sottochiavi diverse per ogni passaggio e ricavate dalla chiave originale.

Durante ogni passo l'output della metà sinistra diventa l'input della destra e viceversa. Dopo il completamento di tutti i 16 passi dell'algoritmo i due sottoblocchi vengono riuniti e sul risultato viene effettuata una sostituzione per invertire la trasposizione iniziale.

L'algoritmo di ogni passo è quindi ricorsivo, cioè utilizza i risultati del passo precedente. Vediamolo in dettaglio le operazioni compiute in ogni passo:

Indichiamo con

- T(i) il risultato del i-esimo passo
- S(i) il semiblocco sinistro
- D(i) il semiblocco destro
- K(i) la sottochiave di ogni passaggio

avremo che:

- T(i)=S(i)D(i)
- S(i)=D(i-1)
- D(i)=S(i-1) XOR f[D(i-1),K(i)]

L'uscita di destra è quindi costituita da un'operazione di OR esclusivo (XOR) bit per bit dell'ingresso di sinistra e di una funzione dell'ingresso di destra e la chiave K(i) per questo blocco. Vediamo come opera la funzione "f":

1. il blocco D(i-1) viene espanso da 32 bit a 48 con un modulo di espansione E. Indichiamo il blocco espanso con E[D(i-1)];
2. si calcola E[D(i-1)] XOR K(i);
3. il risultato precedente viene spezzato in 8 blocchi di 6 bit ciascuno: B(1),B(2)...B(8) contenenti rispettivamente i bit 1-6, 7-12, 13-18 ecc...
4. ciascun blocchetto B(i) viene usato come ingresso ad una funzione Z che restituisce stringhe di 4 bit indicate Z[B(i)]. La funzione Z opera in questo modo: preleva da ogni matrice fissata S-box (Substitution Box) i 4 bit del nuovo blocchetto S(i)=Z[B(i)] posizionati in base alle righe e colonne specificate dai 6 bit del corrispondente B(i);
5. una volta concatenati gli 8 blocchetti S(1), S(2)...S(8) verranno scambiati di posto ottenendo alla fine P[S(1),..S(8)]=f [D(i-1), K(i)].

In precedenza abbiamo più volte parlato di sottochiavi k(i) ricavate dalla chiave originale, ed è

giunto il momento di capire come funziona tutto ciò.

Come detto la chiave è una stringa di 64 bit con 8 bit di controllo che vengono ignorati durante la cifratura/decifratura. Essa viene spezzata in due blocchi di 28 bit, supponiamo di chiamarli, usando la simbologia di prima, S(0) e D(0). Dopodiché per 16 volte i semiblocchi vengono spostati a sinistra ottenendo s(1), D(1), S(2), D(2) ... S(16), D(16). Quindi al primo passo l'algoritmo utilizzerà la sottochiave K(1)=P[S(1)D(1)] dove P al solito indica una permutazione (lo scambio di posto), al secondo K(2)=P[S(2)D(2)] e al 16Â° round K(16)=P[S(16)D(16)]. In questo modo tutte le operazioni effettuate producono sottochiavi K(i) di 48 bit.

Per la decifratura il procedimento è lo stesso, l'unica differenza sta nelle sottochiavi utilizzate in ogni passo: al 1Â° passo verrà utilizzata K(16)=P[L(16)L(16)], al secondo K(15)=P[L(15)L(15)] e cosi via.

Avremo quindi:

T(i)=S(i)D(i)
D(i-1)=S(i)
S(i-1)=D(i) XOR f[S(i),K(i)]

Dove T(i) in questo caso indica il testo cifrato. Queste sono le fasi principali del processo eseguito dal DES per la cifratura e la decifratura del messaggio. Chi volesse conoscere i particolari dei due processi per implementarli in un software, può scaricare dall'indirizzo ftp://ftp.ox.ac.uk/pub/crypto/DES/des-how-to.txt il file (in lingua inglese) che contiene una descrizione passo-passo dell'algoritmo e le matrici S-box nominate sopra.

1.5.2 La sconfitta del DES

Il 17 luglio 1998 la Electronic Frontier Foundation diffonde un comunicato stampa con il quale annuncia la definitiva sconfitta del DES. Per dimostrare i gravi rischi di sicurezza a cui si sottopone chi utilizza il DES, la EFF costruisce il primo apparecchio Hardware non coperto dal segreto di stato per decodificare i messaggi crittografati utilizzando il Data Encryption Standard. In meno di un anno viene costruito un calcolatore costato 250.000 dollari che in meno di sessanta ore era capace di forzare un messaggio cifrato con DES. Tutte le specifiche utilizzate sono documentate in un libro realizzato dalla EFF dal titolo "Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design". Con le informazioni contenute nel libro è possibile realizzabile, a partire da un normale personal computer domestico, il così detto DES cracker. Il testo è disponibile unicamente in versione cartacea perché secondo le leggi USA in materia di esportazioni è reato pubblicare e quindi esportare questo tipo di informazioni su Internet.

1.6 IDEA: l'evoluzione definitiva

IDEA (International Data Encryption Algorithm) è nato nel 1991 sotto il nome di IPES (Improved Proposed Encryption Standard), ed è stato progettato da due famosi ricercatori in Svizzera: Xuejja Lai e James L. Massey. Come il DES è un codice cifrato a blocchi di 64 bit, la differenza sta nel fatto che questa volta però la chiave è di 128 bit, che dovrebbe eliminare elimina qualsiasi possibilità di riuscita di ricerca della chiave procedendo per tentativi, le chiavi possibili sono infatti 2^128.

La cifratura con IDEA comporta una divisione del blocco di 64 bit del testo normale in 4

sottoblocchi di 16 bit. Ogni sottoblocco subisce 8 passi in cui sono coinvolte 52 sottochiavi diverse a 16 bit ottenute dalla chiave a 128 bit. Le sottochiavi sono generate in questo modo:

1. La chiave a 128 bit è divisa in 8 blocchi di 16 che costituiscono le prime 8 sottochiavi
2. Le cifre della chiave a 128 sono spostate di 25 bit a sinistra in modo da generare una nuova combinazione, il cui raggruppamento ad 8 bit fornisce le prossime 8 sottochiavi
3. Il secondo passo è ripetuto finché le 52 sottochiavi sono generate.

Ogni passo comporta calcoli abbastanza semplici come XOR (operazioni di OR esclusivo), addizione e moltiplicazioni in modulo 16 (significa che i risultati non pussono superare i 16 bit quindi quelli eccedenti vengono scartati).

Durante gli 8 passi il secondo e il terzo blocco si scambiano di posto mentre all'ultimo passo i 4 sottoblocchi vengono concatenati per produrre un blocco di testo cifrato a 64 bit.

La decodifica è identica eccetto il fatto che le sottochiavi sono ottenute in maniera diversa dalla chiave principale a 128.

IDEA è al momento il cifrario a chiave segreta più utilizzato quanto riguarda i software commerciali di crittografia vista la sua velocità di codifica e decodifica e la sua elevata sicurezza.

1.7 RSA

La prima applicazione pratica basata sulle tecniche di crittografia a doppia chiave fu sviluppata nel 1978 da tre professori: Ronald Rivest, Adi Shamir e Leonard Adleman, che realizzarono una procedura di calcoli matematici che prenderà il nome di "algoritmo RSA", dalle iniziali dei suoi inventori. Quando ci si rese conto dell'efficacia di questo algoritmo, ritenuto ancora oggi inattaccabile, il governo americano decise che i programmi basati su questo algoritmo potevano essere utilizzati liberamente negli Stati Uniti, ma la loro esportazione costituiva reato. Un altro ostacolo all'immediato sviluppo di questo algoritmo è dovuto al fatto che i tre inventori del sistema RSA decisero nel 1982 di brevettare il loro algoritmo e fondare la RSA Data Security Inc, una compagnia nata per lo sfruttamento commerciale del loro sistema di crittografia. Nonostante le restrizioni statunitensi all'utilizzo dell'RSA, al di fuori degli USA, dove il governo americano non ha potere e gli algoritmi non sono coperti da brevetto, iniziano a diffondersi numerosi programmi ispirati molto da vicino alla tecnica di Rivest, Shamir e Adleman.

Il codice RSA si basa su un procedimento che utilizza i numeri primi e funzioni matematiche che è quasi impossibile invertire. Dati due numeri primi, e' molto facile stabilire il loro prodotto, mentre è molto più difficile determinare, a partire da un determinato numero, quali numeri primi hanno prodotto quel risultato dopo essere stati moltiplicati tra loro. In questo modo si garantisce quel principio di sicurezza alla base della crittografia a chiave pubblica infatti l'operazione di derivare la chiave segreta da quella pubblica è troppo complessa per venire eseguita in pratica. Vediamo in pratica l'algoritmo e le varie funzioni matematiche:

1. Calcolare il valore di n, prodotto di p e q, due numeri primi molto elevati (sono consigliati valori maggiori di 10^100. Negli esempi in seguito utilizzeremo numeri più piccoli per facilitare la lettura. Esempio: n = p*q = 7*5= 35
2. Calcolare il valore di z =(p-1)*(q-1).
   z = (p-1)*(q-1) = 24
3. Scegliere un intero D tale che D che sia primo rispetto a z, il che significa che i due numeri

non devono avere fattori primi in comune. Esempio:
z = 24
d=7

4. Trovare un numero E tale che E*D mod z = 1, cioè che il resto della divisione tra E*D e z sia 1.
E = 7 --> 49 mod 24 = 1

Dopo aver calcolato questi parametri inizia la cifratura. Il testo in chiaro viene visto come una stringa di bit e viene diviso in blocchi costituiti da kbit, dove k è il più grande intero che soddisfa la disequazione 2^k < n.

A questo punto per ogni blocco **M** si procede con la cifratura calcolando **Mc = M^E mod n**. In ricezione, invece, per decifrare **Mc** si calcola **Mc^D mod n**. Come si può capire da quanto appena detto per la cifratura si devono conoscere E ed n che quindi costituiranno in qualche modo la chiave pubblica, mentre per decifrare è necessario conoscere D ed n che quindi faranno parte della chiave segreta.

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.

1.8 Funzioni di Hash

Una di **hash**, detta anche **one way hash**, trasforma un testo normale di lunghezza arbitraria in una stringa di lunghezza relativamente limitata. Questa stringa rappresenta una sintesi del messaggio (message digest) che non è altro che una vera e propria impronta digitale unica che viene definita valore di hash e che gode di tre importanti proprietà:

- dato un messaggio si può facilmente calcolare il suo valore di hash
- dato il valore di hash è impossibile risalire al messaggio (per questo one way hash)
- non si possono generare due messaggi che abbiano la stessa sintesi. Sappiamo che questo non è praticabile ma in genere si intende che la probabilità di collisione (due messaggi con la stessa sintesi) deve essere molto bassa.

Solitamente per le **impronte** vengono utilizzati 128 bit, ma il valore può essere qualsiasi, tenendo conto che più basso è e più alta è la probabilità di collisione.

Per una dimostrazione forniamo 2 valori di hash ottenuti tramite l'algoritmo MD5 di cui parleremo tra poco, che estrae da input di qualunque dimensione valori di hash di 128 bit che possono essere rappresentati con 16 cifre esadecimali:

```
"a" --> 60B725F10C9C85C70D97880DFE8191B3
"Prova hashing per HTMLit" --> EFC56F6C520FFB812BB9854D093AD43
```

come si può notare, da stinge in ingresso di lunghezza significativamente differente, sono stati calcolati due valori hash della stessa lunghezza e apparentemente simili. La lunghezza dei valori di hash varia a seconda degli algoritmi. Quelli a 128 bit sono i più comuni, come il sopra citato MD5 del 1992.

L'**algoritmo MD5** utilizza un buffer di 128 bit inizializzato a un valore prefissato. Divide il messaggio originale (visto come una stringa di bit) in blocchi di 512 bit aggiungendo se necessario dei bit aggiuntivi per arrivare a tale cifra e per ogni blocco di 128 bit vengono eseguiti quattro passi che consistono nel mescolare completamente i 512 bit in ingresso con il buffer di 128 fino a che tutti i blocchi in ingresso sono stati consumati. Alla fine il buffer sarà il message digest del testo in ingresso.

Capitolo 2 – PGP

2.1 Cos'è PGP

Nel giugno 1991 lo statunitense **Philip Zimmermann** realizza e distribuisce gratuitamente il programma **PRETTY GOOD PRIVACY** (PGP) un programma di crittografia diventato ormai uno standard permette di mantenere la privacy e la sicurezza dei propri dati personali in formato elettronico.

l nome gli è stato suggerito da una drogheria di Lake Wobegon, la città natale dello speaker radio Garrison Keillor. La drogheria si chiamava "Ralph's Pretty Good Grocery" ("la drogheria assai buona di Ralph") e il suo slogan era "se non lo puoi trovare da Ralph, probabilmente puoi anche farne a meno".

Per la realizzazione di PGP, Zimmermann viene citato in tribunale dalla RSA Data Security Inc. per violazione del brevetto sull'algoritmo RSA, e accusato dal governo degli Stati Uniti di esportazione illegale di strumenti crittografici. Entrambe le cause finiscono nel nulla. L'accusa di esportazione illegale viene ritirata nel 1996, mentre la controversia con RSA verrà mediata da una successiva collaborazione tra le due parti per la realizzazione delle versioni successive del software.

PGP è un programma di crittografia, anzi, il programma di crittografia per eccellenza, che garantisce la segretezza della posta (ma non solo), l'autenticazione con firma digitale e la compressione.

2.2 Le origini, la storia e i problemi

Questo capitolo non intente essere una successione di date e di eventi per il solo gusto di confondere e complicare la lettura.
Il proposito è invece di sottolineare l'importanza di un applicativo di questo genere.
Le traversie per lo sviluppo e l'utilizzo, le restrizioni, il paragone ad “arma da guerra” vogliono appositamente impressionare per introdurre considerazioni su PGP e sulle problematiche legate alla privacy ed alla sicurezza dei dati.
Tanto rispetto e ringraziamenti per creatore e sviluppatori del programma PGP.

2.2.1 I primordi

PGP fu interessato dalle restrizioni sulla esportazione della crittografia del governo USA, già dagli albori. Questo inconveniente dimostra alcuni dei problemi politici che circondano la crittografia moderna di qualità e costituisce, purtroppo, un pasticcio contingente e difficile da seguire. Che minaccia tempi di prigionia piuttosto elevati e multe salate.

Zimmermann produsse la prima versione del PGP nel 1991. Era stato per molto tempo un attivista anti-nucleare e creò il PGP in modo che i suoi compagni potessero usare sistemi BBS e memorizzare messaggi e files in tutta sicurezza. Non era richiesta licenza se l'uso non era

commerciale, non c'era spesa neanche simbolica e veniva fornito il codice sorgente. PGP si diffuse su Usenet e di qui su Internet.

2.2.2 Limitazioni all'esportazione, indagini giudiziarie e ricadute politiche

In breve tempo, il PGP iniziò a diffondersi al di fuori dei confini degli USA, e nel febbraio 1993 Zimmermann fu formalmente indagato dal governo degli Stati Uniti con l'accusa di "esportazione di armi senza apposita licenza".

In molti trovarono difficilmente comprensibile questa accusa e fu osservato, in tono satirico, che un programma software può facilmente diventare un'arma con l'aggiunta di un detonatore e un po' di esplosivo.

Ciononostante, secondo quanto stabilito dal Regolamento per l'Esportazione dei prodotti e servizi USA i sistemi di crittografia che utilizzassero una chiave maggiore di 40 biterano considerati come munizioni e, dato che il PGP ha sempre adottato chiavi maggiori di 128 bit, all'epoca rientrava perfettamente nella casistica. Oltretutto le sanzioni, per chi fosse riconosciuto colpevole, erano e rimangono rilevanti. I regolamenti all'export statunitensi sono ancora in vigore ma, nel corso della seconda metà degli anni Novanta, hanno subito importanti modifiche: la giurisdizione è stata assegnata al Dipartimento del Commercio invece di essere dipendente dal Dipartimento di Stato; i sistemi di crittografia sono stati riclassificati come duplice uso, invece di essere considerati come munizioni; il processo di approvazione è stato semplificato nel 2000; infine la lunghezza della chiave di norma ammessa è stata portata (in parecchi e confusi passaggi) oltre la soglia dei 40 bits.

Dal 2000 in poi il rispetto dei regolamenti è diventato inoltre più semplice. Ad esempio, nei primi anni Novanta, l'approvazione di una domanda di export doveva ricevere - per definizione - un'approvazione esplicita. Dal 2000 in poi, invece, si ha un periodo di attesa di 30 giorni durante il quale l'approvazione può essere negata, altrimenti vige il principio del silenzio-assenso. Sono state inoltre facilitate le procedure per approvare il codice sorgente, rispetto alla programmazione orientata agli oggetti, come pure altre facilitazioni per sviluppatori *open source* o per piccole aziende. Prima di imbarcarsi in un importante progetto crittografico che potrebbe essere interessato da tali regolamentazioni, sarebbe comunque il caso di consultare un avvocato che sia esperto degli aspetti legali legati alla crittografia.

Come conseguenza delle modifiche normative introdotte il PGP non è, attualmente, più definibile come "arma non esportabile" e può essere utilizzato (ed esportato) ovunque (a condizione che la legislazione locale lo consenta). Inoltre l'incriminazione a carico di Zimmerman è stata archiviata senza che siano state rubricate condotte criminali in capo allo stesso o ad altri soggetti.

Dal momento che la regolamentazione statunitense sull'*export* non è comunque universalmente applicabile, il PGP, dopo il suo rilascio, acquisì un seguito considerevole in tutto il mondo. Tra gli utilizzatori e gli estimatori vi erano sia dissidenti nei paesi totalitari (alcune affezionate lettere di questi ultimi a Zimmerman sono state pubblicate ed utilizzate nelle deposizioni tenutesi davanti al Congresso statunitense), liberali in diverse parti del mondo (cfr. le pubblicazioni delle deposizioni di Zimmerman nel corso di varie udienze) nonché gli attivisti della 'libera comunicazione' che si autodefiniscono *cyberpunk*; questi ultimi si fecero promotori sia della pubblicizzazione che della diffusione del PGP (per i dettagli vedi diversi loro manifesti).

2.2.1 I problemi con i brevetti

Le versioni iniziali di PGP hanno avuto anche dei problemi con i brevetti già registrati. La prima versione ricorreva a un cifrario progettato dallo stesso Zimmermann e denominato *Bass-O-Matic* da uno sketch del Saturday Night Live in cui si utilizzavano dei pesci e dei tritatutto in cucina. Ben presto ci si accorse però che questo cifrario non era sicuro, così venne rimpiazzato dal cifrario IDEA. Entrambi gli algoritmi, sia quello per la chiave simmetrica IDEA che quello per la chiave asimettrica RSA, sono stati brevettati e per utilizzarli è necessaria un'autorizzazione. All'epoca ci fu un acceso dibattito sul fatto che Zimmermann avesse avuto l'autorizzazione per utilizzare il RSA nel PGP. A tal proposito, Zimmermann dichiarò che l'allora RSA Data Security (operativa adesso come RSA Security), in uno dei primissimi incontri, gli avesse dato l'autorizzazione purché fosse a scopo non-commerciale mentre RSA smentiva tutto quanto. Di conseguenza, fu un reclamo partito dalla RSADSI alla Dogana statunitense a creare il cosiddetto *caso Zimmermann* sull'uso dell'algoritmo RSA nel PGP.

Per complicare ulteriormente il quadro, l'algoritmo RSA venne brevettato solamente negli USA (questo per la difficoltà di rispondere alle varie modulistiche di richiesta brevetti nel mondo) con il risultato di poter essere utilizzato liberamente (tenendo sempre conto delle problematiche di brevetto viste in precedenza) in tutte le altre nazioni. Ciò nonostante, gli inventori / proprietari di IDEA furono decisamente più liberali negli USA che nell'Unione Europea. E se non ci fosse già abbastanza confusione, il brevetto sull'algoritmo RSA era parzialmente controllato dal MIT attraverso il proprietario del brevetto, il RSADSI: gli inventori del RSA lavoravano tutti al MIT al momento della sua creazione.

In qualunque modo andò la disputa Zimmermann/RSADSI, il MIT ebbe pochi problemi con il PGP; invece, si trovò parecchio in difficoltà a causa della posizione ostile del RSADSI , contraria all'uso non commerciale del RSA nel PGP. Il risultato del conflitto sulla licenza del RSA su un fork di PGP in:

- una versione USA (conforme al brevetto RSA) che usa una libreria crittografica shareware dell'RSA
- una versione internazionale che usa il codice RSA originale creato da Zimmermann ed i suoi collaboratori

La versione USA fu distribuita direttamente dallo stesso MIT, insieme ad altri, attraverso Internet, le BBS ed utenti e gruppi di sistemi di comunicazione privata come AOL e CompuServe. Alla fine sul sito del MIT, c'era il requisito che l'indirizzo email al quale PGP sarebbe stato inviato fosse negli USA od in Canada, e che il ricevente fosse residente in uno dei due stati.

In Norvegia Ståle Schumacher Ytterborg sviluppò e mantenne la versione internazionale del PGP, che venne chiamata PGP-i (dove *i* sta per *internazionale*). Era desiderabile al tempo che la versione internazionale fosse sviluppata e mantenuta fuori dagli USA per evitare ulteriori difficoltà con le regolamentazioni USA sull'esportazione e con il brevetto RSA.

2.3 La logica e l'algoritmo di funzionamento.

Il software in questione non fa altro che implementare in pratica i sistemi visti nel precedente capitolo, utilizzando un sistema di crittografia misto con tre algoritmi: il sistema a chiavi pubbliche

RSA, quello a chiavi private IDEA e l'algoritmo di hashing MD5. È distribuito gratuitamente per l'uso personale e può essere scaricato dal sito www.pgp.com.

Il suo funzionamento è molto semplice: ammettiamo che l'utente A voglia spedire all'utente B un messaggio. PGP cifra tale messaggio utilizzando IDEA con una chiave K generata casualmente che verrà inviata all'utente B cifrata con al sua chiave pubblica con l'algoritmo RSA, insieme al messaggio cifrato con IDEA. In questo modo solo B può, con la propria chiave Privata, recuperare la chiave K ed usarla per leggere il resto del messaggio.

Come detto nei paragrafi precedenti, gli algoritmi a chiave pubblica risolvono anche il problema della firma digitale e con PGP è possibile firmare i messaggi, o meglio una loro sintesi creata tramite l'algoritmo MD5.

Ogni utente del programma, dopo la procedura di installazione, possiede quindi due chiavi, una privata (da mantenere nel o nei pc che si utilizzano) e una pubblica. PGP prevede la creazione di archivi pubblici elettronici che contengano le chiavi pubbliche dei vari utenti. È importante molto importante utilizzare questo tipo di supporti per dare la massima diffusione della propria chiave pubblica e consentire a tutti di poter comunicare con noi in modo sicuro.

2.4 Quanto è importante PGP

E' personale. E' privato. E non sono affari di nessuno tranne vostri. Voi potreste pianificare una campagna politica, discutere delle vostre tasse, o avere una relazione clandestina. Oppure potreste fare qualcosa che sentite che non dovrebbe essere illegale, ma lo è. Qualunque cosa sia, voi non volete che la vostra posta elettronica privata (e-mail) o i vostri documenti confidenziali vengano letti da altri. Non c'è niente di sbagliato nel voler affermare il proprio diritto alla riservatezza. La riservatezza e' parte della nostra vita come la Costituzione.

Forse potreste pensare che la vostra e-mail sia sufficientemente legittima da non richiedere cifratura. Ma se siete cittadini rispettosi della legge senza nulla da nascondere, perché non spedite sempre la vostra posta usando le cartoline? Perché non vi sottoponete ai test antidroga su semplice richiesta? Perché richiedere un mandato se la polizia vuole perquisire la vostra casa? State cercando di nascondere qualcosa? Dovete essere dei sovversivi o dei trafficanti di droga se nascondete la vostra posta nelle buste. O forse paranoici. Che bisogno hanno i cittadini rispettosi della legge di cifrare la propria e-mail?

Cosa succederebbe se tutti pensassero che i cittadini rispettosi della legge dovrebbero usare le cartoline per la propria posta? Se qualche spirito coraggioso tentasse di difendere la propria riservatezza usando una busta attirerebbe i sospetti. Forse le autorità aprirebbero la sua posta per vedere ciò che nasconde. Per fortuna, non viviamo in questo tipo di mondo, perché tutti proteggono la maggior parte della propria posta con le buste, cosi' nessuno attira sospetti facendo la stessa cosa. C'è sicurezza nei grandi numeri. Analogamente, sarebbe bello se tutti usassero regolarmente la crittografia per la propria e-mail, innocente o meno, di modo che nessuno attirerebbe sospetti difendendo la propria riservatezza. Pensate a questo come ad una forma di solidarietà.

Oggi, se il Governo vuole violare il diritto alla riservatezza dei cittadini ordinari, deve impegnare una certa quantità di denaro e di lavoro per intercettare, aprire col vapore e leggere la posta, e ascoltare e possibilmente trascrivere le conversazioni telefoniche. Questo tipo di sorveglianza ad alto impegno lavorativo non e' pratica se applicata su larga scala. Questo viene fatto solo in casi

importanti, quando sembra che valga la spesa.

Una parte sempre crescente delle nostre comunicazioni private si sta dirigendo verso i canali elettronici. La posta elettronica sta gradualmente rimpiazzando la posta tradizionale. I messaggi e-mail sono semplicemente troppo facili da intercettare e controllare ricercando parole significative. Può essere fatto semplicemente, regolarmente, automaticamente e senza essere scoperti su vasta scala. I cablogrammi internazionali sono gia' controllati in questo modo su larga scala dalla NSA [National Security Agency, N.d.T.]

Ci stiamo muovendo verso un futuro in cui la nazione sarà attraversata da reti dati in fibra ottica ad alta capacita' che collegheranno fra loro tutti i nostri sempre più diffusi personal computers. L'e-mail è ormai normale quasi per tutti. Il Governo proteggerà la nostra posta con protocolli di crittografia progettati dal Governo stesso. Probabilmente la maggior parte delle persone accetterà una simile situazione. Ma forse qualche persona preferirà le proprie misure di protezione personali.

Il progetto di legge 299 del senato, una proposta anti crimine multifunzionale del 1991, conteneva un provvedimento nascosto sconvolgente. Se questa risoluzione non vincolante fosse entrata in vigore, avrebbe costretto i produttori di sistemi di comunicazioni sicure ad inserire delle speciali "trappole" nei loro prodotti, di modo che il Governo potesse leggere i messaggi cifrati di chiunque. Il testo era: "E' volontà del Congresso che i fornitori di servizi di comunicazione elettronica assicurino che i sistemi permettano al Governo di ottenere il testo in chiaro di voce, dati ed altre comunicazioni quando adeguatamente autorizzato dalla legge." Questa misura venne ritirata dopo rigorose proteste di singoli libertari e gruppi industriali.

Nel 1992 fu presentata al Congresso la proposta di sorveglianza della telefonia digitale dell'FBI. Questa avrebbe richiesto a tutti i produttori di sistemi di comunicazione di includervi speciali "porte" di sorveglianza remota che avrebbero permesso all'FBI di controllare dai propri uffici tutte le forme di comunicazione elettronica. Sebbene la proposta non abbia attratto nessun sostenitore al Congresso nel 1992 a causa dell'opposizione dei cittadini, essa fu ripresentata nel 1994.

Più allarmante di tutto e' la baldanzosa nuova politica sulla crittografia della Casa Bianca, sviluppata dalla NSA sin dall'inizio dell'amministrazione Bush, e rivelata il 16 aprile 1993. Il punto centrale di questa iniziativa e' un dispositivo di crittografia costruito dal Governo, chiamato "Clipper", contenente un nuovo algoritmo segreto di crittografia della NSA. Il Governo spinge l'industria privata ad utilizzare il Clipper in tutti i propri prodotti di comunicazione sicura, come telefoni, Fax, ecc. AT&T sta utilizzando il Clipper nei propri prodotti sicuri per trasmissione di voce. Il punto chiave: Al momento della produzione ogni Clipper viene programmato con una sua chiave unica, ed il Governo deve riceverne una copia, che viene tenuta al sicuro. Nessuna preoccupazione comunque il Governo promette che usera' quelle chiavi per decifrare le vostre comunicazioni solo quando debitamente autorizzato dalla legge. Naturalmente il logico passo successivo per rendere il Clipper assolutamente efficace sarebbe quello di rendere illegali le altre forme di crittografia.

Se la propria riservatezza e' fuorilegge, solo i fuorilegge avranno riservatezza. Le agenzie di spionaggio hanno accesso a sistemi di crittografia ottimi, cosi' come i trafficanti di armi e droga, i fornitori della Difesa, le compagnie petrolifere ed altri giganti industriali. Le persone comuni e le organizzazioni politiche minori non hanno generalmente avuto accesso a tecnologie di crittografia a chiave pubblica di "livello militare" a costi abbordabili. Fino ad oggi.

PGP da' alle persone il potere di avere la propria riservatezza nelle proprie mani. C'è un bisogno sociale crescente di questo.

Capitolo 3 – Uso di PGP

3.1 L'applicativo PGP

Il programma PGP ha subito molteplici evoluzioni adeguandosi all'evolversi dei sistemi operativi utilizzati dagli utenti.
Si può trovare “dal DOS ai giorni nostri”.
Per chi volesse approfondire sviluppi e versioni si rimanda al sito di
E' distribuito gratuitamente per l'uso personale, in trial version durata 30 giorni, e può essere scaricato dal sito www.pgp.com.

3.2 Come installare PGP

L'installazione di PGP è sufficientemente guidata da non richiedere troppe spiegazioni. In una delle prime schermate viene chiesto di selezionare i componenti da installare, dei plug-in che andranno ad integrarsi con noti programmi quali Outlook, ICQ, Eudora... ed è possibile anche installare **PgpNet**, che comprende un tool per il rilevamento delle intrusioni, un firewall e un componente per cifrare il traffico un una rete privata.

Successivamente vi verrà chiesto se volete creare la vostra coppia di chiavi o se già ne possedete una. Proseguiamo quindi a creare la nostra coppia di chiavi. Il programma di installazione farà partire un "key generation wizard" che ci guiderà nell'operazione. Per prima cosa viene chiesto di inserire il proprio nome completo e il proprio indirizzo e-mail, necessari per permettere di identificare la chiave pubblica dagli altri utenti. La schermata successiva ci chiederà di inserire una passphrase, che serve essenzialmente per proteggere la vostra chiave privata (residente nel pc) dall'eventuale utilizzo da parte di persone non autorizzate. Il fatto di utilizzare la parola passphrase invece di password sottolinea l'importanza di scegliere una combinazione difficile da indovinare da estranei, anche perché la sicurezza di PGP si basa sulla passphrase. Fatto questo la coppia di chiavi è stata creata e dopo una schermata che ci avvisa di ciò, verrà chiesto di riavviare il computer per cominciare ad utilizzare PGP.

3.3 Come usare PGP

3.3.1 La gestione delle chiavi

PGP non possiede un vero e proprio ambiente di lavoro, ma è costituito da una serie di **tools** che consentono di eseguire una o più funzioni del programma, che possono essere richiamate in vario modo. Una volta installato, PGP viene avviato ogni volta che si accende il pc e l'icona appare nella barra delle applicazioni accanto all'orologio (se non si desidera questo è sufficiente rimuoverlo dall'esecuzione automatica di windows). Cliccando sull'icona appare il menù di PGP che ci permette di accedere a tutte le sue funzioni:

Il comando **Hide** serve per chiudere il programma, About PGP visualizza delle informazioni sul software, Help fa partire la guida e Optino permette una configurazione dettagliata del programma che non è necessaria per un uso di base. L'opzione PGPnet serve per la gestione dei moduli del programma relativi alle opzioni di protezione della comunicazione in una rete privata mentre molto importanti sono i rimanenti comandi PGPtools, PGPkeys e ClipBoard.

3.3.2 PGPtools

Dal comando PGPtools si accede ad una pulsantiera che racchiude tutte le funzioni rese disponibili dal software:

I sette pulsanti sullo schermo, permettono rispettivamente di:

1. PGPKeys: richiama il modulo di gestione delle chiavi per le operazioni di cifratura e firma digitale
2. Encrypt: per cifrare un documento (inteso come file)
3. Sign: applicare la firma digitale ad un file
4. Encrypt and Sign: per eseguire in una volta le due funzioni appena viste
5. Decrypt and verify: per decifrare un file o verificarne la firma
6. Wipe: letteralmente pulire, serve per cancellare un file in modo sicuro, senza che possa essere più reperito in alcun modo
7. Freespace Wipe: fa una pulizia del disco rimuovendo tutti i possibili "pezzi" di file ancora presenti nel disco

3.3.3 PGPkeys

PGPkeys è l'interfaccia che permette la gestione delle chiavi pubbliche possedute dall'utente del software, da utilizzare per le operazioni di crittografia e di firma digitale. La schermata che appare è la seguente:

3.3.4 Salvare la propria chiave privata

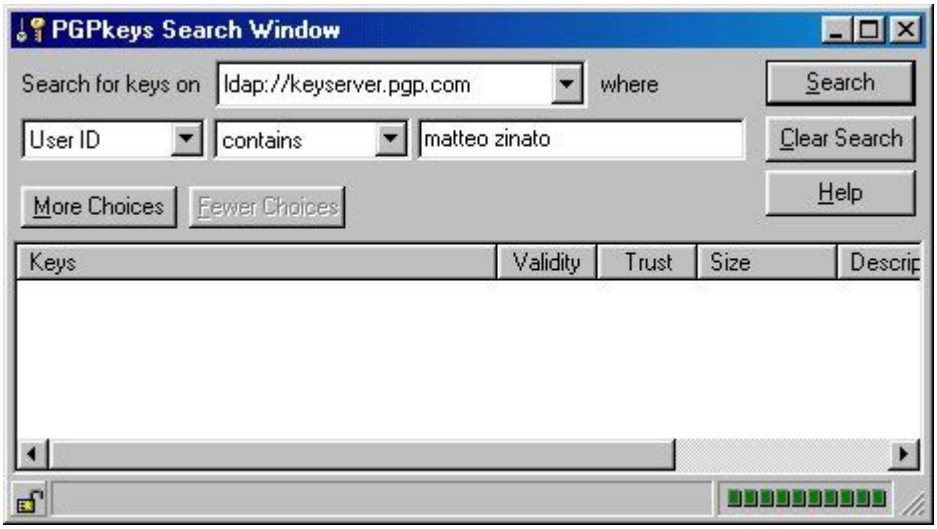
La prima cosa da fare è quella di salvare la propria chiave privata in un supporto magnetico, perché se perdiamo questa, non potremo più decifrare i documenti in nostro possesso. Per fare ciò è sufficiente selezionare la chiave col proprio nome e dal menu Keys cliccare su Export, selezionando poi l'opzione "include private keys". A questo punto la nostra chiave è stata salvata nel supporto da noi indicato in un file.asc.

3.3.5 Pubblicare la propria chiave pubblica

Come detto in precedenza è molto importante dare la massima diffusione della propria chiave pubblica per fa si che chiunque possa comunicare con noi in modo sicuro. PGP mette a disposizione dei propri utenti un server di chiavi (keyserver.pgp.com) su cui è possibile effettuare operazioni di pubblicazione e ricerca di chiavi. Per la pubblicazione è sufficiente selezionare la propria chiave e premere CTRL+K (Server -> Send to -> Domain server). Il programma si collegherà automaticamente col server e effettuerà la pubblicazione. Se per qualche motivo una persona non desiderasse pubblicare la propria chiave, può comunque salvare la propria chiave (attenzione stavolta a non includere quella privata) in un file .asc come appena visto e poi mandare il file alle persone con cui desidera comunicare.

3.3.6 Importare una chiave pubblica

Nel PGPkeys di prima erano presenti soltanto quattro chiavi, quella dell'utente che ha istallato il programma e tre chiavi di PGP per la gestione della sicurezza. Ovviamente in questo stato PGP potrebbe essere usato solo per cifrare documenti per uso personale, cosa più che legittima ma che sicuramente non sfrutta le potenzialità di PGP. È necessario quindi importare le chiavi delle persone con cui si desidera comunicare. Premendo semplicemente CRTL+F (Server -> Search) apparirà una schermata che ci consentirà di effettuare la ricerca su server di chiavi nominato prima.



Effettuando una ricerca per User ID è sufficiente inserire il nome della persona da cercare, ma è possibile anche modificare i criteri di ricerca. Se la ricerca darà esito positivo, potremo importare la chiave semplicemente selezionandola col tasto destro del mouse e cliccando su Import. Ma se invece, tornando all'esempio di prima, dobbiamo importare una chiave da un file.asc, basta premere CTRL+M (Keys -> import) e selezionare il file che contiene la chiave da importare. In un modo o nell'altro possiamo quindi crearci il nostro contenitore di chiavi pubbliche.

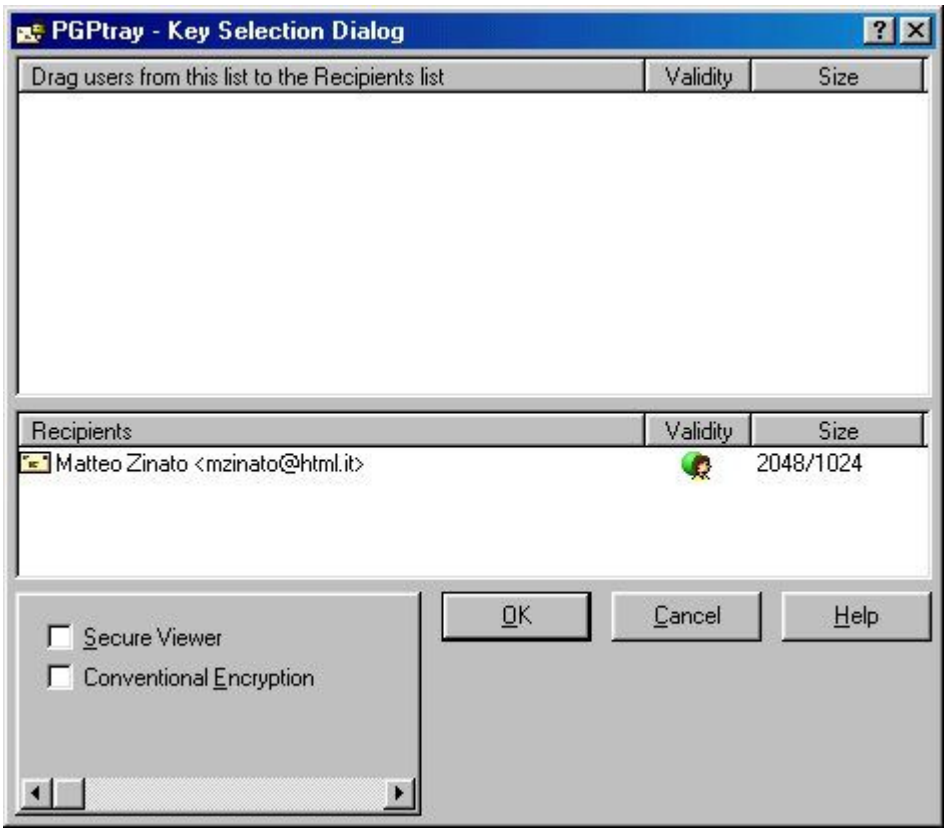
3.4 Utilizzo pratico di PGP

A questo punto tutto è veramente pronto e possiamo vedere in pratica come **cifrare i nostri documenti**, che possono essere costituiti da file di qualsiasi tipo, semplici righe di testo o e-mail.

3.4.1 Cifrare un file

La cifratura di un file può essere fatta mediante il secondo pulsante presente nel PGPtools. Una volta premuto, una tipica finestra di select ci permette di selezionare il file da cifrare. Fatto questo ci apparirà la finestra di PGP per la selezione delle chiavi. Questa è una schermata che apparirà molto spesso durante l'utilizzo del programma perché permette di selezionare le chiavi pubbliche con cui

cifrare. Nella parte alta (in questo caso ancora vuota) sono presenti tutte le chiavi contenute nel PGPkey. Tramite un semplice drag-and-drop è possibile spostarle nella parte inferiore, che conterrà in pratica l'elenco degli utenti in grado di decifrare il file. Da notare che chi esegue la cifratura è presente di default nell'elenco.



A questo punto, una volta dato l'OK, verrà creato un file con lo stesso nome ma solo apparentemente con la stessa estensione, visto che si tratta di un file.pgp, caratterizzato dalla tipica icona con il lucchetto. Il file è cifrato, possiamo inviarlo a nostri destinatari o decifrarlo personalmente. Per farlo si può utilizzare il pulsante Decrypt/Verify dei PGPtools o semplicemente aprire il file con il doppio click del mouse. In entrambi i casi sarà richiesto di inserire la Passphrase che come abbiamo già detto viene richiesta ogni volta che si deve utilizzare la chiave privata, e il nostro file "leggibile" verrà creato.

3.4.2 Cifrare il testo

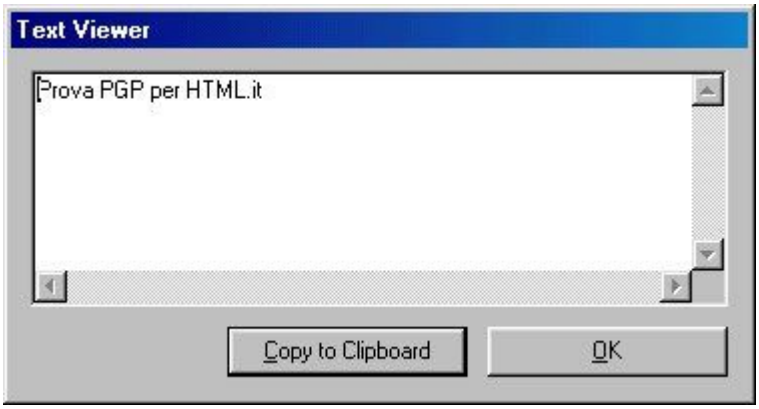
Se non si ha la necessità di inviare l'intero contenuto di un file ma semplicemente del testo scritto, si può utilizzare semplicemente il blocco note di windows e l'opzione Clipboard di PGP. Supponendo di avere del testo scritto, copiamolo negli appunti di windows (CTRL+C). Aprire il menu PGP dall'icona presente nella barra delle applicazioni e selezionare Clipboard -> Encrypt (o encrypt and sign), comparirà la finestra vista sopra per la selezioni delle chiavi di cifratura. Premendo OK la finestra PGP si chiuderà e il nostro testo cifrato sarà presente negli appunti di windows e reperibile tramite il comando "incolla" o CTRL+V. Facciamo un semplice esempio, supponendo di avere il testo:

Prova PGP per HTML.it Una volta cifrato sarà così, anche se ovviamente tutto dipende dalle chiavi utilizzate:

```
-----BEGIN PGP MESSAGE-----
Version: PGPfreeware xx.x.x for non-commercial use <HTTP: www.pgp.com>

qANQR1DBwU4DIpTQJS7yBF4QB/9xgTxW3ynosHkRJJ1qYoLoF8CqVYhQHLt0T091
8gFic6NKxzIMZycC+ZzWmaXmqmFdRR7gXBXds288vHyO6370IsuaYhvCFILX8CCC
HSCF6vrJCPXW9oNFvs3wN6t6cBADJ0rlmn/elO43e+eqeXPFOEU+9Yh3++9lT6zt
DZEd2GlH/txg1acMiYp03qai9fiFKMzXnNoEwzLcLbp+kdmTqkzlBiPE0SHehP7m
dBE9vchlWwILu7u9xwqEOdHz+QL3/h+yvXwO1M1DKUVS2Ygt7HpzQAjOmbuWiMER
zKQUrUXT/VK8h6MEpYi5w5aPspV/QRk3oxXdmsAZ6P3HeKplCADVKJChFyC/RndT
ky7Sd8smGRYva70y/pk6ecf4p1zK3tgRZK0pmgaPFdxxC9Bl70gm/DYjATf5pHK/
2+U0ZSEeTWWHNfTayZd3n6Vz9b6t73XNVk+qesBSB8FuJ1uoxPB7XyjsWOH9XSk5
4S3oXdCJTTdPKRyJdQP4/ub+q4x/pWNwDr0UWH42C3FA5+KWUAUmefdHr9hYJGu9
0A+8sZ8lKb/jf2Ux9ZcOKu3+4iTgfQEwHB3wt5NAgX7wIJZskJ6rPtJE3tKvc9D8
/LXicYMjwlqNcL/vs7D8VMLdb9eDV3F3vK3aZHJvuJ6GfR1gBPPBVMosH9vCwjBa
CuWYH8UyyS72J7tJQC5y/THxgNPgCyc8/DNcdWPC+eKJhb36cmsh9DfgwmknC4OX
VJB9jKaH
=aU5P
-----END PGP MESSAGE-----
```

A questo punto l'operazione inversa di decifratura è del tutto simile. Dopo aver selezionato tutto il messaggio PGP e averlo copiato negli appunti, richiamare l'opzione Clipboard -> Decript and Verify. A questo punto entra in azione la nostra chiave privata e quindi bisogna inserire la passphrase. Fatto questo sullo schermo apparirà un text viewer con visualizzato il in chiaro.



3.4.3 Opzioni per la posta

Con quanto visto finora è possibile inviare un'e-mail con un allegato in modo sicuro ma l'operazione risulta piuttosto laboriosa. Se ben ricordate durante l'installazione di PGP ci era stato chiesto se volevamo installare dei plug-in per alcuni programmi quali Outlook, Eudora o ICQ. Se abbiamo istallato quello relativo ad OutLook, per esempio, possiamo inviare la posta cifrata direttamente del programma Microsoft senza l'utilizzo diretto di PGP. Nella finestra di composizione di un nuovo messaggio sono presenti tre nuovi pulsanti, i primi due (Encrypt messase e Sign message) permettono di applicare la crittografia e la firma digitale al messaggio, il terzo (launch PGPkeys)

richiama semplicemente il PGPkeys. Dopo aver scritto il messaggio di posta e inserito gli indirizzi dei destinatari, basta selezionare uno dei primi due pulsanti appena visti e premere "Invia". PGP avvia subito una ricerca sul keyserver degli indirizzi di posta dei destinatari per trovare le chiavi con cui cifrare il messaggio. In alternativa è possibile interrompere la ricerca e inserire manualmente, tramite la finestra di selezione già vista, le chiavi. Dopo qualche istante il messaggio appare cifrato e viene inviato.

3.4.4 Applicare la firma digitale

Oltre a proteggere il contenuto di un documento PGP permette di applicare ad esso la **firma digitale** con il metodo a doppia chiave visto nelle lezioni precedenti del corso.

Per firmare un file si utilizza il terzo pulsante (**Sign**) presente nel PGPtools, e dopo aver selezionato il file che ci interessa, è necessario inserire la propria passphrase, che come armai avrete capito ci viene chiesta ogni volta che il programma deve utilizzare la nostra chiave segreta. Fatto questo viene creato, nella stessa directory del file di origine, un file apparentemente con lo stesso nome ma con estensione .sig, contraddistinto dall'icona con il foglio firmato. Ora chiunque è in possesso della nostra chiave pubblica può controllare la firma posta sul documento. È sufficiente fare doppio click sul file e, dopo qualche istante, verrà visualizzata una finestra che ci conferma la firma e la data e l'ora in cui è stata applicata.

Se al posto di un file abbiamo la necessità di **firmare un semplice testo**, bisogna seguire una procedura simile a quella usata per cifrare. Una volta copiato negli appunti di windows, dall'icona nella barre delle applicazioni premere Clipboard -> Sign, inserire la passphrase e incollare il contenuto degli appunti in un file di testo, che avrà all'incirca il seguente contenuto:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Prova PGP per HTML.it

-----BEGIN PGP SIGNATURE-----
Version: PGPfreeware 7.0.3 for non-commercial use <HTTP: www.pgp.com>

iQA/AwUBO02Kmaw7fOzd0wvfEQL6SwCg0BTiogk3xubRBMB2FHtvdSkOPKQAoLpO
2i5onhXfJi5TC7FAeW158M7Q
=jXGn
-----END PGP SIGNATURE-----
```

Come si può notare il testo ("Prova PGP per HTML.it") è perfettamente leggibile, infatti abbiamo solo firmato e non cifrato il messaggio. Per controllare la firma il procedimento è inverso. Dopo aver copiato negli appunti il messaggio firmato, premere Clipboard -> Decrypt and Verify e incollare il nuovo contenuto degli appunti in un editor di testo. Il nuovo messaggio ci conferma la firma, la data e l'ora in cui è stata applicata:

```
*** PGP Signature Status: good
*** Signer: Matteo Zinato <MZINATO@HTML.IT>
*** Signed: 12/07/01 13.31.37
*** Verified: 12/07/01 13.32.55
*** BEGIN PGP VERIFIED MESSAGE ***

Prova PGP per HTML.it

*** END PGP VERIFIED MESSAGE ***
```

3.5 Un esempio di PGP

Ho momentaneamente accantonato il programma PGP 9.6 per Windows, scaricato dal sito www.pgp.com come trial version durata 30 giorni (anche per possibile incompatibilità con il sistema operativo installato sul mio vecchio computer).

Ho proceduto a caricare il programma PGP 8.0 reperito in allegato ad una rivista mensile pubblicata anni fa (nella fattispecie “Computer Magazine” Luglio 2003).

La mia chiave pubblica di cui alla mail ugoantin(NoSpam)@tin.it interpretata con “blocco note” è:

-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: PGP 8.0 - not licensed for commercial use: www.pgp.com

mQGiBEZTWKERBADZdnfCTNseJgfRa5AXjcIgdHYC7hJBwtP3HFlHbI38/fGR0Wuf
yqQWs3G9vg6DYlIRbyyKgeMuI2S7FCN+GxMlhhGcttZ0GmwWMCw31zw7qInN3kzJ
DRmG1MHxnMjj7o/OsKzxghl3FXOOoLq4TIF2G6G2t8bppkKObMaXNm56HwCg/6vi
M7199k++I8+3Plap5kXGOrMD/j86ON8A66HPcZzaHlqasTPzdhBHki/RfnENH0XO
eLjOPG3EJGhumdTSmCWR1XwwNcTAlg5WbXBtIaFcnwyRBtABYMaNAc2waP0IquGN
5lI9A1zTW8bCHfZQrq5KUX1sMc13OSo73aOXeErlQFeNgiuESAmWQ6pVRyPULFM6
kulUA/4+UhDLVyZD5yRaHBFpPWgUNsLxmhjH2mmY/Ad/kAO3e0u0G/OTOeYPvlht
++k7r/i8xjFEDPYwToO/Bn/BTrkSDTW9+Lj/gDSEWZIv0R5N3HBTalu27n/eYlxt
zC3JkLw/Jv5ECS7lRnZCUi6T5tFAYkFtV7hgPbP/8gaymS2m9bQfQW5naGlub25p
IFVnbyA8dWdvYW50aW5AdGluLml0PokAWAQQEQIAGAUCRlNYoQgLCQgHAwIBCgIZ

AQUbAwAAAAAKCRBKVhkZCgzx8u9ZAJsFcK2YVKpHxsHfBOfN0v4+tiaxuwCgwkLp
WNdqjP6aE02sYXfeNk0Comi5Ag0ERlNYoRAIAPZCV7cIfwgXcqK61qlC8wXo+VMR
OU+28W65Szgg2gGnVqMU6Y9AVfPQB8bLQ6mUrfdMZIZJ+AyDvWXpF9Sh01D49Vlf
3HZSTz09jdvOmeFXklnN/biudE/F/Ha8g8VHMGHOfMlm/xX5u/2RXscBqtNbno2g
pXI61Brwv0YAWCvl9Ij9WE5J280gtJ3kkQc2azNsOA1FHQ98iLMcfFstjvbzySPA
Q/ClWxiNjrtVjLhdONM0/XwXV0OjHRhs3jMhLLUq/zzhsSlAGBGNfISnCnLWhsQD
GcgHKXrKlQzZlp+r0ApQmwJG0wg9ZqRdQZ+cfL2JSyIZJrqrol7DVekyCzsAAgII
ANOzubHLcYSvXECE0hWPZNl+PA4qKVaLUh9PAz0zYUOyXGkswPX+AwaybXYOAykb
dY3bVzbL18rZD7DeYDmfdzxQFt3WhivXHVVqLMvb9kLRmEeO9siOnrVsDmicTghL
YqHZV1OMVUm8xj2DTp2FwS9dAZ40P8AqwL4HuI6IgyfNgaUuP/VaCa0vZT9tI9y7
QzF1XlNNGmM+dX00fCK2JrdfG/bBJKZF3rT1ad8aFYDZh0nfht5gFDdsWwFkdvkq
Td9dMlQm/MClBPrUSnXPsw2KNs5JmUARIgwLjhjmEFEiF3tiOPK0tfEc9HGwEPjn
PXDzJPwjaBBGdyNwu5fQviKJAEwEGBECAAwFAkZTWKEFGwwAAAAACgkQSlYZGQoM
8fLmkQCg49qFjxCoqHv/5crzPIKB/USPZhcAoOtzM4cMbX/JquWkfNgNbxI6d8tN
=1JHT
-----END PGP PUBLIC KEY BLOCK-----

Per avvalorare le potenzialità del programma si fornisce un esempio di un file di testo in chiaro e la medesima versione cifrata.

Versione in chiaro:

File:	FileDaCifrare.txt
Programma per creazione:	Blocco Note
Dimensione:	63 Byte
Dimensione su disco:	16 KB

Contenuto:

```
Un file da cifrare con PGP.

Prima prova di cifratura.


A U
```

Versione cifrata con PGP:

File:	FileDaCifrare.txt
	(nota: l'estensione del file è la medesima ma l'icona abbinata è diversa e, soprattutto, presenta ben evidente il simbolo di un lucchetto)

Programma per lettura: Blocco Note

Dimensione: 633 Byte

Dimensione su disco: 16 KB

Contenuto:

¨#PGPÁÁN#Â—ô Î â*ƒ*## ¥&#s-í#è¬jF±MRÇÍ##Ñ#ÖÊ#ÎðÄ•tÈ—X-
h,W/s™í@QVo6û˜NR=!Ù&“ý¦}Ã$„Í□èºc#9Ù‰?wjš##½%ˆ(uD##§äÍ##Q=Ï½®n¬

K´#rN^ÀóÍ ¦7\l

bP#ŽÕù‘v%¢ÊÉÁiÑ'ù8Œ°#
Yw*ƒ*##4›#W7

HË##’~##à÷«kA

šY·AORB##pÅ®Ë#•8€*ÓOD5¬j\QU½á

Ð3rp2Ó
˜¢#çÛ}[Þ

Bibliografia, riferimenti e collegamenti esterni

- Simson Garfinkle ha scritto un libro su PGP (O'Reilly and Associates) e la MIT Press ha pubblicato la documentazione di Zimmermann delle diverse versioni di PGP, insieme al loro codice sorcente di volumi separati. PGP ha incluso a lungo la documentazione di Zimmermann in ogni copia. Non era solo necessaria per capire come usare PGP ma era ed è eccellente.
- http://www.pgpi.org - Sito web di Ståle Schumacher. Informazioni sulle versioni open source attualmente disponibili di PGP, incluse le versioni 2.x, e informazioni generali su GPG e PGP. E' l'Home Page internazionale di PGP (Inglese).
- http://www.rcvr.org/varie/pgp/homepgp.htm - Crittografia e privacy (Italiano).
- http://www.pgp.com - PGP Corporation, l'attuale custode, venditore e sostenitore della versione 'ufficiale' di PGP (Inglese).
- http://www.veridis.com/openpgp/en/index.asp - Una versione PGP compatibile con OpenPGP (Inglese).
- http://openpgp.org - Gruppo standard per la versione 'IETF- RFC 2440 di PGP (inglese).
- http://philzimmermann.com - Home Page del creatore di PGP, con numerose informazioni sul programma (Inglese).
- http://it.wikipedia.org/wiki/Pretty_Good_Privacy - La voce dell'enciclopedia on line dedicata a Pretty Good Privacy (Italiano).
- http://www.olografix.org/gubi/estate/tele/node32.html - PGP Crittografia a chiave pubblica per tutti (Italiano).
- http://sicurezza.html.it/guide/leggi/85/guida-crittografia-e-pgp/ - Guida Crittografia e PGP (Italiano).
- http://www.wowarea.com/italiano/aiuto/swpgpit.htm - Crittografare i messaggi PGP (Italiano).
- http://www.wowarea.com/italiano/aiuto/cryptoit.htm - Crittografia (Italiano).
- http://www.ecn.org/crypto/crypto/guidapgp.htm - Guida Introduttiva al Programma di Crittografia a Chiave Pubblica PGP -Pretty Good Privacy- (Italiano).