Introduzione alla crittografia e all'uso del PGP/GnuPG

[by gaxt]

$ iNTRO

$ iNSTALLAZIONE

$ GESTiONE DELLE CHiAVi

$ L'USO Di PGP

/**/

$ iNTRO

[0.0] COSA E' PGP?

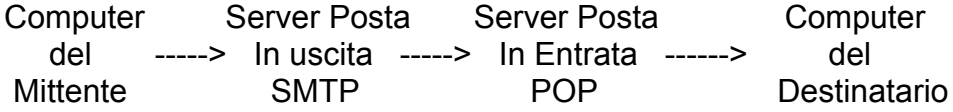
PGP è un software ideato da Philp Zimmermann nel 1991 che permette ad un utente informatico "casalingo" di crittografare i propri documenti o messaggi in modo sicuro, facile e gratuito.
Esistono versioni di questo software per quasi tutti i sistemi operativi esistenti (Win, Linux, Unix, Ms_Dos, VAX/VMS etc.…)

[0.1] PERCHE' USARE LA CRiTTOGRAFiA?

Perché sono affari miei e di nessun altro.
Non importa se state crittando i manuali per la costruzione di una nuova bomba nucleare oppure le ricette segrete di vostra nonna, quello che importa è che: " sono solo fatti miei " come direbbe RazDeGan...

La privacy è un diritto di tutti, potete continuare a spedire mail pensando che solo il destinatario del messaggio legga il suo contenuto, ma non è vero, chiunque mastichi un po' di informatica sa che può non essere così.

Le mail come tutti sanno vengono spedite più o meno nel seguente modo:



La mail viene mandata dal computer del mittente al server di posta in uscita del proprio gestore di posta, che la manda a sua volta al server di posta in entrata che usa il destinatario, la mail poi verrà letta dal destinatario sul proprio sistema.
Capite bene che la mail resta nel server, cioè un computer, dove l' admin o chi per esso può verificarne il contenuto.

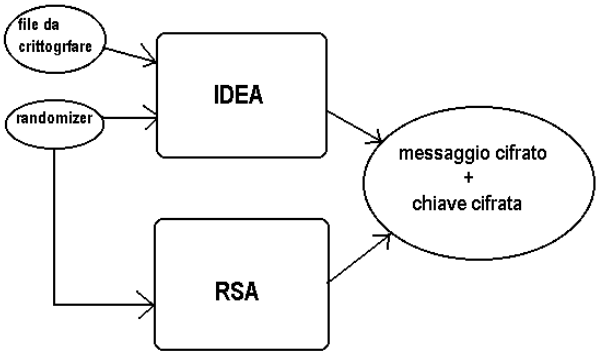
Un utente normale potrebbe letteralmente fottersene di questo e continuare ad usare la posta come l' ha sempre usata, ma noi che forse, e dico forse, abbiamo qualcosa da nascondere dobbiamo per forza utilizzare un sistema di criptazione dei nostri documenti e messaggi.

PGP ci viene in soccorso di tutto questo e ci fornisce una criptazione quasi a livello militare, a costi assolutamente nulli.

[0.2] COME FUNZiONE LA CRiTTOGRAFiA?

Iniziamo col dire che il software PGP è chiamato : "sistema crittografico ibrido", perché utilizza vari tipi di sistemi crittografici.
E' bastato su due algoritmi crittografici, uno a chiave simmetrica e uno a chiave asimmetrica.
I protocolli a chiave simmetrica usati dal PGP sono l'IDEA, CAST e 3-DES
, invece quelli a chiave asimmetrica sono DIFFIE-HELLMANN, RSA e DSA.

Ora vediamo in dettaglio come funzionana PGP con combinando questi due tipi di crittografia:



(Randomizer = generatore di numeri casuali)

[0.2.a.a] DES

Il DES (Data Encryption Standard) è nato nel 1977 dallo sviluppo di un altro sistema crittografico chiamato Lucifer.

Il DES è un' algoritmo cifrato a blocchi, cioè viene cifrato un byte o parola alla volta, Il blocco totale usato per cifrare misura 64 bits, formato da 8 gruppi da 8 bits.
I bits utili per la crittografia sono 64 - 8 = 56 perché l'ultimo bit di ogni gruppo è usato per il controllo.

Quando crittografia un documento con il DES, lui divide tutto in blocchi da 64bits che vengono cifrati uno dopo l'altro.
Ma se il messaggio non raggiunge la dimensione di 64 bits, il DES si può comportare in due modi differenti, o riempie il file di zeri fino a che "pesa" 64 bits, oppure aggiunge dei delle serie di bits casuali fino al riempimento del blocco, questo procedimento è detto "pad".

E' stato ormai sostituito dall'Aes (Andvanced Encryption Standard) che utilizza chiavi simmetriche da 256 bit.

Il DES funziona così:

- Scompone il messaggio in blocchi da 64 bit ciascuno, (8 gruppi da 8bits).
- Poi il blocco da 64bit viene diviso in due da 32bit una metà di destra e una di sinistra.
- Dopo vengono applicate delle funzioni che trasportano e sostituiscono il contenuto tramite delle sottochiavi ricavate dalla chiave originale.Durante ogni passaggio (sono 16) l'output di sx sostituisce quello di dx.
- Alla fine dei 16 passaggi, i due blocchi si riuniscono, poi viene sostituito per invertire la trasposizione iniziale.
- Ricordo che l'algoritmo des utilizza i risultati del passaggio precedente.

- R(i) risultato del passo

- Dx(i) blocco di destra

- Sx(i) blocco di sinistra

- C(i) la sottochiave

Quindi:

- R(i)=Sx(i)Dx(i)

- Sx(i)=Dx(i-1)

- Dx(i)=Sx(i-1) XOR f[Dx(i-1),C(i)]

- Output del blocco di destra deriva da una operazione XOR bit alla volta, del blocco di sinistra e poi dalla funzione f [Dx (i-1) , C (i)], la funzione f dice che:

- il blocco di destra Dx(i-1) viene ingrandito da 32 bit a 48 bit tramite il modulo M. il modulo ingrandito sarà allora: M[Dx(i-1)];

- viene calcolato M[Dx(i-1)] XOR C(i);

- Poi l'output precedente viene diviso in 8 gruppi da 6 bit ciascuno, chiamati B(1),B(2)…B(8), e vi vengono inseriti bit 1-6, 7-12, 13-18, 19-24 ecc...

- ogni gruppo (es B(1)) è usato da una funzione f2 , che restituisce stringhe da 4 bit f2[B(1)]. La funzione funge così: preleva da ogni matrice fissata S-box (Substitution Box) i 4 bit del nuovo blocchetto Sx(i)=f2[B(i)] posizionati in base alle righe e colonne specificate dai 6 bit del corrispondente B(i);

- una volta concatenati gli 8 blocchetti Sx(1), Sx(2)...Sx(8) verranno scambiati di posto ottenendo alla fine P[S(1),..Sx(8)]=f [Dx(i-1), C(i)].

In precedenza abbiamo più volte parlato di sottochiavi C(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, Sx(0) e Dx(0). Dopodiché per 16 volte i semiblocchi vengono spostati a sinistra ottenendo Sx(1), Dx(1), Sx(2), Dx(2) ... Sx(16), Dx(16). Quindi al primo passo l'algoritmo utilizzerà la sottochiave C(1)=P[Sx(1)Dx(1)] dove P al solito indica una permutazione (lo scambio di posto), al secondo C(2)=P[Sx(2)Dx(2)] e al 16° round C(16)=P[Sx(16)Dx(16)]. In questo modo tutte le operazioni effettuate producono sottochiavi C(i) di 48 bit.

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

Avremo quindi:

R(i)=Sx(i)Dx(i)
Dx(i-1)=Sx(i)
Sx(i-1)=Dx(i) XOR f[Sx(i),C(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.

[0.2.a.b] iDEA

iDEA sta per International Data Encryption Algorithm, ed è un algoritmo di codifica che, come il DES, cifra in blocchi da 64 bits con chiave che è però di 128 bits.
E' l'algoritmo a chiave simmetrica usato dal PGP.
Funziona in modo quasi analogo al DES.

Funziona più o meno così:
- Divide il solito blocco di 64 bit in 4 sottoblocchi da 16bit cad.
- Ogni sottogruppo da 16bit viene passato attraverso 8 passaggi in cui intervengono 52 sottochiavi ottenute dalla "big-key" da 128 bit, queste sottochiavi dono quindi da 16 bit.

Le sottochiavi sono create cosi:

- La big-key da 128 viene divisa in 8 parti da 16 bit

- Le cifre della big-key sono spostate di 25 bit a sinistra in modo da generare una nuova combinazione, il cui rgruppo di 8 bit fornisce le prossime 8 sottochiavi

- Poi si continua finchè le 52 chiavi non sono state generate.

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.

[0.2.a.c] BLOWFiSH

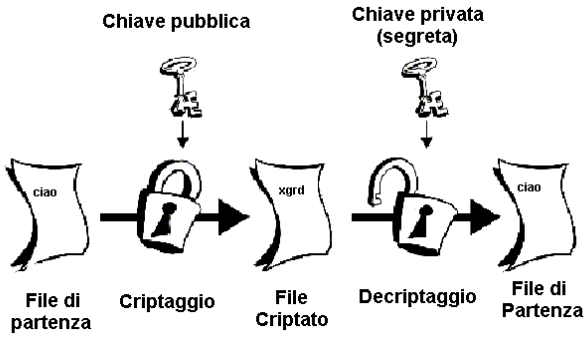
Il blowfish è stato sviluppato da Bruce Schneider ed è un algoritmo basato su un cifrario a blocchi che utilizza varie tecniche tra qui funzioni non invertibile, che fanno del blowfish IL SISTEMA CRITTOGRAFICO PIU' SICURO OGGI ESISTENTE.
Le chiavi che usa sono di dimensioni massimo di 448 bit e i blocchi sono da 64 bit.
Oltre ad essere il sistema più sicuro, ad oggi infatti non sembra esistano modi per aggirarlo, è anche il più veloce, molto più del des e dell'idea.

[0.2.b.a] DiFFiE-HELLMANN

Il Diffide-Hellmann (D.H.) è un sistema di codifica, datato 1976, basato sull'uso di due chiavi, una pubblica e una segreta, ciò significa che un documento cifrato può essere letto in chiaro solo se si dispone di tutte e due le chiavi usate.

La chiave pubblica viene distribuita il più possibile in giro invece la chiave segreta (o privata), NON DEVE essere assolutamente resa pubblica.
Quando crittografiamo un documento con PGP, lui crittografa con la chiave pubblica del destinatario, e poi SOLO lui che possiede la sua chiave privata potrà decriptare il messaggio.

Lo schema riassuntivo sotto rende meglio l'idea di come funziona la crittografia a doppia chiave :



[0.2.b.b] RSA

Questo algoritmo è stato sviluppato nel 1977 da tre ricercatori del MIT (Massachusetts Institute of Technology) di nome Ron Rivest, Adi Shamir e Leonard Adleman e dalle loro iniziali prende il nome R.S.A..

La caratteristica di questo sistema, che lo rende davvero sicuro, è la difficoltà di generare la chiave segreta basandosi su quella pubblica.

Ricordatevi comunque che più la chiave è grande, più è difficile crackare questo sistema. Solitamente per un utente "normale" si usano chiavi di 1024Bits, potete creare anche chiavi di 2048Bits, ma andrà a discapito della velocità.

Ciò vuol dire che più una chiave è grande, più è lenta nel crearsi, nel crittografare etc.. Perciò il mio consiglio è di trovare un compromesso tra velocità e sicurezza, 1024Bits saranno più che sufficienti per un semplice cittadino.

RSA funziona in questo modo:

1. Calcolare il valore di c, prodotto di a e b 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: c = a*b = 7*5= 35

2. Calcolare il valore di z =(a-1)*(b-1).
   z = (a-1)*(b-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 nkbit, 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.

[0.3] L'ALTERNATiVA : GNUPG

Esiste un progetto software simile a PGP, ma con la notevolissima differenza di essere OpenSorce, che si chiama GNUPG o semplicemente GPG.
GPG è nato nel 1997 dalla mente di Werner Koch, l'obbiettivo (raggiunto) di questo progetto era creare un software alternativo al PGP completamente OpenSource così da permettere agli utenti di poter personalizzare in tutto e per tutto il proprio software.

GnuPG non esiste solo per GNU/Linux, infatti sono state sviluppate varie versioni per i più importanti sistemi operativi : Win32, *BSD, Sun OS, IRIX ecc..

GPG utilizza vari sistemi crittografici (ElGamal, DSA, RSA, AES, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 e TIGER).

/**********/

$ iNSTALLAZiONE

[1.0] iNSTALLAZiONE iN MS_DOS

Vediamo ora in specifico come installare PGP 2.6.3i sulla nostra macchina, iniziamo con MS_DOS.
Installiamo questo perché è ritenuto il più stabile e sicuro anche se un po' vecchiotto.

- Per prima cosa scaricatevi il file pgp263i.zip da www.pgpi.org.
- Createvi una dir C:\ pgp con il comando md c:\pgp
- Copiateci il file scaricato
- Unzippatelo con il comando pkunzip -d pgp263i.zip
- Dentro troverete un'altro file zippato (pgp263ii.zip)
- Scompattatelo (pkunzip -d pgp263ii.zip)

- Adesso aprite il file C:\autoexec.bat con EDIT
- Aggiungete sotto:

```
SET PGPPATH=C:\PGP
SET PATH=C:\PGP
SET TZ=MET-1DST
```

- Salvate e riavviate

Bene dopo aver riavviato la macchina digitate nel dos c:\ pgp –h e se vi si aprirà la pagina con i comadi di PGP avete fatto tutto bene.

[1.1] iNSTALLAZiONE iN WIN32

- Per prima cosa scaricatevi il file pgp263i.zip da www.pgpi.org.
- Decomprimetelo con WinZIP
- Ci sarà un altro file compresso pgp263ii.zip
- Decomprimete anche questo
- Copiate il contenuto del file decompresso nella dir c:\pgp
- Adesso aprite il file C:\autoexec.bat con BloccoNote
- Aggiungete sotto :

```
SET PGPPATH=C:\PGP
SET PATH=C:\PGP
SET TZ=MET-1DST
```

- Salvate e riavviate

Provate anche qui ad aprire il prompt del DOS e digitare pgp –h, se vedrete apparire la lista dei comandi di PGP avete fatto tutto bene.

[1.2] iNSTALLAZiONE iN GNU/LiNUX di GNUPG

Con linux useremo gpg, anche se avremmo potuto usare pgp, ma a noi piace sapere di più…..

- Per prima cosa scaricatevi i tarball dell'ultima versione di gpg. Da: www.gnupg.org
- Decomprimeteli e dearchiviateli con il comando: tar xvfz gnupg-versione.tar.gz
- Entrate nella dir creata e inserite il comando: ./configure.
- Fatto questo compilate con il comando: make
- Il passo successivo sarà digitare: make install
- Se tutto è andato nel verso giusto proviamo a digitare: gpg - -gen-key
- Ora possiamo creare le nostre chiavi gpg.

La procedura di creazione delle chiavi è molto simile a quella di pgp, con la differenza di non usare pgp –kg ma useremo gpg - -gen-key.

Qui sotto scrivo i comandi gpg più usati:

Gestione del keyring:

Estrarre una chiave dal proprio keyring:
gpg –a –export –o chiave.asc

Aggiungere una chiave dal proprio keyring:
gpg –a –import nomechiave.asc

Visualizzare le chiavi del keyring:
gpg –list-keys

Uso semplice di gpg:

Cifrare un file o documento
gpg –e UserIDdestinatario nomefile

Decifrare un file gpg
gpg –d nomefile

[1.3] PGP iN iTALiANO

E' possibile tradurre pgp dall'inglese in quasi tutte le lingue, tra cui naturalmente l'Italiano tradotti da Marco Giaiotto.
I file sono chiamati language.it e it.hlp

Per poterli utilizzare bisogna :

- scaricarli da www.pgpi.org/docs/it_pgp.zip
- scompattare il file e copiarli in C:\pgp
- aprire il file config.txt e modificare la 29 riga dove c'è scritto language = en, sostituite en con it
- rinominare il file language.txt in language.eng
- rinomimate language.it in language.txt
- poi provate a digitare pgp dalla shell del dos

[1.4] WiNFRONT

I winfront sono dei programmi che permettono di usare pgp con un interfaccia grafica, al posto della normale riga di comando.
Ce ne sono molti da scaricare in rete, io con win98 uso PGPWinFront3.1, rende più facile e veloce il lavoro di crittografia e decrittografia dei documenti, poiché sostituite completamente la tastiera con il mouse.
Bastano pochi click per crittografare, decrittare, firmare ecc.. ogni file che voi vogliate.

Lo potete scaricare da: ftp://ftp.cnr.it/pub/PC-IBM/win3.cica/util/

Lo trovate nella directory: /pub/PC-IBM/win3.cica/util/pwf31.zip.

/***/

$ GESTiONE DELLE CHiAVi

[2.0] GENERARE LA COPPiA Di CHiAVi

La prima cosa che dobbiamo fare quando abbiamo installato PGP è di creare la coppia di chiavi RSA.
Andiamo nel prompt dei comandi e scriviamo:

pgp -kg (-kg significa appunto **K**ey **G**eneration)

Si aprirà una schermata che vi da la possibilità di scegliere tra 3 diversi formati.
Uno da 512Bits usato da utenti “normali”, e che quindi non hanno bisogno di grande sicurezza ma preferiscono la velocità.
Uno da 768Bits usato da utenti medi, con un buon rapporto tra velocità e sicurezza.
E poi il livello 3 con chiavi da 1024Bits, con un livello di sicurezza “militare” ma con velocità abbastanza ridotte.
Il mio consiglio è sicuramente di scegliere una chiave da 1024Bits.

Scelta la dimensione della chiavi verrà aperta una schermata dove dovete inserire il vostro UserID e la vostra mail.

Inseriteli così : Mario Rossi <m_rossi@hacktheworld.org”>
Naturalmente sostituite Mario Rossi con Il vostro nome o nickname e la mail con la vostra, quello che importa è che la mail sia racchiusa tra parentesi “spigolose” poi confermate con enter.

Il terzo punto ci richiederà di inserire la frase chiave che serve a proteggere la vostra chiave segreta RSA.
Potete inserire qualunque carattere spazio o punteggiatura.
La frase è CASE SENSATIVE cioè tiene conto delle maiuscole/minuscole, quindi scrivere:
La Mia FrasE ChIAve non corrisponde a : la mia frase chiave.

Ricordatevela sempre perché vi servirà ogniqualvolta dobbiate usare la vostra chiave segreta per decrittografare i messaggi.

Nell’ultimo punto ci verrà richiesto di inserire caratteri casuali fino a che il contatore non arriva a 0.
Basta, avete generato la vostra chiave.
Vi ricordo, NON DITE ASSOLUTAMENTE A NESSUNO LA VOSTRA CHIAVE SEGRETA.
Altrimenti sarebbe inutile crittografare….

[2.1] AGGiUNGERE UNA CHiAVE AL NOSTRO KEYRiNG

Bene, adesso avete la vostra coppia di chiavi, ma senza le chiavi pubbliche di altre persone è inutile utilizzare PGP.
Mettiamo caso che voi abbiate per le mani un documento riservatissimo da mandare ad un vostro amico, come abbiamo detto per farlo leggere a lui abbiamo bisogno della sua chiave pubblica, altrimenti non lo possiamo crittografare.
Come possiamo aggiungere al nostro portachiavi la sua chiave pubblica?
Il nostro amico ci ha fornito la sua chiave che si chiama chaivepubblica.asc, noi andiamo nel prompt dei comandi e digitiamo:

pgp -ka chiavepubblica.asc [pubring.pgp] (-ka significa appunto **K**ey **A**dd)

Possiamo omettere [pubring.pgp] perché è il portachiavi publico di default.
Ma se in un futuro creeremo un altro portachiavi pubblico specificheremo il suo nome.

Se la chiave che vogliamo inserire è già presente nel pubring non succederà niente, la chiave non verrà modificata.
Se la chiave che vogliamo inserire è presente per esempio in un file di testo digiteremo:

pgp –ka nomefile.txt

PGP cercherà una chiave valida nel file e l'aggiungerà nel nostro pubring.

[2.2] ELiMiNARE UNA CHiAVE DAL KEYRiNG

Se vogliamo eliminare una chiave esistente nel nostro keyring, non dobbiamo far altro che andare nel solito prompt dei comandi e digitare:

pgp -kr UserID [portachiavi] (-kr significa **K**ey **R**emove)

[portachiavi] può essere omesso, ma se vogliamo eliminare una nostra chiave segreta dobbiamo specificare [secring.pgp].

[2.3] COPiARE UNA CHiAVE

Se vogliamo distribuire la nostra chiave pubblica ad un amico (o non), dobbiamo usare il seguente comando:

pgp –kx UserID

Digitando questo ci verrà chiesto in quale file copierà la chiave, es. C:\chiave.asc
PGP copierà la chiave corrispondente all'UserID indicato senza cancellare la chiave di partenza.

[2.4] VERiFiCARE iL CONTENUTO DEL KEYRiNG

Come facciamo se vogliamo vedere che chiavi ci sono nel nostro portachiavi?
Semplice, con il comando:

pgp –kv (che sta per pgp **K**ey **V**iew)

Con questa istruzione, pgp, restituisce a video tutte le chiavi contenute nel file public.pgp. Se invece vogliamo vedere le nostre chiavi segrete, dobbiamo inserire dopo –kv, il percorso del file portachiavi secring.pgp. es. c:\pgp\secring.pgp, quindi:

pgp –kv c:\pgp\secring.pgp

[2.5] REVOCARE UNA CHiAVE

Cosa fare se qualcuno sia venuto a conoscenza della vostra frase segreta?
Dobbiamo revocare la chiave, cioè creare un “certificato di chiave compromessa o revocata”.

pgp –kd MyUserID

Questo è il comando usato per creare il certificato, ora non dobbiamo fare altro che distribuirlo in giro ai nostri amici, così che loro sappiano di non utilizzare più la nostra chiave pubblica.

[2.6] FiNGER PRiNT

Il finger print è un hash formato da 32 cifre in hexadecimale che identifica la chiave in modo univoco.

Se ricevete la chiave di qualcuno, potete verificarne l’autenticità confrontando il fingerprint che già avete con quello che estraerete dalla chiave appena ricevuta, se le due chiavi coincidono, significa che la chiave è sicuramente quella giusta.

pgp -kvc UserID

Per vedere il proprio fingerprint dobbiamo digitare:

pgp -kvc MyUserID

[2.7] LiSTA COMANDi GESTiONE CHiAVi

Qui sotto trovate la lista di comandi usati per la gestione delle chiavi, premetto che non li ho spiegati tutti prima, ma sono molto intuitivi, io ho spiegato solo quelli essenziali.

Se vogliamo generare una nuova coppia di chiavi:
pgp -kg

Se vogliamo aggiungere una chiave di altri nel nostro pubring:
pgp -ka filechiavi [keyring]

Se vogliamo copiare la nostra chiave per poi distribuirla ad altre persone:
pgp -kxa ID filechiavi [keyring]

Se vogliamo rimuovere una chiave dal portachiavi o pubblico o segreto:
pgp -kr ID [keyring]

Se vogliamo vedere che chiavi abbiamo nel portachiavi:
pgp -kv[v] [ID] [keyring]

Se vogliamo verificare le firme e vedere cosa c'è nel pubring:
pgp -kc [ID] [keyring]

Se vogliamo firmare e certificare le firme di latri nel nostro keyring:
pgp -ks suo_ID [-u mio_ID] [keyring]

Se vogliamo rimuovere firme da un ID:
pgp -krs ID [keyring]

Se vogliamo modificare la frase segreta:
pgp -ke ID [keyring]

Se vogliamo revocare la nostra chiave per sempre, rilasciano un certificato di compromissione:
pgp -kd mio_ID

Se vogliamo abilitare/disabilitare una chiave del keyring:
pgp -kd ID

/**********/

$ L'USO Di PGP

[3.0] CRiTTARE UN FiLE

Questa la parte più importante della guida, dove imparerete a crittare/decrittare/firmare i documenti.

[3.1.a] PER UN DESTiNATARiO SiNGOLO

Il comado che utilizzeremo per crittografare i messaggi e i documenti sarà:

pgp –e file UserID

Es. pgp –e lettera.doc Michele

Il comando crea un file chiamato lettera.pgp
Se il messaggio verrà spedito in una e-mail allora dobbiamo converirlo in ASCII "radix-64" , per fare ciò dobbiamo inserire il parametro a dopo –e

Es. pgp –ea mail.txt Michele
Adesso abbiamo il file mail.pgp che è pronto per essere spedito via mail.

[3.1.b] PER PiU' DESTiNATARi

Se vogliamo crittografare un file che verrà ricevuto da più destinatari dobbiamo far:

pgp –e file.txt Michele Matteo Fabio

Questo file verrà crittografato e potrà essere letto da qualunque di questi destinatari.

[3.1.c] CRiTTOGRAFiA CONVENZiONALE

Se ad esempio vogliamo crittare un file per noi stessi, così per sicurezza e decrittarlo quando lo vogliamo noi dobbiamo fare uso della crittografia convenzionale.
Il file verrà crittato cosi:

pgp –c filediesempio.doc

Sarà così creato un file chiamato filediesempio.pgp che potrà essere decrittato da noi inserendo poi la nostra pass phrase.

[3.2] FiRMARE UN FiLE

Firmare significa dare una certificazione dell'autenticità di un file, per assisicurare che è stato manipolato solo da noi che lo abbiamo creato.

Per firmare un file in chiaro con la vostra chiave segreta, battete:

pgp -s filetesto [-u mio_ID]

Si noti che le [parentesi] indicano che un parametro e' opzionale, quindi non inseritele nel comando.

Questo comando genera un file firmato chiamato filetesto.pgp. Esempio:

pgp -s lettera.txt -u Bob

PGP cerchera' un certificato di chiave nel vostro portachiavi segreto che contenga la stringa Bob nel campo ID. Naturalmente vi dovete chiamare Bob... La ricerca non tiene conto di maiuscole/minuscole. Se una chiave segreta corrispondente e' trovata, la si usa per firmare "lettera.txt" e produrre un file firmato "lettera.pgp".

Se non mettete il vostro ID, verra' usata la prima chiave del vostro portachiavi segreto.

PGP tenta di comprimere il messaggio dopo averlo firmato. E' quindi probabile che il file firmato sia piu' piccolo dell'originale, il che e' utile per le applicazioni di archivio. Pero', questo rende il file illeggibile anche se il messaggio originale era in chiaro. Sarebbe bello poter generare un file firmato, ma leggibile. Soprattutto se volete spedire un messaggio firmato via e-mail.

(tratto dal manuale d'uso, non riuscivo a dirlo meglio di così…)

[3.3] FiRMARE E CRiTTARE

Per firmare un file in chiaro con la vostra chiave segreta, e poi cifrarlo con la chiave segreta del destinatario dobbiamo utilizzare il comando:

pgp -es filetesto dest_ID [-u mio_ID]

Si noti che le [parentesi] indicano che un parametro e' opzionale, quindi non inseritele nel comando.

Questo esempio produce un file annidato chiamato filetesto.pgp. La vostra chiave segreta per firmare e' trovata automaticamente nel portachiavi segreto. La chiave pubblica del destinatario e' trovata automaticamente nel portachiavi pubblico. Se non inserite il dest_ID nel comando, vi sara' richiesto.

Se non inserite il vostro ID, verra' usata la prima chiave del vostro portachiavi segreto.

[3.4] DECRiTTARE (E VERiFiCA FiRMA) Di UN FiLE CRiTTATO

La decrittazione è l'altro punto fondamentale della crittografia.
In questa sezione spiegheremo come decrittare un file crittato e come verificare la firma di questo file.

Per decrittare un file usiamo il comando:

pgp filecrittato.pgp

E ci verrà chiesta la pass phrase, inseriamola e…il pgp ci restituirà un file senza estensione chiamato filecrittato.
Faremo così per ogni file crittato.

Se il file era stato anche crittato il pgp automaticamente verificherà anche la firma apportata al messaggio.

/**********/

$ OUTRO

[4.0] CREDiTS

Autore : gaxt
Mail : gaxt@hacari.org
Web : colander.tk
PGP key : colander.altervista.org/keys/gaxt.asc

Data : dal 1o/1o/2oo4 al 23/1o/2oo4
Musica Ascoltata : Metallica (kill'em all, …and justice for all, ride the lighting)
Slayer (reign in blood)
Slipknot (iowa, slipknot) and much much more.....
Consumo : beh in 13 giorni ho mangiato solo un paio di volte....he he he...

[4.1] SALUTi E FUCK

Tnx to: OverIP, Traktopel, Depippis, Peeter, JJ, [Sky.Witch], <guari>, Linus Torsvald, Richard Stallmann, Raoul Chiesa, Alor, Naga e a tutti i grandi hacker italiani (e non.,) poi un grazie speciale và chiaramente chi legge questo documento.
Fuck to: tutti quelli che non credono in me, a Silvio, a George W. , alla mia scuola, alla mia ex profe di informatica (muori vecchia bastarda), al Grande Fratello, e soprattutto a quelli che pensano che, chi ascolta metal vada in giro ad ammazzare la gente…Fuck you all

[4.2] DiSCLAiMER

Tutto quello che ho scritto potrebbe essere (o è) una marea di cazzate, capitemi, non sono certamente un hacker e forse neanche un newbie quindi non crediate ciecamente a ciò che ho scritto nella guida.
Se vi serve aiuto, oppure volete solo mandarmi affancu** mailatemi a gaxt@hacari.org

[4.3] BiBLiOGRAFiA

Kriptonite:
http://www.ecn.org/kriptonite

La crittografia nell'era Internet:
http://enricozimuel.net

Manuale d'usu PGP:
http://www.pgpi.org/docs/italian.html

Guida pratica a PGP:
http://www.pgpi.org/docs/g_pgp952.htm#ref0

The PGP attack FAQ:
ftp://ftp.infonexus.com/pub/Philes/Cryptography/PGP/PGPattackFAQ.gz

Sicurezza.html.it (algoritmi des e rsa)
http://sicurezza.html.it

"Crittografia" di Andrea Sgarro, Franco Muzzio Editore.

"Crittografia - Principi, Algoritmi, Applicazioni" di P. Ferragina e F. Luccio, Bollati Boringhieri Editore.

[4.4] MY KEY

Bits/KeyID User ID
1024/5BE4A0D9 gaxt <gaxt@hacari.org>

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3i

mQCNA0EHv4AAAAEEAMAqk04CG3SLJ2ABo0Dj28y5jgh2TAHER5rTOkx2YGNTo+nz
EC5IcAMX4MA2Ta7ue48U4ux/ew5TqPHtx65xHl4PTLb1DmZ2B/9Yuwnd30Xb7kM4
wP+9b43ioHwuc8qRvJjMOSFwxZmM0ZobuSescB3/xnWwneK7QY/vKxdb5KDZAAUR
tBZnYXh0IDxnYXh0QGhhY2FyaS5vcmc+iQCVAwUQQQe/gI/vKxdb5KDZAQGZKwP9
GsBECcDiK6NxZjh1XjHhCmiV9/MTLnhcg2VTZZKWtY8MUZ3BoOaGROxNorDnr3Qy
j0ANPqHB3GfGCYsZpAIvgrHwNjTgkVZZ+n7KSf9ayDE23mcYvhwdqW1eNe2YgKyB
tgPVeEbNQZ7rP6qzOJRU6ILtuPcAN7Uwf0z5mMnAUJI=
=oh0N
-----END PGP PUBLIC KEY BLOCK-----

FingerPrint: B1 40 C3 D8 7E 98 2D D7 6E D0 BE F7 C2 AC 33 33