Pretty Good Privacy
In arte PGP

Relazione realizzata da:
Frigo Andrea
Fusaro Elisabetta
Magalini Fabio

L'esigenza di sicurezza esiste da sempre ma con l'avvento e la popolarità crescente di Internet questo bisogno si è esteso ad un gruppo di persone molto più ampio. La trasmissione di un'e-mail attraverso Internet è, potenzialmente, molto meno sicura di una lettera spedita attraverso l'ufficio postale o il fax. Il contenuto dei vostri testi che transitano normalmente in chiaro da un mittente ad un destinatario possono essere da chiunque letti o modificati a piacere. La crittografia tutela i nostri diritti alla segretezza.
Bisogna prima precisare che nessun crittosistema, PGP incluso, può proteggere informazioni che sono disponibili (o possono essere ottenute) in altro modo. Questi sistemi non possono impedire che le informazioni che l'utilizzatore vuole proteggere possano essere ottenute tramite intercettazione, ricatto o la semplice ricerca di documenti inavvertitamente gettati nella spazzatura. La maggioranza degli utenti di crittosistemi non prende in considerazione questo tipo di attacchi, da cui è molto più difficile difendersi che dai più "equi" attacchi basati solamente sulla crittoanalisi. La vera sicurezza delle informazioni richiede utenti consapevoli dei diversi rischi sistemi crittografici progettati e realizzati nel modo più attento possibile. La sicurezza assoluta è probabilmente impossibile.
La crittografia è usata per proteggere le informazioni e per mantenerle confidenziali. Un documento che è stato cifrato deve, in primo luogo, essere decifrato per essere comprensibile dagli applicativi o per essere in forma leggibile. Il software di crittografia è esistito per molti anni ed il metodo più comune di crittografia doveva proteggere un documento con una parola d'accesso e per decifrare il documento si usava la stessa parola d'accesso. La crittografia con parola d'accesso è facile da utilizzare ma spesso ne è sconsigliato l'utilizzo. Ciò capita perché la maggior parte degli utenti che utilizzano il software (al contrario di ciò che raccomandano tutti i manuali di crittografia), usano parole d'accesso facili da indovinare, come il nome della moglie, data di un compleanno, ecc. Inoltre, il mittente deve comunicare in maniera segreta la sua parola d'accesso al destinatario che desidera decifrare il documento, e come può farlo?
Esistono sostanzialmente due macro famiglie di crittografia: la crittografia simmetrica e la crittografia asimmetrica (più conosciuta come crittografia a chiave pubblica).

Crittografia simmetrica

La crittografia simmetrica è quella più semplice da comprendere, e si basa su un algoritmo che modifica i dati in base a una chiave (di solito una stringa di qualche tipo), che permette il ripristino dei dati originali soltanto conoscendo la stessa chiave usata per la cifratura. Per utilizzare una cifratura simmetrica, due persone si devono accordare sull'algoritmo da utilizzare e sulla chiave.
Uno schema di crittografia simmetrica (Fig. 1) è caratterizzato dalla proprietà che, data la chiave di cifratura "e", sia facilmente calcolabile la chiave di decifratura "d". Nella pratica, tale proprietà si traduce nell'utilizzo di una sola chiave sia per l'operazione di cifratura che quella di decifratura. La forza della crittografia simmetrica è dunque riposta nella segretezza dell'unica chiave utilizzata dai due interlocutori che la usano, oltre che nella grandezza dello spazio delle chiavi, nella scelta di una buona chiave e nella resistenza dell'algoritmo agli attacchi di crittoanalisi.
Generalmente gli algoritmi di crittografia simmetrica sono molto più veloci di quelli a chiave pubblica, per questo vengono usati in tutte le operazioni di cifratura che richiedono performance. Oltretutto i cifrari a crittografia simmetrica permettono l'uso di uno spazio delle chiavi lungo quanto il messaggio: ad esempio nella codifica XOR (algoritmicamente semplicissima) è possibile scegliere una chiave lunga quanto il messaggio da cifrare, rendendo il messaggio cifrato assolutamente sicuro.

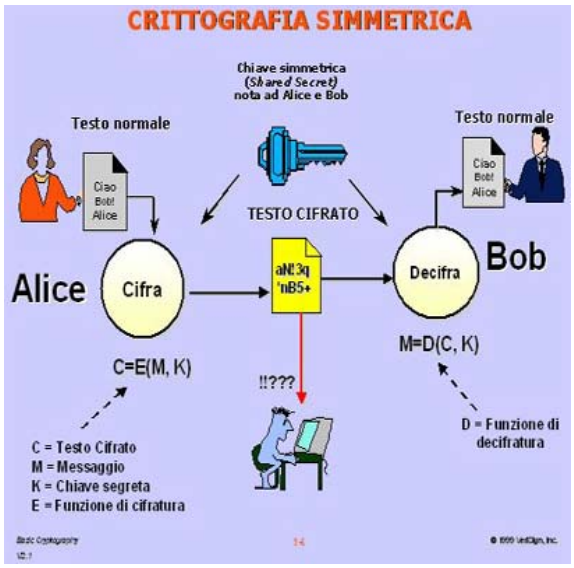


Fig.1

Crittografia asimmetrica

La **crittografia asimmetrica**, conosciuta anche come **crittografia a chiave pubblica** è un tipo di crittografia dove ad ogni "attore" coinvolto è associata una coppia di chiavi:

- la chiave privata, personale e segreta, viene utilizzata per decodificare un documento criptato;
- la chiave pubblica, che deve essere distribuita, serve a crittare un documento destinato alla persona che possiede la relativa chiave privata.

Questo è un metodo molto più complesso rispetto al precedente, che però ha il vantaggio di essere più pratico quando riguarda la comunicazione con molte persone. Il principio di funzionamento si basa sul fatto che esistono due chiavi complementari, e un algoritmo in grado di cifrare con una chiave e di decifrare utilizzando l'altra. In pratica, la cifratura avviene a senso unico attraverso la chiave di cui dispone il mittente di un messaggio, mentre questo può essere decifrato solo con l'altra che possiede solo il destinatario. Le due chiavi vengono chiamate *chiave pubblica* e *chiave privata*, attribuendogli implicitamente un ruolo specifico. Dal punto di vista pratico, chi vuole mettere i propri interlocutori in condizioni di inviare dei messaggi o altri dati cifrati che poi possano essere decifrati solo da lui o da lei, dovrà generare una propria coppia di chiavi, e dovrà distribuire la propria chiave pubblica.

Chi invece intende inviare informazioni cifrate a questa persona, dovrà usare la propria chiave privata e la chiave pubblica diffusa dal destinatario. E' evidente pertanto che la chiave privata deve rimanere segreta a tutti, tranne che al suo proprietario; se venisse trafugata permetterebbe di decifrare i messaggi che potrebbero essere intercettati. Per questa ragione, il proprietario di una coppia di chiavi asimmetriche deve essere la stessa persona che le genera.

La coppia di chiavi pubblica/privata viene generata attraverso un algoritmo (ad esempio RSA o DSA) a partire da dei numeri casuali. Gli algoritmi asimmetrici sono studiati in modo tale che la conoscenza della chiave pubblica e dell'algoritmo stesso non siano sufficienti per risalire alla chiave privata. Tale meccanismo è reso possibile grazie all'uso di funzioni unidirezionali. In realtà, in molti casi, l'impossibilità di risalire alla chiave privata non è dimostrata matematicamente, ma risulta allo stato attuale delle conoscenze in matematica e della potenza di calcolo disponibile, un problema complesso.

Per firmare un'e-mail o un file che trasmettete ad altri, utilizzerete la vostra chiave privata per l'autenticazione. I destinatari potranno allora usare la vostra chiave pubblica per determinare se siete stato realmente voi a trasmettere il messaggio o se il contenuto è stato alterato durante il trasferimento. Per contro, quando qualcuno vi trasmette un documento o un file con la sua firma digitale, userete la sua chiave pubblica per controllare la firma e per verificare che nessuno alteri il relativo messaggio.

Fig.2

Come mostrato nella Fig.2, il mittente può cifrare il messaggio soltanto a patto di possedere una chiave privata e di conoscere la chiave pubblica del destinatario. Ma come è possibile distribuire la propria chiave pubblica? Attraverso certificati digitali firmati e rilasciati da un'Autorità di Certificazione universalmente accettata e ritenuta affidabile. La prima cosa da fare per ottenere un certificato digitale è generare una *coppia di chiavi*; mentre la *chiave privata* resterà residente sul Personal Computer della persona richiedente, la *chiave pubblica* verrà inviata all'Autorità di Certificazione, che provvederà a firmarla con la propria chiave privata ed a distribuirla all'interno del certificato digitale rilasciato al richiedente.
Quando si ottiene una qualsiasi chiave pubblica, bisogna controllare che la chiave appartenga realmente al proprietario dichiarato. Una volta che si è sicuri che la chiave pubblica è valida, si può applicare un certificato a quella chiave. Molta gente può applicare più certificati alla stessa chiave pubblica. Ci sono parecchi modi per controllare l'autenticità di una chiave, cioè, se la chiave realmente appartiene al relativo proprietario. A meno che si riceva fisicamente una chiave, per esempio, per mezzo di un dischetto, o a meno che realmente ci si fidi della persona che ha applicato il suo certificato su una chiave pubblica, il modo migliore per verificare l'autenticità di una chiave consiste nel confrontare la relativa impronta digitale (fingerprint). Contattare il proprietario indicato dalla chiave che si desidera verificare; chiedergli di comunicarvi l'impronta digitale della sua chiave privata abbinata a quella della sua chiave pubblica. Dopo questa verifica potete decidere se certificare la sua chiave pubblica.

Fig.3

Il certificato digitale è dunque un documento elettronico che, oltre a contenere i dati essenziali dell'intestatario, contiene la sua chiave pubblica. Per questa ragione il certificato digitale identifica un individuo o un'organizzazione.

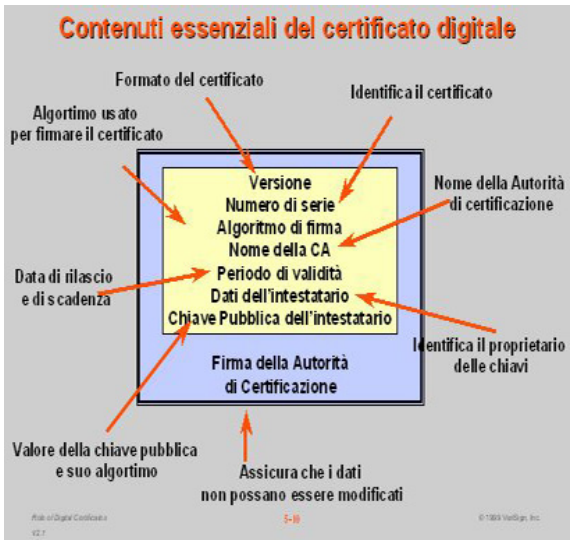


Fig.4

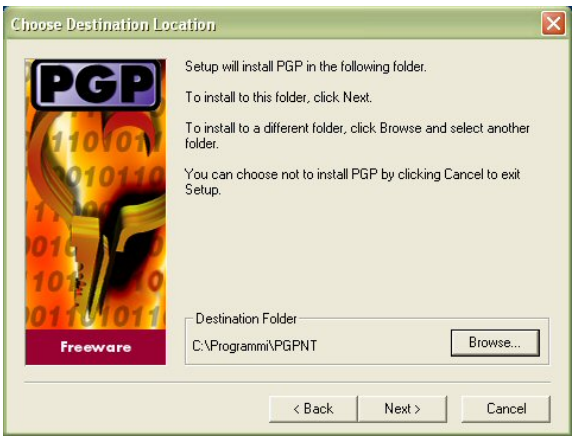
La differenza sostanziale tra le due soluzioni, sta nel fatto che nella tradizionale crittografia simmetrica viene utilizzata un'unica chiave sia per codificare, sia per decodificare i messaggi. Le informazioni (la chiave e l'algoritmo) necessarie per chi deve inviare il messaggio sono quindi identiche a quelle necessarie a chi deve riceverlo. Per concordare una chiave con il proprio interlocutore c'è bisogno di mettersi preventivamente in contatto con lui incontrandolo di persona, telefonandogli, scrivendogli una lettera, mandandogli un messaggio o in qualsiasi altro modo. In qualsiasi caso, esiste il pericolo che la chiave venga intercettata durante il tragitto, compromettendo quindi l'intero sistema comunicativo.
La crittografia a chiave pubblica permette invece a due (o più) persone di comunicare in tutta riservatezza senza usare la stessa chiave e anche se non si sono mai incontrate precedentemente.
In realtà il problema della sicurezza riguardante la comunicazione non è del tutto risolto con la crittografia a chiave pubblica, perché non si può essere certi che la chiave (per esempio una chiave

presente sul keyserver) appartenga davvero alla persona nominata nell'intestazione della chiave stessa. Una soluzione resta sempre il contatto fisico tra i due interlocutori, i quali, scambiandosi le chiavi pubbliche hanno una reciproca autenticazione.
Grazie alla forte sicurezza delle funzioni multiple, la crittografia a chiave pubblica come quella utilizzata dallo standard OpenPGP è probabile che sostituisca la crittografia convenzionale in molti ambiti.

Pretty Good Privacy (PGP) è un programma che permette di usare autenticazione e privacy crittografica. Nelle sue varie versioni è probabilmente il crittosistema più usato al mondo. Questo è un programma, sviluppato da Paul Zimmermann e distribuito gratuitamente in rete, con cui è possibile cifrare praticamente tutti i tipi di file, anche se questi non fossero destinati a essere inviati tramite una rete telematica. Tuttavia la sua area di applicazione più comune è la cifratura della posta elettronica, scopo per cui, del resto, PGP è nato.
La chiave pubblica del destinatario serve al mittente per cifrare una chiave comune (detta anche chiave segreta o convenzionale) per un algoritmo di crittografia simmetrica; questa chiave viene quindi usata per cifrare il testo in chiaro del messaggio. Il destinatario di un messaggio protetto da PGP lo decifra usando la "chiave di sessione" con l'algoritmo simmetrico. La chiave di sessione è inclusa nel messaggio in maniera criptata e viene decifrata usando la chiave privata del destinatario. L'utilizzo di due cifrature è giustificato dalla notevole differenza nella velocità di esecuzione tra una cifratura a chiave asimmetrica ed una a chiave simmetrica (l'ultima è generalmente molto più veloce).
Per testare la bontà di PGP abbiamo deciso di prelevare da internet due dei tanti software presenti in rete che implementa questo standard, scaricandoli dai siti specificato in fondo a questa relazione. I programmi scaricati sono stati testati sul sistema operativo Windows XP o su UBUNTU.

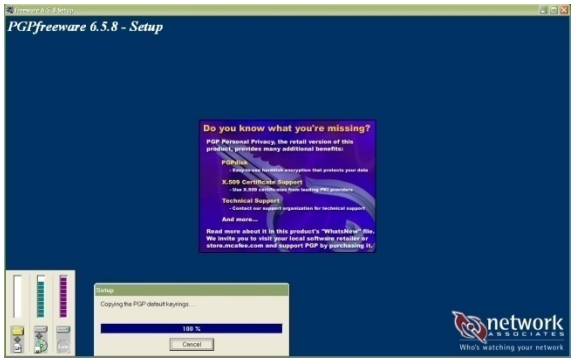
Installazione di PGPFW658Win32

Dopo aver prelevato il software (in prova per 30 giorni) clicchiamo due volte sul file Setup.exe. Fatto questo partirà la schermata iniziale per l'installazione, in cui è presente una finestra che spiega che stiamo installando un programma che implementa lo standard PGP, che è un programma di tipo freeware, che è protetto da copyright e che ogni uso non consentito sarà perseguito legalmente.



Finito di leggere le informazioni possiamo cliccare sul bottone "Next" e finiremo in una schermata che ci informa della licenza che dobbiamo accettare per installare il software e di tutte le novità presenti in questa installazione rispetto a quelle precedenti. Fatto questo la procedura ci richiederà il nostro

nome e la compagnia in cui lavoriamo, proseguendo ci richiederà in quale directory installare il programma.



Nella finestra successiva verranno richiesti quali componenti installare e quindi di scegliere le specifiche dell'installazione (c'è la possibilità di installare dei plug-in aggiuntivi per EUDORA); di seguito la procedura ci farà un resoconto delle scelte fatte (nome e cognome, tipo di installazione, directory) e partirà l'installazione vera e propria.

Finita l'installazione la procedura richiederà se possediamo già una chiave che vogliamo utilizzare e riavvierà il PC.

Utilizzo del software

All'avvio del programma vediamo che sono presenti sette pulsanti (Fig.5). Il primo pulsante a sinistra serve per creare le varie chiavi da utilizzare per criptare i messaggi e che dovremo utilizzare al primo avvio del programma.

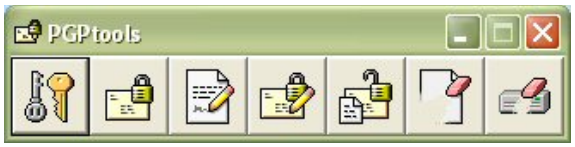


Fig.5

Dopo aver cliccato comparirà una finestra in cui viene spiegata in modo sintetico la differenza tra chiave pubblica e chiave privata. Cliccando su avanti finiremo in una schermata in cui viene richiesto il nome intero dell'utente e l'indirizzo e-mail da associare alla chiave che andremo a creare. Nei passi successivi vengono richiesti quali algoritmi utilizzare per creare la chiave, quale lunghezza questa deve avere e se deve avere un tempo di validità limitato. Continuando ci viene richiesta la passphrase, cioè la frase-chiave che servirà per proteggere la nostra chiave privata e che quindi è meglio scegliere di lunghezza sufficiente a rendere la parola o frase di difficile individuazione (la difficoltà ci viene rappresentata anche da una barra in questa schermata). Nelle schermate successive verrà visualizzato lo stato della creazione della passphrase e se pubblicarla sul server. Finiti questi passaggi ci siamo creati la passphrase. Per criptare un messaggio dobbiamo cliccargli sopra con il tasto destro del mouse, scegliere PGP, cliccare su "Encrypt" e scegliere quale chiave utilizzare.
Per decriptare un file basta cliccarci sopra due volte, inserire la passphrase e poi indicare dove collocare il file decodificato.
Il programma ci è sembrato poco adatto per utenti poco esperti ed abbiamo trovato difficoltà nel suo utilizzo.

Installazione e commenti sul software prelevato dal sito del PGP

Questa descrizione sarà molto più approfondita rispetto alla precedente, perché ci è sembrato il software più intuitivo e completo dei due che abbiamo testato.
Quando si avvia la procedura di installazione il programma chiede se si è un nuovo utente o se si hanno già a disposizione una o più chiavi. Se si è un nuovo utente, prima di creare la chiave (o le chiavi) consigliamo di impostare alcune voci dei tab "General", "Servers" ed "Advanced" come indicato man mano che si legge:
avviare *PGPkeys* cliccare sul menu *Edit>Options* , appare così il tab "General". Queste sono le voci presenti:

- Always encrypt to default key se attivato, tutti i messaggi o i file, cifrati con la chiave pubblica di un destinatario, saranno cifrati anche con la propria chiave pubblica impostata come default, in modo da poter essere in grado di decifrare il documento.
- Faster key generation è consigliabile disabilitare tale voce, così facendo si avrà una generazione delle chiavi molto più sicura.
- Comment block commento visualizzato in tutti i file cifrati.
- Cache passphrase while logged on per ogni tipo di azione, memorizza la passphrase in memoria fino al logoff dell'utente.
- Cache passphrase for memorizza la passphrase in memoria per il tempo indicato e per l'azione indicata.
- Do not cache passphrase non memorizza la passphrase in memoria.
- Share passphrase cache among modules permette di passare da un modulo all'altro senza digitare nuovamente la passphrase (se memorizzata in memoria).
- Number of passes numero di passaggi da effettuare sull'hard disk quando si esegue la cancellazione sicura di un file (3 passaggi possono bastare).

Nel tab "Files" sono indicati i percorsi del portachiavi pubblico e del portachiavi privato. La coppia di chiavi, PGP la memorizza in due file criptati: pubring.pkr che contiene le chiavi pubbliche e secring.skr che contiene le chiavi private. Se si perde la chiave segreta, non sarà possibile decifrare nessuna informazione e la chiave pubblica associata sarà quindi inutilizzabile. Prestare molta cura al file *secring.skr* poiché contiene la chiave privata personale (o più di una). C'è da dire che, al contrario del file di chiavi pubbliche, questo file viene memorizzato in modo criptato: per questo motivo quando si utilizza la chiave privata viene chiesto l'inserimento della passphrase personale. Per maggiore sicurezza è meglio non condividere tale file in rete, non lasciarlo memorizzato su un computer che non è il proprio, eventualmente crittografare ulteriormente questo file con il metodo "*Conventional Encryption*" (descritto più avanti). In poche parole: mai nessuno deve entrare in possesso del file secring.skr perché analizzandolo potrebbe scoprire la propria passphrase segreta. Un'efficace precauzione sarebbe quella di memorizzare il file *secring.skr* su di un supporto mobile (una penna USB o un semplice floppy) ed impostare il programma in modo che quando lo si utilizza andrà a ricercare la chiave privata nel file sul supporto mobile anziché sull'hard disk. Un altro trucco può rivelarsi molto efficace per gli amministratori di Server: rinominare i file di default che usa il programma (*secring.skr* e *pubring.pkr*) e quindi impostare il programma per usare i file chiamati diversamente.

Tab **"Email"**:

- Use PGP/MIME when sending email se il programma di posta elettronica che si usa supporta il formato PGP/ MIME, attivando questa opzione si farà si che il messaggio sarà criptato o firmato automaticamente. Il destinatario deve avere un programma compatibile con il formato PGP/ MIME (ora chiamato OpenPGP, http://www.openpgp.org e http://www.ietf.org/ rfc/ rfc2440.txt), ad esempio Eudora.
- Encrypt new messages by default cifra automaticamente tutti i messaggi in uscita.
- Sign new messages by default firma automaticamente tutti i messaggi in uscita.
- Automatically decrypt/verify when opening messages procede a decriptare/ verificare i messaggi in modo automatico quando vengono aperti.
- Always use Secure Viewer when decrypting permette di visualizzare i messaggi decifrati con caratteri speciali in una finestra chiamata "*Secure Viewer*". I messaggi cifrati con questa opzione non possono essere salvati in chiaro quando li si visiona.
- Word wrap clear-signed messages indica il numero di colonne dopo il quale sarà applicato un ritorno a capo all'interno del testo che contiene la firma digitale. Questo comando è stato introdotto perché alcuni programmi di posta inseriscono un ritorno a capo in modo errato, compromettendo la leggibilità.
- Wrap at column la colonna a cui applicare obbligatoriamente il ritorno a capo.

Tab **"HotKeys"**, consente l'impostazione delle scorciatoie da tastiera, per i comandi di maggior utilizzo, davvero utile in modo da non dover ricorrere sempre all'icona sulla Tray Bar:

- Purge passphrase caches cancella la cache che contiene la passphrase.
- Encrypt current window cripta il contenuto della finestra che possiede il focus.
- Sign current window firma il contenuto della finestra che possiede il focus.
- Encrypt & Sign current window cripta e firma il contenuto della finestra che possiede il focus.
- Decrypt & Verify current window decripta e verifica il contenuto della finestra che possiede il focus.

Tab **"Servers"**:
impostazioni a titolo di esempio, si possono vedere in Fig.6:

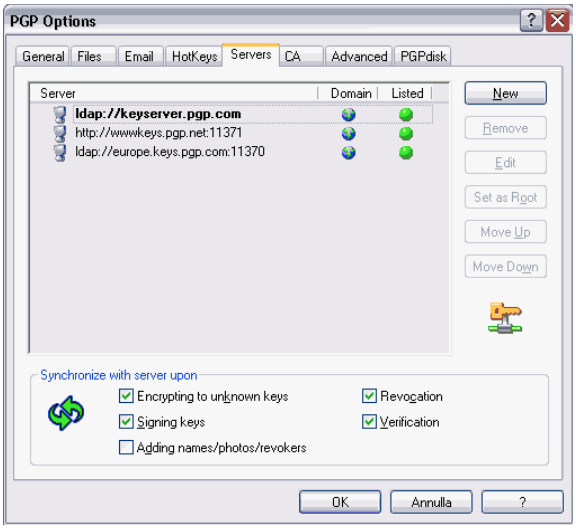


Fig.6

In genere i primi due keyserver sono presenti di default. E comunicano tra loro, scambiandosi le chiavi. Quindi basterà mandare la/e propria/e chiave/i pubbliche ad uno dei due. Se non esistono i keyserver, effettuare questa serie di operazioni:

1) Cliccare sul pulsante " *New*"
2) Nel campo "Type" selezionare dal menu a discesa la voce PGP Keyserver LDAP
3) Nel campo "Name" digitare keyserver.pgp.com
4) Premere il pulsante " *Ok*" .

Passiamo all'altro keyserver.

1) Clicare sul pulsante " *New*"
2) Nel campo "Type" selezionare la voce PGP keyserver http
3) Nel campo " Name" digitare wwwkeys.pgp.net e nel campo "Port" digitare 11371.

Vediamo le opzioni di questo tab, che si riferiscono al momento in cui bisogna effettuare una sincronizzazione fra le chiavi del proprio portachiavi locale e le chiavi sui *keyserver* :

- Encrypting to Unknown Keys se riceviamo un documento criptato, e non si possiede la chiave pubblica del mittente, il programma cerca di importare la chiave pubblica dal keyserver predefinito.
- Signing Keys prima di firmarla, controlla sul keyserver se la chiave è scaduta o revocata. Poi aggiorna la registrazione sul keyserver.
- Adding Names/Photos/Revokers prima di effettuare l'operazione di aggiunta nome, foto, o addetti alla revoca, controlla la validità delle chiavi e la invia nuovamente al keyserver.
- Revocation aggiorna la chiave che sarà revocata e la restituisce al keyserver dopo l'operazione.
- Verification alla verifica di un file o messaggio cifrato del quale non si possiede la chiave pubblica del mittente, il programma ricercherà ed importerà automaticamente la chiave dal keyserver.

Tutte le operazioni sopra indicate, andranno a buon fine se è attiva una connessione ad internet. Questo è un consiglio che diamo vivamente, difatti mentre provavamo questi menu, non ci eravamo accorti che era saltata la linea ADSL e, quindi, non riuscivamo a farli funzionare, perdendo più di un'ora su questo problema.

Tab **"CA"**: opzioni riservate all'uso dei certificati *X.509*.

- URL l'indirizzo della *Root Certificate Authority*.
- Revocation URL l'indirizzo alla quale è disponibile la *Certificate Revocation List* della CA.
- TYPE il tipo di CA utilizzato.
- Root Certificate informazioni sul certificato della root CA.
- Clear Certificate cancella quanto alla voce precedente.
- Select Certificate specifica un certificato di root CA.

Tab **"Advanced"**:

- Preferred algorithm permette di specificare l'algoritmo simmetrico per criptare. Di default è impostato il *CAST*.
- Allowed Algorithms solo gli algoritmi selezionati verranno usati per cifrare.
- Display Marginal Validity Level visualizza le chiavi in parte non valide oppure permette di visualizzarne la validità mediante un cerchio colorato: verde per una chiave valida, grigio per chiavi non valide.

- Treat Marginally Valid Keys as Untrusted se attivato, tratta tutte le chiavi parzialmente valide come non degne di fiducia e avvisa di questo.
- Warn When Encrypting Keys to keys with an ADK avvisa prima di procedere con l'utilizzo di una chiave pubblica processata utilizzando l'*ADK* (*Additional Decrytpion Key*).
- Export format – Compatible esporta le chiavi con un formato compatibile alle versioni precedenti di PGP.
- Export format – Complete esporta le chiavi nel nuovo formato, che comprende anche l'ID fotografico, i certificati X.509 ed altro.

Proprietà generiche di una chiave

Per visualizzare le proprietà di una chiave, fare clic con il tasto destro del mouse su di essa, e dal menu contestuale scegliere la voce " *Key Properties* " . Appariranno queste voci:

- ID identificativo univoco associato alla chiave. Per una chiave di tipo V3 (versione 3, utilizzato dalla versione 5.x di PGP) significa i 64 bit meno significativi del modulo pubblico della chiave RSA. Nella versione V4 (da PGP 6.x in poi) corrisponde ai 64 bit meno significativi del fingerprint.
- Type l'algoritmo a chiave pubblica utilizzato. Esso può essere:
    - *RSA* per le operazioni di cifratura della chiave e di firma verrà utilizzato l'algoritmo RSA. Con le nuove chiavi *V4* si utilizzano due coppie di chiavi, una per le operazioni di cifratura, un'altra per le operazioni firma/ verifica.
    - *DH/DSS* per cifrare la chiave verrà utilizzato l'algoritmo *DH* mentre per firmare si ricorrerà a quanto previsto dal *Digital Signature Standard* (*DSS*).
- Size la dimensione varia in base all'algoritmo utilizzato.
- Created data di creazione della chiave.
- Expires data di scadenza della chiave. Se indicato *Never* non scadrà mai.
- Cipher algoritmo simmetrico utilizzato.
- Enabled se disabilitato, impedisce l'utilizzo della chiave.
- Fingerprint se la chiave è *V3*, consiste nel risultato dell'applicazione dell'algoritmo *MD5* alla chiave pubblica, senza considerare la lunghezza della chiave stessa. Per una chiave *V4* invece, sono considerati i 160 bit risultanti dall'utilizzo dell'algoritmo *SHA-1* avente in ingresso: *packet tag*, che indica la tipologia di pacchetto (chiave pubblica, chiave privata,…) di dimensione di un ottetto; *lunghezza del pacchetto* , di dimensione pari a due ottetti; tutto il pacchetto chiave pubblica a partire dal campo versione.
- Hexadecimal permette di visualizzare il *fingerprint* in formato esadecimale. Quando è disattivato, il *fingerprint* è visualizzato con delle parole in inglese la cui pronuncia è inconfondibile con altre parole se dettate al telefono.

Poi ci sono le opzioni legate al concetto di *Web of Trust* (ragnatela di fiducia) nel riquadro Trust Model e sono:

- Implicit Trust permette di dichiarare una chiave valida implicitamente, caratteristica che hanno le chiavi create di persona.
- Untrusted-Trusted livello di validità della chiave; essa può essere " non valida", "marginalmente valida" e "valida" .

Esempio di proprietà di una chiave è raffigurata nella Fig.7:

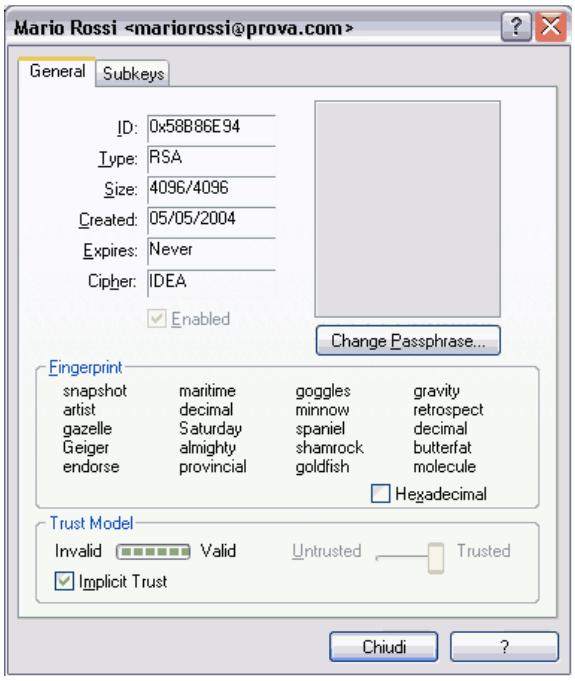


Fig.7

Per creare una nuova coppia di chiavi procedere nel seguente modo: avviare *PGPkeys* , fare clic sul menu *Keys>New key*..., apparirà la finestra raffigurata in Fig.8:

Fig.8

Se si fa clic sul pulsante "Expert" si passa alla modalità per esperti. Cliccando sul pulsante "*Avanti*" inserire il nome e l'indirizzo e-mail nei rispettivi campi. Cliccare ancora sul pulsante "*Avanti*" ed apparirà la finestra in cui inserire nel primo campo la *Passphrase* , che può includere qualsiasi combinazione di caratteri della tastiera. L'avanzamento della barra "*Passphrase Quality*" indicherà la "bontà" della combinazione immessa. Dopo aver immesso la passphrase anche nel campo *Confirmation*, fare clic su "*Avanti*" ed attendere la generazione della chiave muovendo il mouse e premendo tasti a caso sulla tastiera, infine cliccare su "*Fine*" . Per quanto riguarda la modalità Expert, le informazioni da inserire sono quasi identiche: quello che si imposta manualmente è il tipo della chiave e la dimensione. In Fig.9 si può vedere la finestra che appare se si accede alla modalità *Expert*:

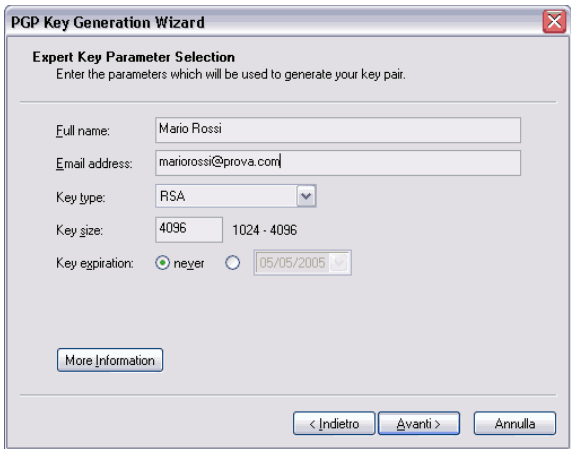


Fig.9

Il tipo della chiave (campo "*Key type* :") può essere:

- Diffie-Hellman/DSS la cui coppia di chiavi generata sarà compatibile con le versioni PGP a partire dalla 5.x in poi. In teoria, la dimensione massima è di 4096 bit. In genere però le chiavi generate saranno di 1024 o al massimo 2048 bit.
- RSA nuova versione dell'algoritmo RSA V4 con grandezza di 4096 bit. Con questo algoritmo si può impostare la dimensione massima di 4096 bit e le chiavi saranno generate effettivamente con questa dimensione. Le chiavi tuttavia saranno compatibili a partire dalla versione 7.x di PGP.
- RSA Legacy grandezza massima delle chiavi è di 2048 bit. Le chiavi saranno compatibili con tutte le versioni di PGP.

Quale scegliere? Ovviamente per la massima sicurezza è consigliabile *RSA*. Non è importante che serva una versione più aggiornata del programma per leggere questo tipo di chiavi, visto che si può scaricare gratuitamente dal sito. Ricapitolando, terminata l'installazione e riavviato il computer, l'icona del programma si posiziona accanto a quella dell'orologio (area SysTray), e si hanno ha disposizione due chiavi: una privata (da conservare molto scrupolosamente come descritto) e una pubblica. Se dopo il riavvio del computer non dovesse comparire l'icona del programma accanto all'orologio, fare clic su *Start>Programmi>PGP>PGPtray*.

DICHIARARE VALIDE LE CHIAVI PUBBLICHE

Quando desideriamo comunicare in sicurezza con un destinatario, dobbiamo dichiarare valida la chiave pubblica del nostro corrispondente per poterla utilizzare. Se possediamo il file .asc (o .txt) del nostro destinatario, basta fare clic sul menu *Keys>Import…* e poi selezionare il file con la chiave dal percorso in cui risiede. Per recuperare una chiave da un keyserver, clic sul menu *Server>Search...* e inserire un qualsiasi dato che identifica la persona che si vuole cercare (può essere il nome, l'indirizzo e-mail o altre informazioni).
Scaricata la chiave, come si può notare, se non si dichiara valida la chiave importata o scaricata, essa non può essere utilizzata correttamente, come mostra il cerchio di colore grigio nel campo '*Validity*'. Per validare la chiave, fare clic su di essa con il pulsante destro del mouse e scegliere la voce " *Sign…*" . Dovrebbe apparire una finestra simile a quella in Fig.10:

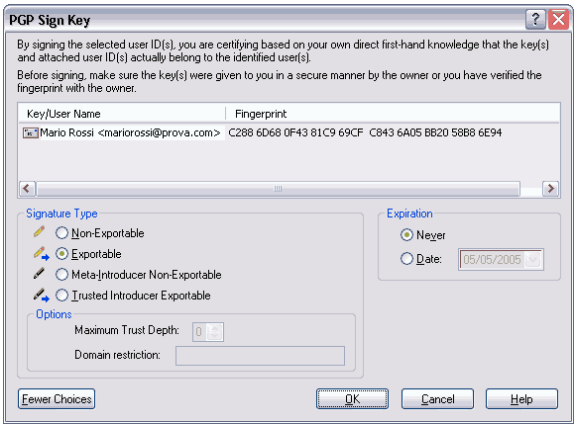


Fig.10

Se non appare il riquadro “Signature Type” , fare clic sul pulsante “*More Choices*” . Ecco cosa significano le varie voci:

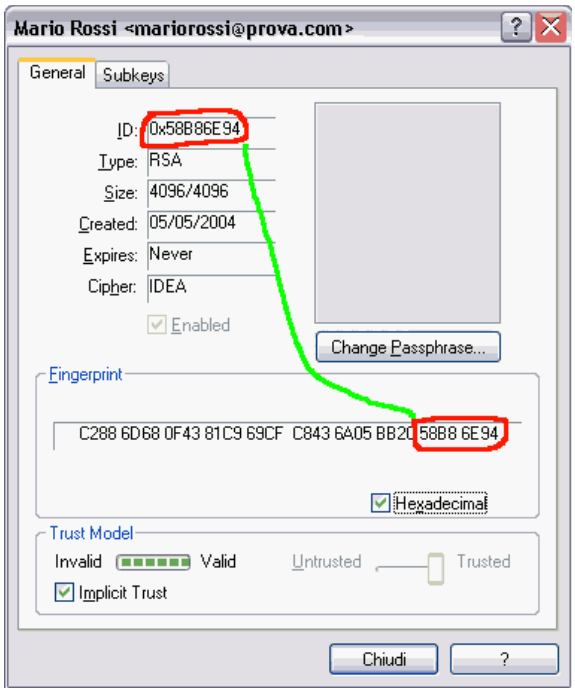
- Non-Exportable la chiave è per noi valida, ma non desideriamo che altri si basino sulla nostra validazione. La nostra fiducia dunque non sarà inviata al keyserver, ma rimarrà in locale.
- Exportable come per la precedenze opzione, in questo caso però si acconsente la decisione della fiducia ad altri.
- Meta-Introducer Non-Exportable firmando la chiave, si darà fiducia al proprietario della stessa ed a quelle dichiarate valide da quest’ultimo, nonché ai *trustud introduced* creati dalla stessa chiave. In poche parole le chiavi dichiarate valide da un meta-introducer (paragonabile ad una CA root) saranno valide anche per noi.
- Trusted Introducer Exportable oltre a dar fiducia al proprietario della chiave, è considerato garante per le chiavi che ha firmato (un delegato, simile ad un delegato di una CA root). E che quindi sono valide anche per noi. E’ possibile indicare la profondità con cui è nidificata una chiave.

Il riquadro “Expiration” indica la data di validità:

- Never significa che non scadrà mai.
- Date la firma scadrà nella data specificata.

In genere è sufficiente utilizzare una delle prime due voci (*Non-Exportable* oppure *Exportable*). C’è un caso particolare: se una chiave importata ne contiene delle altre al suo interno, è possibile dichiarare valide o meno quelle che contiene agendo nel riquadro ‘*Trust Model* ’ che appare quando si aprono le proprietà di una chiave. Questa voce è da collegare a quelle sopra elencate (*Meta-Introducer Non-Exportable* , ecc). Anche se non si modifica il concetto di ‘*Trust Model* ’ questa chiave è comunque utilizzabile, al contrario di quelle che definisce al suo interno. Prima di procedere a firmare la chiave assicurarsi che la stessa appartenga veramente alla persona interessata mediante la verifica del *fingerprint* (detto anche *impronta digitale*).

Nota: per controllare il *fingerprint*, cliccare con il pulsante destro del mouse sulla chiave, scegliere la voce “ *Key Properties* ” e nel riquadro “ *Fingerprint*” attivare la voce “*Hexadecimal*”.



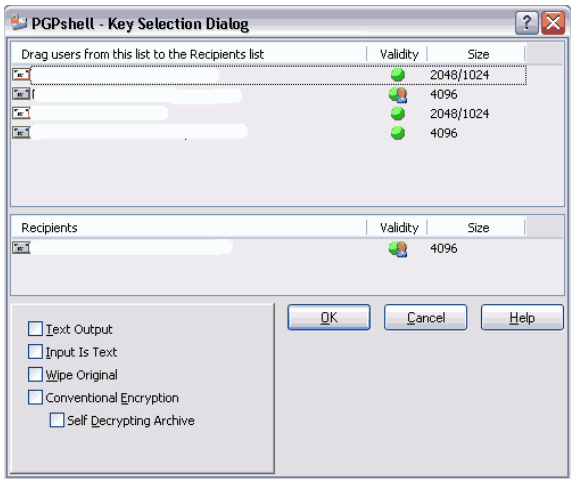
Analizzare la stringa dell'ID con la corrisponde stringa dei bit meno significativi del *fingerprint*: se solo un carattere è diverso fra quelli cerchiati, vuol dire che la chiave pubblica è stata modificata o che si tratta di un'altra persona e non di quella desiderata. In questo caso corrispondono.
Espandendo la chiave, fra le altre firme noteremo anche la nostra. Ricordarsi che una chiave pubblica può contenere anche altre firme, non solo quella del proprietario.

REVOCARE UNA CHIAVE

Se si ha la necessità di revocare una propria chiave, l'operazione è molto semplice: basta cliccare con il tasto destro del mouse sulla chiave e scegliere la voce Revoke. Se poi si aggiorna la chiave revocata anche sul *keyserver*, quando qualcuno la scarica non sarà funzionante.
Ricordarsi che quando si revoca una chiave con cui era stata firmata una chiave pubblica di un certo destinatario, quest'ultima non sarà più utilizzabile se non la si firma nuovamente con la nuova chiave.
Quando si deve cifrare un documento, dobbiamo porci una sorta di domanda: chi deve essere in grado di decifrare il messaggio che stiamo per inviare? La risposta sarà nelle chiavi pubbliche che utilizzeremo.

INVIO DI FILE CIFRATI

Cercare il file desiderato all'interno di 'Esplora Risorse', clic con il pulsante destro del mouse e scegliere la voce *PGP>Encrypt* (se si vuole anche firmarlo digitalmente scegliere *Encrypt & Sign*), poi dovrebbe apparire la seguente finestra:



Nella parte alta della finestra, come si vede, sono presenti le chiavi pubbliche dei destinatari, cioè chi sarà in grado di decifrare il nostro messaggio. Tramite un semplice drag-and-drop, trascinare il/ i destinatario/ i che deve essere in grado decifrare il messaggio cifrato, nel riquadro " Recipients" sotto alla nostra chiave pubblica. Infatti chi esegue la cifratura è presente di default, se attiva la voce Always encrypt to default key vista prima.
Premendo sul tasto OK, il file sarà crittografato, sarà aggiunta l'estensione *.pgp* (apparirà l'icona a forma di lucchetto), ed è pronto per essere allegato, oppure decifrato da noi stessi.
Le opzioni disponibili significano:

- Text Output crea il file cifrato solo caratteri ASCII Armored, non in binario.
- Input Is Text da selezionare quando il documento in questione contiene solo caratteri alfanumerici.
- Wipe Original da usare con cautela, in quanto se attivo cancella in modo irrecuperabile il file originale.
- Secure Viewer aggiunge un elemento di sicurezza contro sofisticate tecniche di spionaggio industriale. In altre parole visualizza i file o testo decifrati, utilizzando caratteri speciali all'interno della *Secure Viewer*. E non è possibile salvare in chiaro questi messaggi ne copiare il testo in memoria, appunto perché sono visibili esclusivamente nella finestra *Secure Viewer* . Tuttavia questa opzione non è compatibile con alcune precedenti versioni di PGP.
- Conventional Encryption permette di cifrare il messaggio senza usare la chiave pubblica, e possiamo scegliere una frase convenuta tra noi ed il destinatario. Così facendo quando il destinatario riceverà la nostra e-mail (o file) crittografata, dal menu di *PGPtray*, gli basterà scegliere *Current Windows >Decrypt & Verify*, e quindi digitare la frase convenuta.
- Self Decrypting Archive nel caso in cui il destinatario non possiede PGP.

Se, invece di un intero file, vogliamo cifrare un testo, procedere nel seguente modo:

- ➔ Per esempio all'interno di un programma di videoscrittura (o client e-mail), clic su *PGPtray>Current Window>Encrypt*. Dalla finestra che appare trascinare i destinatari che devono essere in grado di decifrare il messaggio.
- ➔ In alternativa, selezionare tutto, premere CTRL+C (per copiare tutto il contenuto), clic su *PGPtray>Clipboard>Encrypt* . Trascinare i destinatari dalla solita finestra. Incollare poi il testo (premendo CTRL+V) per esempio nel programma di posta elettronica che si utilizza.

Il destinatario che riceve il messaggio cifrato, dovrà semplicemente selezionare dal *PGPtray* la voce *Current Window>Decrypt & Verify* (o scorciatoria da tastiera se l'ha impostata) ed inserire la sua parola chiave. Oppure copiare il testo e scegliere la voce *Decrypt & Verify* questa volta dal menu *Clipboard* .
La prima procedura è meno elaboriosa della seconda, e noterete comunque che il primo metodo effettua le stesse operazioni della seconda (cioè la copia in memoria del contenuto della finestra corrente), risparmiando qualche operazione all'utilizzatore del software.
Ricapitolando dunque avviene questo: il mittente cripta il messaggio che vuole inviare con la chiave pubblica del destinatario, a sua volta il destinatario decripta il messaggio cifrato con la sua chiave privata.

INVIO DI E-MAIL CRIPTATE

Scrivere il messaggio nella finestra del proprio client e-mail come al solito, inserire i destinatari nei campi " *A:*" , in " *Cc:*" , " *Bcc:*" , e poi cliccare su *PGPtray>Current Window>Encrypt* (oppure *Encrypt & Sign*). Ci appare una finestra, come visto in precedenza, che presenta solo le opzioni Secure Viewer e Conventional Encryption descritti sopra.
Esempio: avviato il client e-mail preferito, inserire i dati come sempre. Supponiamo di scrivere questo testo:
Ciao,
tutto bene? Ci vediamo stasera?
cliccare poi su *PGPtray>Current Window>Encrypt* e come risultato la finestra del proprio client mostra qualcosa simile a questo:

-----BEGIN PGP MESSAGE-----

qANQR1DBwUwDLzYFSlW5Z44BEAC0GsrWUZlmh9n/ b/ cXXTlewVQPZYHt36j8+UC/
DJwpTjiX0nhdJMEiE5wHiAXTxDDoexK0y9WRrbsxScTSmZneVgjEn0cK42S1oVzA
TmVmDZxlUoa9F9zyaomThe+4XxLT5m5pf/ 2zLbCsR07QDwBn2uE+Jpe4bwXzWl/ 8
5yRS/ fHj2gk2w/ N9eXCzvewUg8QrV23C6rBXnDqsa11iGLtQXOKVfrYGOt/ FOdrP
YxfBfw6rWaVqwgM03/ XmFk8vGUKfy9cFN4lBkzwuakNt955Eyg99Z6GfDx3XbGer
tYtWc0pyCWkgCKBufDZx+lbvjstuzddm3co4NctQRXfa4vBTADbBGegtyMHC9CKV
03slcM6tpWdAj7iN+w8Mgf8R/ OxTAYWHptCQTW5KBZXqGMF1BdnCPpNtM9gufJsd
QuTHRQVjR7xX37eZylYgpYjGbEUvdu1rP8G/ VaNewIk5dY5Ujef0m/ TzaYvpVM/ v
MTJN4yekxpad+HXZGz5ULZBwWs+FRFEoT5mdK8ApWh/ ETmfgnzZUd8XOlzlCuMth
QJu/ vvgvF8W7ANhG8vSmVasIC1DfbZsge2uBSb5mwYfuSb3DsiuqUR8VVPoKxWTT
krimxF56QavN01Ep9Znr18itRSg+S1ijzqsH5Gx6ItG1bnCaSHfb5DGKwxNSFqsu
ufzpP8HBTAPAdgFdxfp8mQEP/ 3ITHzH9KiQjp0VbGzW6jeAb3/ gcqTpNhMOMCn/ +
p5zDLN6Y5/ wd/ 9pyKaBn69aO97dWuAiS1EYDHFUchUhc8NbZ4ASbocILv+6fyrHn
M4ikpK1+CO31kFjWuemawbqA3pLiS2jpdanJwmTFedN3BUB1ZvX4D283ZCvxSjL1
BK8gyv9DvLV30n8r4K9Gov1X+QG0L1Dh5ccsKO6bn0GD3KPxO2J1aTFRA/ / Q1CU4
y6CUDuPWLtEoMWGKigDE0dIL21b4xWU2yt/ +tF5BVj+424g4YbTv6YY/ i1yOWmqg
w+TIDf8V2TQpUbbGA0MdrQcaCULZRYkWQmW7W582qrcs7jQTHhfDZlgR8BqhrcxF
n3eMnXtPu0HNi3yupyyY0IK2cM5hBqZbb8d7qAXqBFYC/ jADuY22li+v+2eqBjS0
cVxBAuyVDqvQ0wsYqEox2dvUzPh4U3uSRHF4cbJ1/ SBvvbDJ+H7WvgHW5IH6Prye
BSQQDJkRZd66aEzFd/ AprLdhIXLBKO/ Lhe/ v3eaSbuA6/ tIVm5eXUX5iNmHdr+3e
WevKDEjVnzjcWtTpL4LB7Hs24omV0ufa8A2ezaMxCrODtFMUPxmbz3f+mycUsSn1

R67ENlm9KjGWl18FNj9Imy0GtXABnM9KnFAygdnWpXiyjQ9NDX2UAm99fhyaAKwB
1w1fyT2W4e1KhzYnlkbjsLCkipa9SwpxSovJ/ noWA0KuFx1A9EI3hze95JDn9e59
h4y3I6/ n6dF6i/ mokpwwzTju
=DniC
-----END PGP MESSAGE-----

Se qualcuno vuole provare a risalire al messaggio originale " *Ciao, tutto bene? Ci vediamo stasera?* " partendo da quest'ultimo criptato, è libero di farlo ;-)

VERIFICARE UN DOCUMENTO

Per verificare un documento firmato digitalmente all'interno di un programma, cliccare su *PGPtray>Current Window>Decrypt&Verify* (oppure usare la scorciatoia da tastiera). Il programma utilizzerà la corrispondente chiave pubblica del mittente associata alla chiave per verificare il documento.

- Se la verifica avviene in modo corretto apparirà il messaggio 'PGP Signature Status: Good'.
- Se qualcuno ha modificato il documento, anche di un solo bit, apparirà il messaggio 'PGP Signature Status: Bad'.
- Per verificare un file sull'hard disk, cliccare su di esso con il tasto destro del mouse e scegliere *PGP>Verify signature.*

INVIO DOCUMENTI A DESTINATARI SENZA IL PROGRAMMA PGP

Supponiamo che un nostro destinatario non possiede il programma PGP, perché nelle procedure descritte fino adesso, si presuppone che il nostro destinatario abbia il PGP installato. Come fare per mandargli una e-mail o un file in maniera sicura? Selezioniamo con il tasto destro del mouse un qualsiasi documento sull'hard disk, e dal menu contestuale che ci appare selezionare *PGP>Create SDA* (*Self Decrypting Archive*), oppure *PGP>Encrypt*, poi spuntare la casella *Self Decrypting Archive* (la casella *Conventional Encryption* si auto-seleziona). Vi verrà chiesto un nome per il nuovo file cifrato (non modificare l'estensione che sarà assegnata automaticamente). Il documento potrà essere allegato a qualsiasi messaggio di posta elettronica, ed il destinatario lo potrà leggere inserendo la frase convenuta (che abbiamo scelto tra di noi prima) al momento dell'apertura. La voce *Conventional Encryption* è disponibile anche per l'invio di e-mail, così facendo non si userà alcuna chiave pubblica, però in questo caso è necessario che il destinatario abbia installato il PGP, mentre nel caso visto sopra, come già detto, non c'è bisogno.

Riepilogo passaggi quando si applica la crittografia:

- il file del mittente viene compresso con un algoritmo di tipo *zip*
- il file compresso viene criptato mediante l'algoritmo *IDEA*
- viene generata una sequenza casuale di 128 bit (o più) chiamata *session key* con appositi algoritmi per ottenere una sequenza di numeri pseudo casuali che hanno determinate proprietà statistiche, devono essere equiprobabili (come se si tirasse un dado: escono numeri da 1 a 6)

- la *session key* viene cifrata con l'algoritmo *RSA* (oppure con l'algoritmo *Diffie-Hellman/DSS*) utilizzando la chiave pubblica del destinatario ed il risultato è concatenato al documento
- la *session key* criptata viene concatenata al file criptato
- infine, viene applicato l'algoritmo di trasformazione reversibile a testo ASCII chiamato Armor Radix-64 (a volte indicato Base64 o Armored). Questo algoritmo produce un documento formato da solo caratteri ASCII compatibile con tutti i server e client di posta elettronica.

Brevemente, questo formato converte un input a gruppi di 24 bit come una stringa di 4 caratteri codificati ASCII. I 24 bit vengono formati a gruppi di 8 bit (quindi 3 byte) concatenati fra loro, e poi ogni gruppo di 6 bit è codificato con un carattere stampabile compreso tra un indice da 0 a 63. Alla fine si hanno dunque 4 caratteri stampabili. Ogni singola linea, nel file di output, non avrà più di 76 caratteri. Questi 64 caratteri fanno parte del cosiddetto *ASCII 'basso'*, cioè i caratteri rappresentati dai numeri decimali nell'intervallo 0-127 (o esadecimali da 00 a 7F oppure ancora 27-1 bit).

Riepilogo passaggi quando si applica la firma digitale:

- il file del mittente viene compresso con un algoritmo di tipo *zip*
- viene calcolato l'*hashing* del file mediante l'algoritmo *SHA-1* (o *MD5*). Il codice *hash* generato viene criptato tramite *RSA* con la chiave privata del mittente ed accodato al file di origine
- eventualmente il file risultante dalle due operazioni precedenti viene trasformato in codice *ASCII Armored*
- il destinatario effettua le operazioni al contrario è verifica se i codici di *hashing* sono identici.

PGPdisk: disco virtuale o cartella protetta

Installando l'utility PGPdisk, come già detto, è possibile creare uno spazio virtuale protetto da passphrase sull'hard disk, al cui interno è possibile memorizzare ogni tipo di file e persino installarci programmi. L'utilizzo di quest'aggiunta è a pagamento, al contrario del resto del programma; per questo motivo non abbiamo potuto sperimentare il software da questo punto di vista.

PGP Shredder

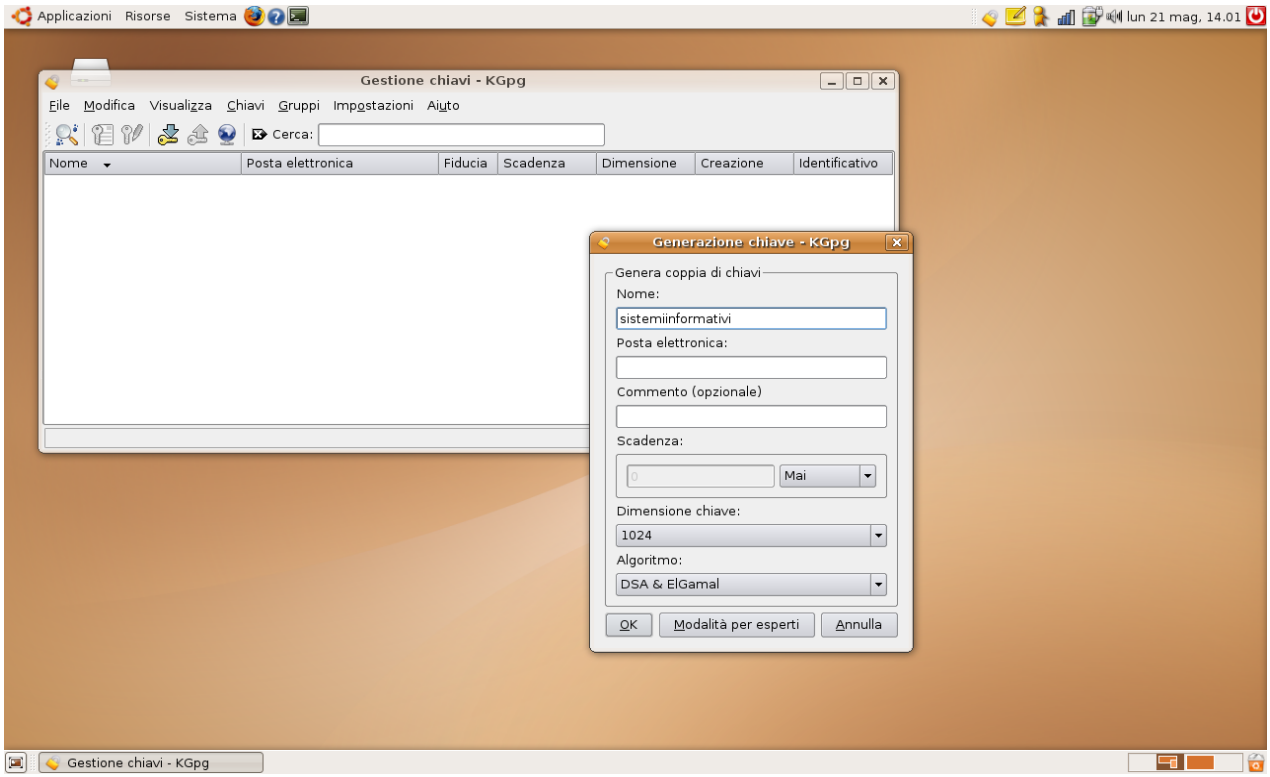
Installando PGP Desktop andremo anche a creare un cestino totalmente sicuro. Difatti, come detto all'inizio, i software come PGP non riescono a proteggere i dati se questi possono essere intercettati in metodi differenti dalla sola rete Internet, come per esempio dati gettati nella spazzatura per sbaglio ed entrati nel dimenticatoio. Questo cestino si chiama PGP Shredder; per gettare qualcosa in questo cestino dobbiamo selezionare il file da eliminare e trascinarlo al suo interno. Dopodiché il software apre una schermata in cui chiede se si è sicuri di voler eliminare i file selezionati. Questo cestino però viene anche utilizzato nel caso noi decidiamo di eliminare la copia in chiaro dei nostri dati dopo averli criptati. Cercando anche tra la documentazione non abbiamo trovato che metodi di eliminazione sicura vengono utilizzati. Solitamente vengono utilizzati i metodi DOD 5220.22 o il metodo di Gutmann, ma non siamo stati in grado di trovare niente.
In alternativa a PGP si può usare GPG (GNU Privacy Guard, http://www.gnupg.org) di cui è presente anche una versione compilata per sistema Windows e molte altre piattaforme. GPG può

leggere anche le chiavi PGP e viceversa (comunque ci sono delle limitazioni in entrambi i casi); entrambi sono conformi allo standard OpenPGP.

KGPG SU UBUNTU FEISTY 7.04

INSTALLAZIONE

Per installare KGpg su Ubuntu i passi da fare sono molto semplici.
Innanzitutto occorre andare su Sistema > Amministrazione > Gestore driver Synaptic e marcare KGpg per poi installarlo sul proprio sistema operativo.
Il primo passo consiste nell'indicare a KGpg dove si trovano il file di configurazione di gpg, solitamente al percorso /home/nome_utente/.gnupg/option.
Nel secondo passo, KGpg dà la possibilità di installare sul desktop un distruttore di file: in questo modo, trascinando un file dal File Manager sul distruttore, il file verrà sovrascritto 35 volte. Si consiglia l'attivazione di questa opzione.
Al passo tre, si procede alla creazione delle chiavi: viene chiesto obbligatoriamente il nome (anche di fantasia), l'indirizzo di posta elettronica, l'eventuale scadenza, dimensione di chiave e l'algoritmo di generazione.
Successivamente viene chiesta la password da utilizzare per cifrare/decifrare/firmare documenti, ecc.. come si può vedere in figura.



Successivamente,dopo aver inserito un nome ed una mail ed una password, verrà generata una coppia di chiavi da utilizzare per la crittografia.

UTILIZZO

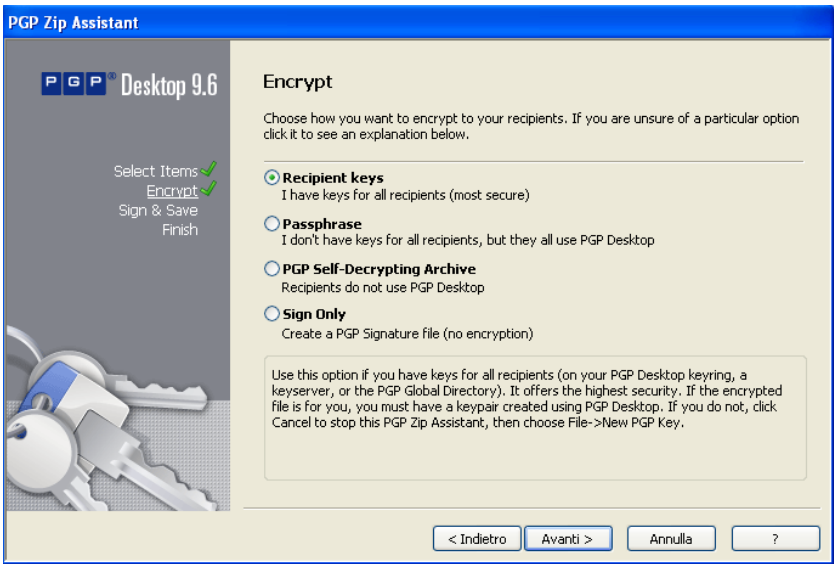
KGpg gestisce in modo trasparente il mazzo di chiavi a disposizione. Dalla finestra principale è possibile vedere l'elenco delle chiavi in nostro possesso. Con KGpg è possibile importare o esportare chiavi sia tramite file locali che mediante server dedicati: i keyserver. Analizziamo come utilizzare i keyserver. Dal menù principale di KGpg cliccare su File > Finestra server delle chiavi. Per importare una chiave basta indicare il suo identificativo composto da un numero esadecimale di dieci cifre: 0x0123456F6A. Pigiando "Cerca" viene cercato l'identificativo sul keyserver sopra indicato. Trovata la chiave pubblica cercata basta selezionare "Importa". Per esportare una chiave, selezionare "Chiave da esportare" indicare la chiave che si vuol trasmettere al keyserver. Con questo metodo possiamo trasferire la nostra chiave pubblica a tutti i keyserver che vogliamo.
Oltre ai keyserver possiamo importare o esportare chiavi da file solitamente contrassegnati dall'estensione .asc. In questo caso viene utilizzata la voce "Chiavi" dal menù principale seguito dalle voci "Esporta chiavi pubbliche" e "Importa chiave". Si segnala che la chiave può essere importata sia come file, sia dagli appunti mediante il copia & incolla.
Altra funzione importante è l'autenticità di una chiave. Cliccando con il pulsante destro del mouse su una chiave, oltre a poterla gestire, è possibile selezionare "Firma chiavi". In questo modo possiamo stabilire il livello di affidabilità di una chiave. Questa scelta, da utilizzare con oculatezza, ha riflessi anche sul gestore delle email: infatti le email firmate le cui chiavi pubbliche non sono affidabili, appaiono bordate di colore rosso, mentre quelle firmate con chiavi affidabili vengono bordate di colore verde.

APPLICAZIONI PRATICHE

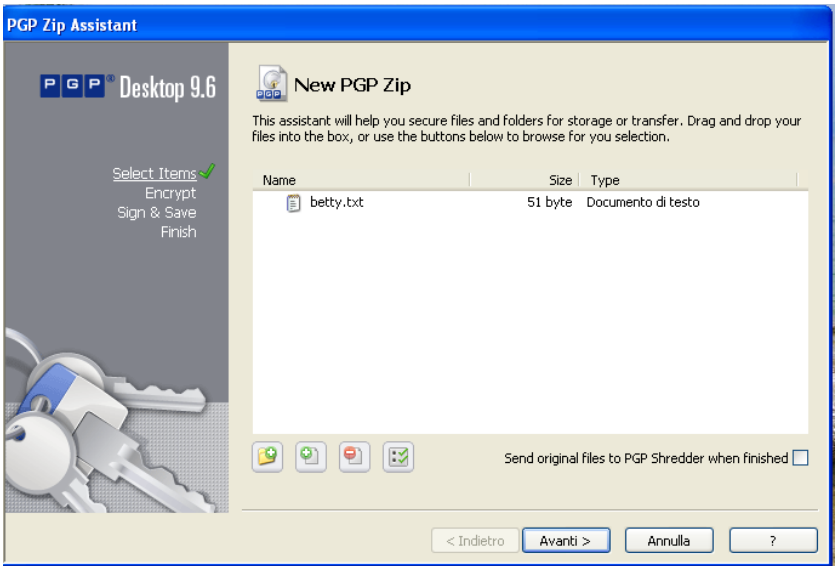
E' possibile utilizzare programmi per la crittografia dei file tra sistemi operativi uguali o diversi. Noi abbiamo provato a crittografare dei file tramite sistemi operativi uguali (Windows con Windows) e anche tra sistemi operativi diversi (Windows con Ubuntu 7.04)

- Applicazioni sullo stesso sistema operativo

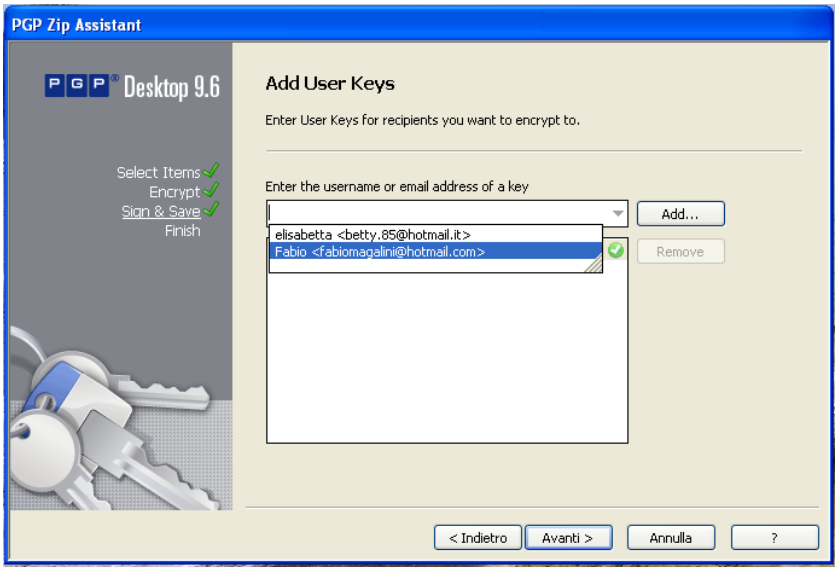
Per decriptare un file bisogna che questo sia stato criptato con la propria chiave pubblica. Occorre quindi, prima di iniziare qualsiasi operazione, scambiarsi le chiavi pubbliche.
Illustriamo ora uno scambio di dati da Betty verso Fabio.



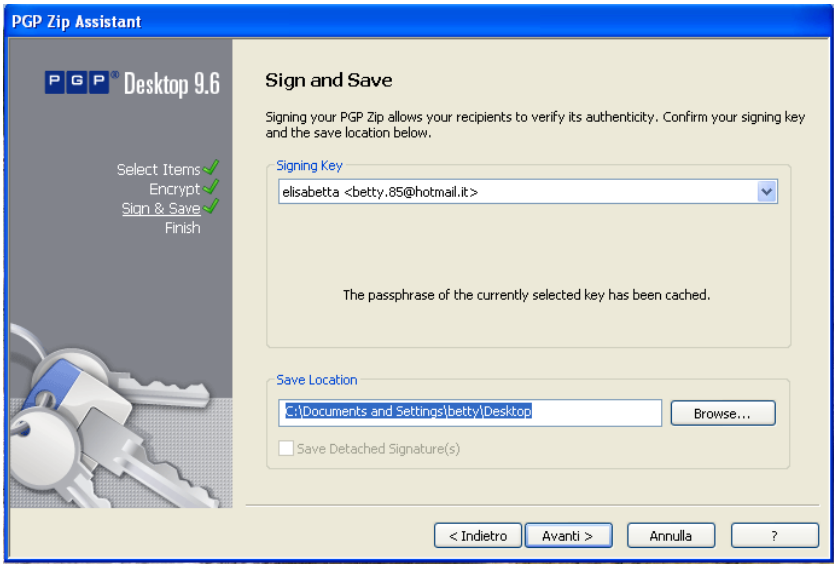
Dopo aver creato il file .txt da passare, lo si cripta con PGP, cliccandogli sopra con il tasto destro e scegliendo "PGP Desktop" → "Add "NomeFile.ext" to new PGP zip". Fatto questo si aprirà il wizard che ci segue nella creazione del file criptato. In questa finestra possiamo decidere di cancellare definitivamente (grazie a PGP Shredder) il file in chiaro.



Cliccando sul pulsante "Avanti" ci verranno presentate delle scelte su come creare il file protetto; possiamo scegliere l'opzione in cui dichiariamo di avere la chiave pubblica dell'utente a cui mandiamo il file; che non abbiamo le chiavi pubbliche di tutti gli utenti che dovranno ricevere i file codificati, ma che anche loro usano PGP Desktop; che vogliamo creare un file PGP auto-decriptante; o firmare solamente il file.

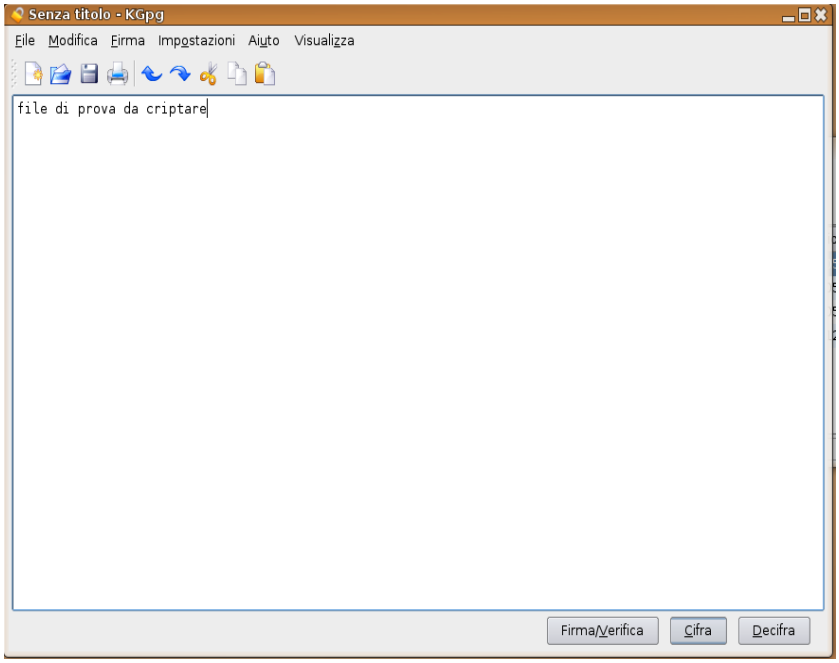


Dopo aver cliccato sul pulsante "Avanti" ci verrà presentata una finestra in cui dobbiamo scegliere chi potrà leggere il file che mandiamo, cioè andiamo a scegliere tra le chiavi pubbliche, quelle delle persone a cui vogliamo far leggere il file criptato. Da notare ch e a fianco del nome del "contatto" vengono visualizzati la sua e-mail, se siamo in possesso solamente della sua chiave pubblica o anche di quella privata e che livello di fiducia abbiamo dato alla chiave in questione.

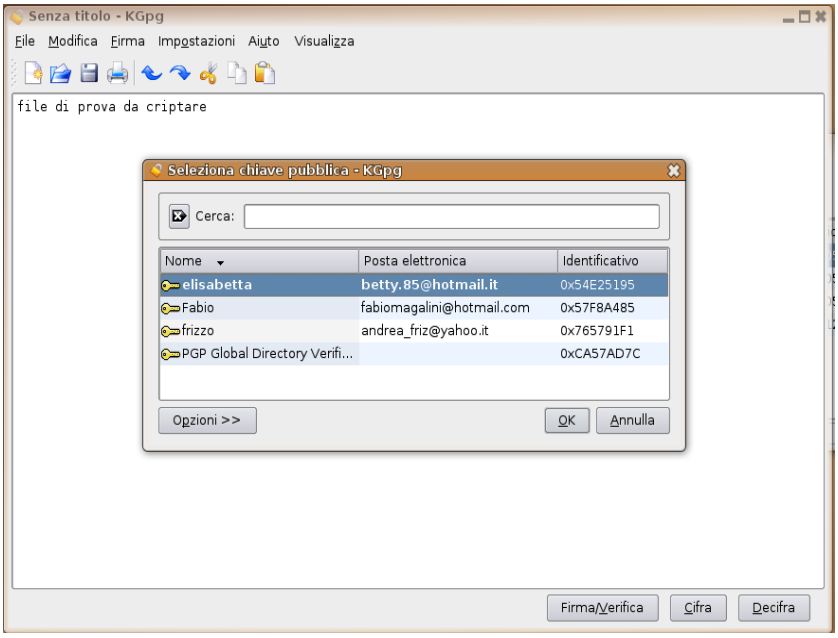


Dopodiché dobbiamo decidere in quale posizione dell'Hard Disk vogliamo mettere il file criptato con PGP ed eventualmente, se richiesto, inserire la parola chiave associata alla chiave privata che abbiamo selezionato.

-Applicazione con sistemi operativi diversi



Per criptare il file con Ubuntu usando Kgpg occorre aprire l'editor di testo e digitare il nostro testo. In alternativa si può aprire un file già esistente (File > Apri)



Per criptare cliccare con il mouse su "Cifra" e scegliere la chiave pubblica del destinatario.

Provando a criptare un file con Ubuntu e decriptarlo con Windows, abbiamo riscontrato alcuni problemi: inizialmente Windows non era in grado di aprire il file, non riconoscendone il formato. Per ovviare a questo problema abbiamo dovuto rinominare il file aggiungendo l'estensione ".txt.PGP ". Effettuata questa operazione ed aprendo il file con PGP siamo riusciti a leggere il contenuto del file senza particolari inconvenienti.

Conclusioni

Studiando il mondo della criptazione abbiamo notato che, prima dell'arrivo di PGP, questo era un mondo ancora agli albori, anche se già nell'antichità si è cercato di trovare il modo di criptare messaggi. Difatti prima dell'arrivo di PGP nessun algoritmo di criptazione era arrivato ad un livello simile di sicurezza, sbaragliando tutti i suoi concorrenti.
Bisogna però ricordare che non basta avere un algoritmo intaccabile brutalmente per avere l'idea di essere sicuri della propria privacy, bisogna avere intorno gente fidata che non venderebbe mai a nessuno la sacra chiave privata. Oltre a questo bisogna ricordarsi di svuotare sempre il cestino quando si cancellano file in chiaro e che non si vuole far sapere in giro dell'esistenza o darli direttamente in pasto a PGP shredder.
Ognuno di noi dovrebbe mettere del proprio per rendere il mondo di internet un mondo in cui si possa avere la propria privacy, anche se non si è spacciatori o terroristi; basterebbe che ognuno di noi inizi a mandare messaggi codificati per realizzare questo sogno.

Link
http://pgp.unito.it/int.html
http://wikipedia.it/

http://www.dia.unisa.it/professori/ads/corso-security/www/CORSO-9900/PGP/come.htm
http://www.tmcrew.org/privacy/PGP.htm
http://sicurezza.html.it/guide/leggi/85/guida-crittografia-e-pgp/
Immagini tratte da
http://www.trustitalia.it/decode.php?id=cfDulE002763
Software prelevati da
http://www.pgp.com
http://www.amagri.it/PGP_6.5.8_Freeware/installazione.htm