GnuPG Mini Howto (Vietnam)

*(**Translation into Vietnam By: Nguyen Xuan Binh - binhnx2000@yahoo.com**)*

(Member Of Viet Team Forum)

Tài liệu này được dịch nguyên bản và tham khảo từ:

1. Khái niệm (Concepts)

2. Cài đặt (Installtion)

3. Sử dụng khoá (Using Key)

4. Mã hoá và giải mã hoá (Encrypt And Decrypt)

5. Quá trình ký nhận và kiểm tra chữ ký (Sign And Checking Signatures)

6. Front End

- 6.1 Giao diện đồ hoạ (Graphic Interface)
- Sử dụng các chương trình E-mail (E-mail Program)

7. Nơi cung cấp thông tin về GnuPG (Source Of Information)

- 7.1 PGP
- 7.2 Keyserver
- 7.3 Book
- 7.4 Other Document

8 About Document

- Version 0.1.0 (English) April 30th 1999, Dutch Queen's Day.
- Version 0.1.1 (German)
- Version 0.1.2 (English) April 3, 2002
- Version 0.1.3 (Dutch) May 17, 2002
- Version 0.1.3 (Vietnam) December 3, 2002

Click here to view info about me:

- English
- German

1. Khái niệm

1.1 Mã hoá bằng khoá công cộng (Public Key Encryption)

Phương pháp mã hoá cổ điển chỉ sử dụng một Key cho sự mã hoá. Người gửi sẽ mã hoá thông điệp của họ bằng Key này. Về phía người nhận để giải mã hoá thông điệp thì họ cũng cần phải có Key đó. Như vậy chắc chắn người gửi sẽ phải gửi cho người nhận Key đó. Trên lý thuyết bản thân key đó rất có thể sẽ bị một người khác chiếm được trong quá trình vận chuyển. Nếu như một người khác có được Key đó thì họ sẽ có thể giải mã hoá và xem được nội dung thông điệp của người gửi. Như vậy tính an toàn và bảo mật cho thông tin không còn nữa.
Khoá công cộng (Public Keys) được ra đời để giải quyết vấn đề này. Thực chất khoá Public Keys là việc sử dụng 2 Keys có liên quan với nhau:

- **Public Key**: Được sử dụng để mã hoá những thông tin mà bạn muốn chia sẻ với bất cứ ai. Chính vì vậy bạn có thể tự do phân phát nó cho bất cứ ai mà bạn cần chia sẻ thông tin ở dạng mã hoá.

- **Private Key**: Đúng như cái tên, Key này thuộc sở hữu riêng tư của bạn và nó được sử dụng để mã hoá thông tin cho mục đích cá nhân của riêng bạn. Chỉ mình bạn sở hữu nó, Key này không được phép và không lên phân phát cho bất cứ ai.
Khi cần thực hiện các nhu cầu mã hoá các thông tin riêng tư của chính mình một cách bí mật mà không muốn chia sẻ thông tin ấy cho bất cứ ai. Bạn cần phải sử dụng Private Key. Ngược lại khi muốn mã hoá các thông tin với mục đích chia sẻ nó với một số đối tượng mà bạn muốn chia sẻ thông tin bạn sẽ sử dụng Public Key để mã hoá chúng rồi gửi cho họ. Sau đó bản thân họ sẽ sử dụng Private Key của chính mình để giải mã hoá.
Điều quan trọng ở khái niệm này là bạn phân biệt và hiểu được tính năng và cách sử dụng của 2 loại Keys: Public Key và Private Key.

1.2 Chữ ký điện tử (Digital Signatures)

Để chứng minh rằng một thông điệp đã thực sự được gửi bởi chính người gửi mà không phải là do một kẻ khác giả mạo. Khái niệm Digital Signatures được ra đời. Khi sử dụng Digital Signatures bạn sẽ kiểm tra được tính xác thực của một thông điệp. Việc sử dụng Digital Signatures sẽ giảm bớt nguy cơ giả mạo thông điệp (đặc biệt là các thông điệp giả mạo các hãng Security, Software lớn với mục đích phát tán Virus hay Trojan tới bạn). Bởi bạn có thể dễ dàng xác minh được thông điệp đó có phải thực sự đến từ đó hay không ?

Digital Signatures là sự kết hợp giữa Secret Key (khoá bí mật) và text. Tiếp đó nó sẽ sử dụng Public Key của người gửi để thẩm tra thông điệp. Nó không chỉ kiểm tra, thẩm định thông tin về người gửi mà nó còn có thể kiểm tra cả nội dung của thông điệp. Như vậy bạn sẽ biết được rằng thông điệp đó không bị giả mạo và nó không bị sửa đổi hay can thiệp vào nội dung trong quá trình vận chuyển.

1.3 Mạng an toàn (Web Of Trust)

Một điểm yếu trong thuật toán của Public Key. Cho phép một người sử dụng có thể mang và lưu hành một Public Key với User ID không chính xác. Kẻ tấn công có thể lợi dụng yếu điểm này để giải mã và đọc những thông điệp của bạn. Kẻ tấn công sẽ khai thác nó bằng cách sử dụng các Public Key với các thông tin về User ID giả mạo trong đó.

Chính vì vậy giải pháp PGP (GnuPG - Gnu Privacy Guard) đã được ra đời để giải quyết vấn đề này. Public Key có thể được ký nhận bởi người sử dụng khác. Chữ ký này (Signatures) thừa nhận Key được sử dụng bởi UID (User Identification - Định danh người dùng) thực sự thuộc về những người sở hữu nó chính thức. Bạn có thể tin tưởng vào sự tin cậy của Key đó, khi bạn tin tưởng người gửi Key đó và bạn biết chắc chắn rằng Key đó thực sự thuộc quyền sở hữu của người gửi đó. Chỉ khi bạn thực sự tin tưởng vào Key của người ký nhận cũng như tin tưởng vào Signatures đó. Để tăng thêm tính tin tưởng vào Key bạn có thể so sánh Finger Print bởi các kênh đáng tin cậy

1.4 Những danh giới tới sự an toàn (Boundaries to security)

Nếu bạn có dữ liệu và bạn muốn giữ an toàn cho dữ liêuj của bạn. Khi đó bạn cần xác định nó sử dụng thuật toán mã hoá nào. Bạn đang nghĩ về sự an toàn một cách tổng thể cho hệ thống của bạn. Trên lý thuyết PGP được chúng ta coi là an toàn, nhưng khi bạn đọc tài liệu này đã có một số tính dễ tổn thương của PGP được biết đến. Trong cuộc sống không một điều gì có thể được coi là tuyệt đối. Tính an toàn của PGP cũng vây. Nhưng tôi dám khẳng định với bạn rằng việc tấn công PGP không phải là việc dễ. Đa số các cuộc tấn công đề xảy ra phần lớn do sự bất cẩn của người sử dụng. Chẳng hạn như việc đặt Password không tốt sẽ dẫn đến việc Secret Key bị Crack. Hay một vài nguyên nhân khác mặc dù khó xảy ra nhưng chúng ta không thể không đề phòng như: PC của bạn bị dính Trojan, Keylogger, một ai đó đọc các thông tin về PGP hiển thị trên màn hình của bạn...

Những sự kém an toàn tôi đã nêu ở trên không hề có ý gì khác chỉ mong bạn hiểu rằng. Không một công cụ nào có tính an toàn tuyệt đối cả (PGP cũng vậy). Bạn sẽ có sự an toàn tuyệt đối trên hệ thống của mình khi bạn thường xuyên để ý, thắt chặt và thực hiện nghiêm túc các chính sách và nội quy về bảo mật, an toàn hệ thống...

2. Cài đặt (Installtion)

2.1 Nguồn GnuPG (Source For GnuPG)

Bạn có thể lấy GnuPG từ Site chính thực của GnuPG (http://www.gnupg.org/) từ bạn có thể chọn các Mirmor có vị trí địa lý ở gần bạn để Download GnuPG về.

Nếu PC của bạn đã cài đặt các phiên bản của GnuPG hay PGP, bạn cần kiểm tra chữ ký của các Files.

2.2 Cấu hình (Configuration)

Bạn có thể Down về GnuPG ở dạng các gói Debian (Debian Packages), RPM (Redhat Packages Manager) hay ở dạng mã nguồn (Open Source). Thông thường ở các Distributed thông dụng của Linux như Redhat, Mandrake, SuSE...GnuPG thường được cài đặt mặc định kèm ngay khi cài đặt Linux. Để kiểm tra xem GnuPG đã được cài đặt trên hệ thống của bạn chưa ? Bạn dùng lệnh:

rpm -q gnupg

Thông thường các gói RPM thiết đặt các file nhị phân, những công cụ tài liệu trên nền tảng hệ thống Linux. Nếu bạn sử dụng các hệ thống khác như: Sun, *BSD, Unix...Bạn lên Down GnuPG ở dạng mã nguồn và trực tiếp cấu hình và biên dịch chúng để đạt được yêu cầu tối ưu hoá GnuPG cho hệ thống của bạn. Bạn có thể tìm được danh sách những hệ thống tương thích GnuPG trên Site chính thức đã nêu trên.

Việc cài đặt GnuPG từ các Packages khá đơn giản lên tôi không nhắc đến. Bây giờ tôi sẽ nêu qua các thao tác cấu hình và cài đặt GnuPG từ mã nguồn của nó.

Đầu tiên bạn hãy bung nén tập tin mã nguồn của GnuPG:

tar xvzf gnupg-?.?.?.tar.gz

?.?.? ở đây chính là thông tin về phiên bản GnuPG mà bạn Down về. Di chuyển vào thư mục vừa bung nén mã nguồn:

./configure

Nếu có khúc mắc bạn có thể sử dụng tuỳ chọn Help để biết thêm thông tin:

./configure --help

2.3 Biên dịch (Complie)

Để biên dịch GnuPG chúng ta gõ:

make

Mọi việc diễn ra xuôn xẻ thì không có gì để nói. Nếu như mọi việc diễn ra không như ý muốn. Bạn hãy tham khảo thêm thông tin về gỡ rối ở trang chủ của GnuPG (http://www.gnupg.org/) hoặc bạn hãy đưa vấn đề của mình lên Mail List của GnuPG.

2.4 Cài đặt (Installtion)

Bây giờ để cài đặt bạn gõ:

make install

Theo mặc định thì các trang man của GnuPG được chứa ở /local/share/gnupg. Bạn có thể chạy GnuPG như SUID root hay bất cứ SUID nào khác. Như vậy chương trình đó sẽ được chạy với tất cả đặc quyền mà User đó có. Việc này ngoại trừ khả năng những bộ phận nhất định của chương trình được lưu giữ ở một nơi khác và nó có thể bị sử dụng bởi một User khác trên cùng hệ thống đó. Nhưng cho lý do an toàn bạn không lên sử dụng GnuPG với SUID root. Nếu như không muốn GnuPG chạy với SUID root bạn thiết lập tuỳ chọn "no-secmen-warning" vào trong file:

~/.gnupg/options

3. Sử dụng khoá (Using Keys)

3.1 Tạo khoá (Creating Keys)

Với việc sử dụng lệnh:

gpg --gen-key

Một cặp khoá mới sẽ được tạo (gồm Secret Key và Public Key). GnuPG sẽ hỏi bạn sử dụng thuật toán mã hoá nào. Bạn có thể tham khảo thông tin về các thuật toán mã hoá:

- http://www.hertreg.ac.uk/ss/pgpfaq.html - PGP DH vs. RSA FAQ

2 thuật toán mã hoá được sử dụng rộng rãi là DSA và RSA. Tuy nhiên theo ý kiến của bản thân tôi thì RSA được sử dụng rỗng rãi và có khả năng mã hoá dữ liệu ở mức độ cao hơn DSA.

Lựa chọn kế tiếp của bạn sẽ là độ dài của Key (Key Lenght). Bạn cần lựa chọn giữa 2 tính năng sự bảo mật và thời gian. Nói một cách dễ hiểu nếu độ dài của Key lớn thì khả năng mã hoá thông điệp càng cao. Chính vì vậy thời gian mà PC của bạn dành để thực hiện công việc mã hoá và giải mã hoá sẽ lớn. Mặc định với GnuPG giá trị cực tiểu độ dài của key là 768 bits và giá trị cực đại là 2048 bits.

GnuPG sẽ lần lượt yêu cầu bạn vào các thông tin về bạn như: Họ và tên đầy đủ (Fullname), địa chỉ (Comment), địa chỉ mail (E-mail). Để làm cơ sở cho công việc tạo ra cặp khoá mới của GnuPG. Bạn có thể thay đổi các thông tin này sau.

Cuối cùng bạn sẽ phải nhập vào một Password (có chấp nhận ký tự Space). Nó được sử dụng để điều khiển Secret Key của bạn. Một Passphrase tốt chứa đựng những yếu tố sau:

- Nó phải có độ dài hợp lý
- Chứa đựng những ký tự đặc biệt

- Đảm bảo an toàn không bị suy đoán một cách dễ dàng (không sử dụng các thông tin liên quan đến bạn như: tên, ngày sinh, địa chỉ, số nhà...)

Lên nhớ rằng bạn không được phép quên Password đã nêu ở trên. Bởi nếu quên nó bạn sẽ không thể phục hồi lại nó cũng như điều kiểm soát Secret Key mà bạn đã tạo ra.

Sau cùng bạn dùng bàn phím nhập vào các ký tự ngẫu nhiên đủ yêu cầu số bit mà GnuPG cần để tạo ra một cặp khoá mới (để đảm bảo tính ngẫu nhiên và sự bảo mật cho cặp khoá mới). Bạn đợi trong giây lất, GnuPG đang phân tích, tính toán các thông tin mà bạn đưa vào để tạo ra cho bạn một cặp khoá mới. Quá trình này hoàn tất bạn sẽ có trong tay 2 Key: Public Key và Secret Key.

3.2 Xuất khoá (Exporting Key)

Để xuất Key cho một người dùng bạn sử dụng lệnh:

gpg --export [UID]

Nếu không có thông tin về UID thì tất cả các khoá hiện hành sẽ được xuất ra. Theo mặc định đã được thiết lập tới tuyến stdout. Khi sử dụng tuỳ chọn -o thông tin đó sẽ được gửi đến một file. Khi sử dụng tuỳ chọn -a nó sẽ ghi ra một khoá ASCII 7 bit thay cho file nhị phân Binary.

Chắc đến đây bạn sẽ đặt câu hỏi tại sao lại phải xuất Keys ? Hiểu một cách đơn giản khi xuất khoá bạn sẽ có khả năng trao đổi dữ liệu một cách an toàn với nhiều dùng khác trên Internet. Khi xuất Public Key bạn sẽ chia sẻ nó với bất cứ ai muốn trao đổi thông tin với bạn một cách an toàn. Bạn có thể Up Public Key mình lên một địa điểm nào đó trên Internet để chia sẻ với mọi người như:

- Up lên chính Homepage của bạn.
- Up lên các Key Server thông dụng như: http://www.pca.dfn.de/dfnpca/pgpkserv/
- Hay bất cứ phương pháp nào mà bạn cho là hợp lý...

3.3 Nhập khoá (Import Keys)

Khi bạn có được Public Key của một ai đó. Bạn cần phải Add nó vào Key Database của bạn để sau này sẽ sử dụng đến nó. Bạn sẽ dùng chính nó để giải mã hoá các dữ liệu đã được chính chủ nhân của nó mã hoá bằng Public Key mà bạn đang có ở các lần sau.

Để bổ xung một Public Key vào Key Database của mình bạn dùng lệnh:

gpg --import [filename]

Nếu giá trị filename bị bỏ xót thì giá trị thay thế mặc định sẽ là stdin

3.4 Huỷ bỏ khoá (Revoke A Keys)

Bởi một vài lý do như: Secret Key bị mất, UID bị thay đổi, nó không còn đáp ứng được các nhu cầu của bạn nữa...hay đơn giản là bạn không muốn sử dụng Key đó nữa. Bạn muốn huỷ bỏ chúng. Okies! hãy sử dụng lệnh:

gpg --gen-revoke

Để thực hiện điều này bạn cần một Secret Key khác để đảm bảo rằng chỉ có chủ sở hữu thực sự mới có quyền huỷ bỏ các Key đó. Lúc này! nếu như không biết Passphrase của Key đó thì mọi việc sẻ trở lên vô ích, thật bất lợi. Để khắc phục vấn đề này, GnuPG sẽ cấp cho bạn một sự cho phép huỷ bỏ Key "License Revoke" ngay khi bạn tạo một cặp khoá mới. Bạn lên cất giữ nó một cách cẩn thận...Bởi nếu bị lọt ra ngoài thì hậu quả của nó sẽ rất nghiêm trọng.

3.5 Quản lý Key (Administration Keys)

Ở trên tôi đã đề cập đến việc tạo, xuất, nhập và huỷ bỏ Key. Bây giờ chúng ta sẽ tiếp tục tìm hiểu về việc quản lý các Key.

Đầu tiên! để liệt kê tất cả các thông tin về Key được cất giữ trong Key Database của bạn:

gpg --list-keys

Để liệt kê những Signatures:

gpg --list-sigs

Để liệt kê thông tin về Fingerprint:

gpg --fingerprint

Để liệt kê thông tin về Secret Keys:

gpg --list-secret-keys

Trên đây là những lệnh được sử dụng để liệt kê và hiển thị các thông tin về Keys. Bây giờ sẽ là các lệnh được sử dụng để can thiệp trực tiếp vào các Keys đó.

Để xoá bỏ một Public Key

gpg --delete-key UID

Để xoá bỏ một Secret Key

gpg --delete-secret-key

Để chỉnh sửa thông tin về các Keys:

gpg --edit-key UID

Đây là một lệnh khá quan trọng trong quá trình sử dụng các Keys. Nó được sử dụng để thay đổi thông tin về thời hạn cuả Keys (Expiration Dates), thêm vào Fingerprint...cũng như chỉnh sửa các

thông tin quan trọng khác. Trước khi bắt đầu quá trình chỉnh sửa, để đảm bảo an toàn GnuPG sẽ yêu cầu bạn vào thông tin về Passphrase. Khi thông tin về Passphrase được nhập chính xác, bạn sẽ thấy một dòng đợi lệnh có dạng:

command>

4. Mã hoá và giải mã hoá (Encrypt And Decrypt)

Sau khi mọi công việc như cài đặt và cấu hình đã xong xuôi. Bây giờ chúng ta bắt đầu xem xét đến tính năng chính của GnuPG là mã hoá và giải mã hoá.

Bạn cần biết rằng trong quá trình mã hoá và giải mã hoá không chỉ cần Public Key và Secret Key của bạn mà còn cần đến Public key của những người mà bạn muốn trao đổi dữ liệu với họ một cách an toàn. Khi mã hoá một đối tượng dữ liệu cho người khác thì bạn sẽ phải chọn chính Public Key của họ để mã hoá nó. Sau đó gửi cho họ, họ sẽ dùng chính Secret Key của mình để giải mã hoá dữ liệu mà bạn đã mã hoá bằng chính Public Key của họ. Chính vì vậy phương pháp mã hoá dữ liệu này tỏ ra rất an toàn. Tuy để quá trình này diễn ra như ý muốn , trước hết bạn cần phải có Public Key của họ, tiếp đó bạn cần phải bổ xung Public Key của họ vào Database Key của bạn.

4.1 Mã hoá (Encrypt)

Để mã hoá dữ liệu bạn dùng lệnh

gpg -e Recipient [Data]

hay

gpg --encrypt Recipient [Data]

Lưu ý: Recipient chính là tên của người mà bạn muốn trao đổi thông tin mã hoá với họ. Trước khi muốn mã hoá dữ liệu và trao đổi với họ bạn phải có và đã bổ xung Public Key của họ vào Database Key của bạn. Nói một cách dễ hiểu tôi đã dùng chính Public Key của họ để mã hoá dữ liệu rồi gửi lại cho họ.

4.2 Giải mã hoá (Decrypt)

Quá trình giải mã hoá thì đơn giản hơn, sau khi nhận được dữ liệu đã mã hoá của tôi gửi cho. Về phía người nhận nếu họ muốn giải mã hoá, họ chỉ việc dùng lệnh:

gpg -d [Data]

hay

gpg --decrypt [Data]

Thực chất của quá trình giải mã hoá dữ liệu là người nhận sẽ dùng chính Secret Key của họ để giải mã hoá dữ liệu mà ta đã mã hoá bằng chính Public Key của họ. Dĩ nhiên, khi họ muốn trao

đổi dữ liệu mã hoá bằng GnuPG với tôi thì họ cũng làm những việc tương tự như đã nêu ở trên đối với tôi.

Bạn có thể sử dụng thêm tuỳ chọn "-o" để xuất nội dung giải mã hoá dữ liệu ra một file ở bên ngoài.

5 Quá trình ký nhận và kiểm tra chữ ký (Sign And Checking Signatures)

Thực chất có quá trình ký nhận và kiểm tra chữ ký nhằm mục đích tăng cường tính an toàn cho các phiên trao đổi dữ liệu mã hoá. Nó có tác dụng chứng thực người mã hoá giảm khả năng giả mạo người mã hoá và các Key mã hoá.

Để ký nhận dữ liệu bằng Key của mình bạn dùng lệnh:

gpg -s [Data]
hay
gpg --sign [Data]

Khi các kết quả được hiển thị không rõ ràng. Nếu như bạn muốn có một kết quả rõ ràng hơn bạn có thể sử dụng lệnh:

gpg --clearsign [Data]
Nếu bạn muốn tách riêng chữ ký của mình ra một file riêng biệt ? Tính năng này thường được sử dụng để mã hoá những file nhị phân (Binary). Bạn có thể sử dụng lệnh:

gpg -b [Data]
hay

gpg --detach-sign [Data]
Để an toàn bạn vừa muốn mã hoá dữ liệu lại vừa muốn ký nhận cho nó...Không vấn đề gì cả! Bạn có thể sử dụng lệnh sau:

gpg -u Sender -r Recipient --armor --sign --encrypt [Data]
hay

gpg --local-user --recipient --armor --sign --encrypt [Data]

Sau khi quá trình mã hoá và ký nhận dữ liệu được thực hiện. Theo nguyên tắc chữ ký sẽ được kiểm tra khi dữ liệu được giải mã hoá. Bạn có thể kiểm tra chữ ký của dữ liệu đã được ký nhận và mã hoá bằng lệnh:

gpg --verify [Data]

Tất nhiên quá trình này chỉ hoạt động và cho ra kết quả chính xác khi bạn đã bổ xung Public Key của người nhận vào Database Key của mình.

6 Front End

Nếu bạn không quen với việc sử dụng giao diện dòng lệnh. Có rất nhiều các khác để sử dụng GnuPG một cách dễ dàng hơn. Bạn có thể chọn và sử dụng những chương trình có giao diện đồ hoạ dễ sử dụng có hỗ trợ chuẩn mã hoá và giải mã của GnuPG. Những chương trình dạng này được gọi là những chương trình dạng Front End. Bạn có thể tìm thấy danh sách các chương trình GnuPG Front End ở:

- http://www.gnupg.org/frontends.html

6.1 Graphic Interface

GPA

GNU Privacy Assistant một chương trình GnuPG với giao diện đồ hoạ người dùng thân thiện. Với GPA bạn có thể xem Keyring của mình, xuất và nhập Key, tạo Key, Chỉnh sửa Key...Và tất cả các tính năng mà phiên bản GnuPG chuẩn ở dạng dòng lệnh có thể làm được. Để cài đặt GPA bạn chỉ việc Down gói Tarball về bung nén nó ra và thực hiệnc các câu lệnh sau:

```
./configure
make
make install
```

Khi mọi việc đã xong xuôi, để khởi sử dụng GPA bạn gõ:

```
gpa
```

Bạn có thể Download GPA ở:

- http://www.gnupg.org/gpa.html

Seahorse

Cũng là một chương trình ở dạng Front End của GnuPG được viết trên môi trường đồ hoạ GNOME. Bạn có thể sử dụng nó để Sign, Encrypt, Decrypt, Verify dữ liệu. Dữ liệu có thể được xuất vào bộ nhớ đệm Clipboard rồi nó sẽ được xử lý bởi Seahorse. Nó thực quản lý các Key cũng như chỉnh sửa các thuộc tính của các Key đã được lưu trữ trong Key Ring của bạn.

Bạn có thể Download Seahorse ở:

- http://seahorse.sourceforge.net/

Geheimnis

Là một người đồng nghiệp của GPA và Seahorse nhưng nó được viết cho môi trường đồ hoạ KDE.

Bạn có thể Downloaf nó ở:

- http://geheimnis.sourceforge.net/

6.2 Các chương trình E-Mail (E-Mail Programs)

Gần như các chương trình E-mail chuyên dụng đều hỗ trợ GnuPG:

- Mozilla
- Netscape Messenger
- Kmail
- Pine
- Eudora
- Mutt
- exmh

Bạn có thể sử dụng GnuPG kèm với các chương trình E-mail chuyên dụng để thực hiện công việc mã hoá và giải mã hoá dữ liệu với việc sử dụng các Public Key của những người mà bạn muốn trao đổi dữ liệu mã hoá.

Mozilla và Enigmail

Về mặc định chương trình hỗ trợ E-mail của Mozilla không hỗ trợ GnuPG. Để sử dụng GnuPG với Mozilla bạn phải sử dụng thêm một Plugin có tên là "Enigmail". Bạn có thể Download ở:

- http://enigmail.mozdev.org/

Enigmail là một Plugin cho Mozilla/Netscape Mail để sử dụng các tính năng mã hoá và giải mã hoá của các chương trình mã hoá thông dụng như: PGP và GPG...

Kmail

Một chương trình thư tín mặc định được tích hợp trên môi trường KDE. Mặc định nó hỗ trợ các chuẩn mã hoá như GnuPG và PGP. Bạn có thể ký nhận và giải mã hoá các thông điệp. Để thực hiện được điều này bạn phải cho Kmail biết "GnuPG User ID" của bạn. Thông thường khi gửi một thông điệp bạn sẽ không gặp phải yêu cầu ký nhận cũng như mã hoá thông điệp đó. Muốn Kmail mã hoá các thông điệp bằng GnuPG bạn phải nhấn vào biểu tưởng "ổ khoá" trên thanh công cụ.

7. Nơi cung cấp thông tin về GnuPG (Source Of Information)

Có rất nhiều nguồn để bạn có thể tham khảo thêm nhiều thông tin về GnuPG:

- Trang chủ của GnuPG - http://www.gnupg.org/
- Trang cnug cấp tài liệu, giải đáp các thắc mắc và Mail List của GnuPG - http://www.gnupg.org/docs.html
- Hay đơn giản bạn chỉ cần sử dụng tuỳ chọn -h của GnuPG

7.1 PGP

PGP là một chương trình mã hoá dữ liệu tương tự như GnuPG đã được ra đời từ nhiều năm trước. Đã có rất nhiều tài liệu viết về PGP. Đây có thể xem là nguồn thông tin rất hữu ích. Rất nhiều thông tin hữu ích có thể áp dụng được cho GnuPG:

- Trang chủ của PGP - http://www.pgpi.com
- Thông tin về 2 thuật toán RSA và PGP, đây là 2 thuật toán mã hoá được sử dụng trong GnuPG - http://www.hertreg.ac.uk/ss/pgpfaq.html

7.2 Key Server

Đây là các Server chuyên dụng để lưu trữ các Public Key của bạn. Bạn có thể upload Public Key của mình lên đây và trao đổi với mọi người:

- http://www.keyserver.net/
- http://france.keyserver.net/
- http://belgum.keyserver.net/
- http://www.keys.eu.pgp.net
- http://www.pca.dfn.de/dfnpca/pgpkserv/

7.3 Book

Có rất nhiều sách được viết về vấn đề này. Tuy nhiên trong khuôn khổ tài liệu được dịch ra từ phiên bản gốc thì chỉ thấy đề cập đến 2 quyển:

- Sách tiếng Anh được ấn bản với tên "Applied Cryptography, Second Edition" của B. Schneier (1996)
- Sách tiếng Đức được ấn bản với cái tên "Angewandte Kryptographie" của Addison-Wesley (1996)

7.4 Các nguồn tài liệu khác (Other Document)

Ngoài những nguồn tài liệu đã nêu trên bạn còn có thể kiếm được rất nhiều tài liệu khác về GnuPG qua các Search Engineer trên Internet.

8. About this document

8.1 Versions

Original German versions: Version 0.1 was the first version in German

Version 0.1.0 (English) April 30th 1999, Dutch Queen's Day.

- This version is the translation of the German version in English with some adjustments.

Version 0.1.1 (German)

- New section "Boundaries to security"
- Improved explanation of signatures
- Changes after comments from Werner Koch (thanks!)

Version 0.1.2 (English) April 3, 2002

- Corrected a few typos.
- New section about front ends.

Version 0.1.3 (Dutch) May 17, 2002

- This version is a translation of the English version into Dutch.

Version 0.1.3 (Vietnam) December 3, 2002

- This version is a translation of the English version into Vietnam.

All changes are documented in a diff file: dieses Dokument

For the English or Dutch version: All remarks for this document can be sent to Brenno J.S.A.A.F. de Winter (brenno@dewinter.com). or Arjen Baart (arjen@andromeda.nl). Comments help us make a better document and are greatly appreciated.

For the German version: Anregungen, Kritik, Verbesserungen und Erweiterungen einfach an Michael Fischer v. Mollard (fischer@math.uni-goettingen.de) senden, damit dieses Dokument weiter verbessert werden kann.