Sử dụng GNU Privacy Guard (GnuPG)

Lâm Vĩnh Niên
nien.lam@gmail.com

Phiên bản: v1.2 (050604)

1 Giới thiệu

GNU Privacy Guard (GnuPG) là công cụ dùng để lưu trữ an toàn dữ liệu và thông tin. Nó cũng được dùng để mã hoá dữ liệu và tạo chữ kí số. GnuPG tuân theo chuẩn OpenPGP, như vậy nó tương thích với PGP (NAI, Inc.). GnuPG được phát hành dưới giấy phép GPL.

GnuPG có thể hoạt động trên nhiều môi trường: GNU/Linux, FreeBSD, OpenBSD, NetBSD, Windows 95/98/ME/NT/2000/XP, MacOS X,...

Bản mới nhất của tài liệu này có thể được tìm thấy ở http://labang.sourceforge.net.

2 Cài đặt

Hầu hết các bản phân phối Linux đều có gói GnuPG được đóng sẵn. Nếu muốn biên dịch từ nguồn, có thể tải gói nguồn xuống từ trang chủ của GnuPG tại http://www.gnupg.org/, giải nén và chạy lần lượt các lệnh

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

Có thể xem các tuỳ chọn cấu hình với lệnh `./configure --help`.

Trên Windows, có thể giải nén vào `c:\gnupg`; nếu cài đặt vào một thư mục không phải `c:\gnupg`, cần thay đổi các đường dẫn trong tập tin `gnupg-w32.reg` (bằng Notepad chẳng hạn), rồi chạy lệnh

```
start c:\đường\đến\gnupg-w32.reg
```

Sau đó cần thêm thư mục cài đặt GnuPG (`c:\gnupg` hoặc nơi đã chọn khác) vào biến môi trường PATH: trên Windows 95/98/ME biên tập tập tin `c:\autoexec.bat` (các đường dẫn cách nhau bằng dấu `;`), trên Windows NT/2000/XP vào Control Panel –> System Properties –> Advanced tab –> Environment Variables –> System variables.

3 Khoá công cộng và khoá cá nhân

Mỗi người dùng có một cặp khoá gồm *khoá cá nhân* (private key) và *khoá công cộng* (public key). Khoá cá nhân cần được giữ bí mật; còn khoá công cộng có thể được đưa cho người cần liên lạc. Một tài liệu (văn bản hoặc nhị phân) có thể được mã hoá với bất cứ khoá nào trong hai khoá đó và được giải mã với khoá kia. Ví dụ, A muốn gửi cho B một tài liệu, A sẽ dùng khoá công cộng của B để mã hoá; khi nhận, B sẽ dùng khoá cá nhân của mình để giải mã và không ai khác có thể giải mã. Ở trường hợp khác, B muốn

bảo đảm là tài liệu được gửi từ A, A sẽ dùng khoá cá nhân của mình để mã hoá hoặc kí tài liệu và B sẽ dùng khoá công cộng của A để mở tài liệu hoặc xác nhận chữ kí.

4 Tạo cặp khoá

```
gpg --gen-key
```

Quá trình tạo khoá sẽ đưa ra một số câu hỏi. *Loại cặp khoá* nên dùng mặc định (khoá chữ kí DSA - khoá chính, và khoá mã hoá ElGamal - khoá phụ). Khoá chữ kí dùng để tạo chữ kí số, và nó cũng thu thập các chữ kí của những người khác đã xác nhận danh tính của bạn. Khoá mã hoá dùng để giải mã tài liệu được mã hoá. *Kích thước khoá* mặc định là 1024 bit đủ dùng trong phần lớn trường hợp (ngay cả với chữ kí ElGamal). Kích thước khoá DSA là từ 512 đến 1024 bit, còn khoá ElGamal có thể dùng với bất kì kích thước nào. Với loại cặp khoá DSA và ElGamal, nếu kích thước đưa ra lớn hơn 1024 bit thì khoá DSA vẫn có kích thước là 1024, còn khoá ElGamal sẽ có kích thước được yêu cầu. Về *thời gian hết hiệu lực*, thông thường chữ kí được dùng suốt đời (để tiếp tục sử dụng các tài liệu đã kí), còn khoá phụ mã hoá thì được thay đổi định kì (để tăng tính bảo mật cho tài liệu). *Căn cước người dùng* (user ID) cần được cung cấp để tạo liên hệ giữa khoá với người thực và *cụm từ vượt* (passphrase) được dùng để bảo vệ khoá cá nhân. Cụm từ vượt không bị giới hạn số kí tự và có thể chứa khoảng trắng. Để phát sinh cặp khoá, GnuPG cần thu thập các bit ngẫu nhiên qua hoạt động của máy tính; do đó có thể thúc đẩy quá trình này bằng cách gõ vài kí tự ngẫu nhiên vào bàn phím.

5 Xuất, nhập và trao đổi khoá

5.1 Nhập xâu khoá đã có

Trường hợp này gặp khi cần dùng bộ xâu khoá trên nhiều máy tính.

```
gpg --import /đường/dẫn/secring.pgp
```

để nhập xâu khoá cá nhân. Và rất có thể sẽ cần nhập xâu khoá công cộng:

```
gpg --import /đường/dẫn/pubring.pgp
```

Ngoài ra, cũng có thể nhập từng khoá thay vì nhập cả xâu khoá.

Tuy nhiên, việc nhập khoá hay xâu khoá không phải là giải pháp tốt nếu thao tác trên máy tính không phải của chính người dùng. Khi đó, có thể chép các xâu khoá vào thiết bị lưu trữ ngoài (đĩa mềm, CD,...) và trực tiếp dùng chúng bằng các tuỳ chọn `--keyring`

`xâu_khoá_công_cộng` hoặc `--secret-keyring xâu_khoá_cá_nhân` (có thể thêm tuỳ chọn `--no-default-keyring` để không dùng các xâu khoá mặc định). Nếu không ghi rõ đường dẫn cho các xâu khoá thì chúng sẽ được tìm trong thư mục nhà (`~/.gnupg` nếu không dùng `--homedir`).

5.2 Công bố khoá công cộng

```
gpg --output mypubkey.gpg --export uid
```

sẽ xuất ra khoá công cộng ở dạng nhị phân. *uid* có thể là *Real Name* hoặc địa chỉ email có ở căn cước người dùng hoặc 8 kí tự cuối cùng của dấu vân tay (viết liền, có thể thêm 0x phía trước).

Để thuận tiện cho việc trao đổi qua email hoặc xuất bản trên web, khoá công cộng có thể được bọc lớp vỏ ASCII. Nói chung, bất cứ các kết quả xuất nào từ GnuPG (khoá, tài liệu mã hoá, chữ kí) cũng đều có thể bọc lớp vỏ ASCII với tuỳ chọn `--armor`.

```
gpg --armor --output mypubkey.txt --export myname
```

Tập tin *mypubkey.txt* có dạng

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.4 (GNU/Linux)

[...]
-----END PGP PUBLIC KEY BLOCK-----
```

Chép toàn bộ nội dung tập tin này, gồm cả các dòng bắt đầu với "`------`" vào trang web hoặc nội dung email để công bố khoá công cộng.

Lưu ý: Có thể xuất khoá cá nhân theo cách này, với tuỳ chọn `--export-secret-keys` thay vì `--export`. Khi công bố khoá, cần kiểm tra dòng đầu tiên chứa chữ `PUBLIC` hay `PRIVATE` và dĩ nhiên không công bố khoá cá nhân.

5.3 Nhập một khoá công cộng

```
gpg --import pubkey
```

pubkey là khoá công cộng cần nhập, có thể ở dạng nhị phân hoặc ASCII. Với khoá dạng ASCII, GnuPG chỉ đọc những dòng nằm trong khối tạo bởi `BEGIN PGP PUBLIC KEY BLOCK` và `END PGP PUBLIC KEY BLOCK`.

Một khi khoá được nhập, nó cần phải được *xác thực* (validated).

```
gpg --edit-key uid
```

sẽ vào chế độ dòng lệnh nội tại (*uid* có thể là *Real Name* hoặc địa chỉ email trong căn cước người dùng của khoá vừa nhập):

```
Command>
```

Nên kiểm tra đối chiếu dấu vân tay của khoá (lệnh `fpr`) với người chủ trước khi kí (lệnh `sign`) khoá để xác thực. Khoá đã kí có thể được kiểm tra (lệnh `check`) để liệt kê các chữ kí có trên nó – mỗi căn cước người dùng có một hoặc nhiều chữ kí tự thân (self-signature) và một chữ kí cho mỗi người đã xác thực khoá.

6 Mã hoá và giải mã tài liệu

Trường hợp này dùng khoá công cộng để mã hoá và dùng khoá cá nhân để giải mã.
Tuỳ chọn `--encrypt` được dùng để mã hoá tài liệu.

```
gpg --output doc.gpg --encrypt --recipient uid doc
```

Tài liệu cần mã hoá là *doc* và tài liệu xuất đã được mã hoá là *doc.gpg*; `--recipient` chỉ định khoá công cộng được dùng, có thể dùng nhiều `--recipient`.
Giải mã tài liệu với tuỳ chọn `--decrypt`.

```
gpg --output doc --decrypt doc.gpg
```

Quá trình này dùng khoá cá nhân và cần nhập cụm từ vượt.
Để giải mã tài liệu với xâu khoá bên ngoài:

```
gpg --secret-keyring xâu_khoá -output doc -decrypt doc.gpg
```

Tài liệu cũng có thể được mã hoá mà không cần khoá công cộng, được dùng khi muốn bảo mật tài liệu.

```
gpg --output doc.gpg --symmetric doc
Enter passphrase:
```

Cụm từ vượt ở đây nên khác với từ vượt dùng bảo vệ khoá cá nhân.

7 Tạo và xác nhận chữ kí

Trong trường hợp này, chữ kí được tạo từ khoá cá nhân và được xác nhận bằng khoá công cộng tương ứng.

7.1 Tạo tài liệu nhị phân

```
gpg --output doc.sig --sign doc
```

Tài liệu *doc* sẽ được nén trước khi kí và kết quả xuất *doc.sig* ở dạng nhị phân.
Một tài liệu đã được kí có thể được kiểm tra chữ kí (`--verify`) hoặc vừa kiểm tra chữ kí và vừa khôi phục tài liệu (`--decrypt`).

```
gpg --output doc --decrypt doc.sig
```

7.2 Tạo tài liệu *clearsigned*

Thường thông điệp email không cần (và cũng không được mong muốn) nén khi kí. Tuỳ chọn `--clearsign` gói tài liệu trong chữ kí dạng ASCII mà không thay đổi tài liệu.

```
gpg --clearsign doc
```

sẽ được tập tin *doc.asc* có dạng

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

[...]
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.2.4 (GNU/Linux)

iD8DBQFByvLH4+iEI365Zq8RAmiEAKCDEv6WDgjfTrWND4mmmHNm2r/2QwCg0+M+
Qo5/YuYlQonM50WUDvNfGAg=
=DnKK
-----END PGP SIGNATURE-----
```

Dùng tuỳ chọn `--verify` để xác nhận chữ kí.

7.3 Chữ kí tách rời

Chữ kí và tài liệu tách rời nhau, nhờ đó không cần phải khôi phục tài liệu đã được kí.

```
gpg --output doc.sig --detach-sig doc
```

Việc xác nhận chữ kí cần cả tài liệu và chữ kí tách rời.

```
gpg --verify doc.sig doc
```

7.4 Cùng lúc mã hoá và kí tài liệu

```
gpg -sear uid doc
```

với `s`: sign, `e`: encrypt, `a`: armor, và `r`: recipient. Bỏ tuỳ chọn `a` nếu muốn tạo tập tin nhị phân.

8 Quản lí khoá

8.1 Thao tác trên xâu khoá

Trên xâu khoá công cộng (`~/.gnupg/pubring.gpg`):

```
gpg --list-keys
```

liệt kê các khoá

```
gpg --list-sigs
```

như `--list-keys`, nhưng kèm các chữ kí

```
gpg --fingerprint
```

như `--list-keys`, nhưng kèm các dấu vân tay

```
gpg --delete-key uid
```

xoá khoá khỏi xâu khoá.

Trên xâu khoá cá nhân (`~/.gnupg/secring.gpg`):

```
gpg --list-secret-keys
```

liệt kê các khoá

```
gpg --delete-secret-key uid
```

xoá khoá khỏi xâu khoá cá nhân và công cộng.

Có thể thêm tham số *uid* vào các lệnh liệt kê nêu trên để xem khoá của riêng uid đó.

8.2 Thao tác trên khoá

8.2.1 Xem cặp khoá

```
gpg --edit-key uid
```

hiển thị khoá công cộng cho dù khoá cá nhân có hay không. Cột đầu tiên cho biết loại khoá (`pub`: khoá công cộng chính (public master signing key), `sub`: khoá công cộng phụ (public subordinate key)). Cột thứ hai cho biết chiều dài bit, loại (`D`: khoá DSA, `g`: khoá ElGamal chỉ dùng mã hoá, và `G`: khoá ElGamal có thể dùng cho mã hoá và kí), và ID của khoá. Cột thứ ba và thứ tư cho biết ngày tạo và ngày hết hạn của khoá. Tiếp theo các khoá là danh sách căn cước người dùng.

Lệnh `toggle` chuyển qua lại giữa khoá công cộng và khoá cá nhân, nếu cả hai cùng có mặt. Thông tin ở khoá cá nhân được hiển thị tương tự như ở khoá công cộng. Từ `sec` cho biết khoá cá nhân chính (private master signing key) và `sbb` cho biết khoá cá nhân phụ (private subordinate key).

8.2.2 Thêm và xoá thành phần của khoá

Thêm căn cước người dùng: `adduid`, có ích khi cần nhiều căn cước, thí dụ căn cước riêng cho nơi làm việc, cho hoạt động xã hội, cho gia đình,...

Thêm khoá phụ (subkey): `addkey`. Căn cước người dùng đi kèm khoá công cộng chính được người liên hệ bên kia xác thực, và họ cần tái xác nhận mỗi khi khoá chính thay đổi. Việc này sẽ khó khăn và tốn thời gian nếu người dùng liên lạc với nhiều người khác. Định kì thay đổi khoá phụ là điều tốt, vì nếu một khoá bị phá thì chỉ những tài liệu mã hoá với khoá đó bị lộ.

Xoá căn cước người dùng và khoá phụ: `deluid` và `delkey`. Trước đó cần chọn căn cước hoặc khoá phụ muốn thao tác (lệnh `uid` hoặc `key <số thứ tự của căn cước hoặc khoá phụ>`). Xoá các thành phần của khoá để bỏ những thông tin không cần thiết khỏi khoá công cộng của người khác thì tốt, nhưng có thể không hay khi xoá chúng từ chính khoá của mình, vì khi người khác cập nhật khoá công cộng mới, thông tin sẽ được trộn lẫn và anh ta vẫn giữ khoá mà người kia đã xoá. Để cập nhật đúng cách, anh ta phải xoá phiên bản khoá cũ và nhập phiên bản mới. Đó là gánh nặng cho anh ta. Vì vậy, khi cập nhật khoá của chính mình người dùng nên thu hồi các thành phần khoá thay vì xoá chúng.

8.2.3 Thu hồi các thành phần của khoá

Thu hồi khoá phụ với lệnh `revkey` sau khi chọn khoá phụ cần thao tác.

Thu hồi căn cước người dùng bằng cách thu hồi chữ kí tự thân (self-signature) trên căn cước người dùng đó. Chữ kí được thu hồi bằng lệnh `revsig`, sẽ có dấu nhắc để quyết định những chữ kí nào cần thu hồi. Lệnh `check` liệt kê những chữ kí nào đã bị thu hồi.

8.2.4 Cập nhật thời gian hết hiệu lực

Cập nhật thời gian hết hiệu lực (expiration time) của một khoá bằng lệnh `expire`. Nếu không chọn khoá nào thì khoá chính sẽ được cập nhật.

8.2.5 Xác thực khoá

Phần 5.3 đề cập việc xác thực từng khoá, cách này sẽ khó khăn nếu có số lượng lớn khoá cần xác thực. GnuPG giải quyết vấn đề này với *mạng lưới tin tưởng* (web of trust), ở đó việc xác thực một khoá công cộng được uỷ thác cho những người được tin tưởng (đánh giá qua lệnh `trust`).

8.3 Tạo chứng chỉ thu hồi

Chứng chỉ thu hồi nên được tạo ngay lập tức sau khi tạo cặp khoá.

```
gpg --output revoke.asc --gen-revoke mykey
```

Chứng chỉ này có thể được công bố để báo cho người khác biết khoá công cộng không còn được sử dụng. Trường hợp này gặp khi người dùng quên cụm từ vượt, khoá cá nhân bị mất hoặc lộ,...

Khoá công cộng bị thu hồi vẫn có thể được dùng để xác nhận chữ kí trong quá khứ, nhưng không thể được dùng để mã hoá thông điệp gửi đến người chủ. Nó cũng không ảnh hưởng đến việc giải mã thông điệp cũ nếu người dùng vẫn còn khoá cá nhân.

8.4 Tạo khoá công cộng từ khoá cá nhân

Việc này có thể thực hiện từ phiên bản GnuPG 1.2.1.

```
gpgsplit --no-split --secret-to-public secret.gpg >publickey.gpg
```

Trước hết nên xuất khoá cá nhân và chỉ tạo khoá công cộng từ nó; tuy nhiên, dùng cả xâu khoá cá nhân vẫn được.

8.5 Phân phối khoá

Khoá có thể được trao trực tiếp giữa người sử dụng. Trong thực tế, khoá thường được gửi qua email, nếu số lượng trao đổi ít, hoặc đặt trên một trang web cá nhân hoặc trên máy phục vụ công cộng lưu trữ chữ kí (public key server). Lưu trữ chữ kí trên key server có ưu điểm là nhiều người có thể truy cập và lấy chữ kí về.

Gửi khoá lên key server:

```
gpg --keyserver key_server --send-keys uid
```

Gọi khoá từ key server hoặc kiểm tra chữ kí mới:

```
gpg --keyserver key_server --recv-keys uid
```

Có thể liệt kê nhiều *uid* cùng lúc. Nhiều key server công cộng có thể tìm thấy ở các tên miền pgp.net và keyserver.net; dùng lệnh "host -la <domain>" hay "dig @<server> axfr <domain>" để có danh sách mới nhất. Có thể dùng key server tại www.keyserver.net, subkeys.pgp.net, pgp.mit.edu,... Một số nơi có thể gửi và lấy khoá bằng giao diện web, như http://keyserver.kjsl.com:11371/, http://keyserver.kjsl.com/pks/, http://pgp.nic.ad.jp/,... Có thể dùng tuỳ chọn `--search-keys` để tìm kiếm khoá.

9 Thuật ngữ

Public key: khoá công cộng

Private key: khoá cá nhân

Passphrase: cụm từ vượt

Validate: xác thực

Expiration time: thời gian hết hiệu lực, thời gian hết hạn

Web of trust: mạng lưới tin tưởng

Revocation certificate: chứng chỉ thu hồi

Digital signature: chữ kí số

Self-signature: chữ kí tự thân

Keyring: xâu khoá

10 Tài liệu tham khảo

1. The GNU Privacy Handbook - The Free Software Foundation - 1999
2. GnuPG Frequently Asked Questions, Version 1.6.3 - The Free Software Foundation - Jul 30, 2003
3. A Practical Introduction to GNU Privacy Guard in Windows - Brendan Kidwell - Nov 8, 2003