Lathund för kryptering med GnuPG

STTS Södermalms talteknologiservice
http://stts.se

23 juni 2011

Innehåll

1 GnuPG

GnuPG är en fritt tillgänglig variant av pgp-metoden för kryptering, som går ut på att man har en öppen och en hemlig nyckel—den öppna delar man med sig av till den som man skall byta data med, den hemliga håller man hemlig. För att kryptera data till en viss mottagare, behöver man mottagarens öppna nyckel. Den hemliga nyckeln skyddas av ett (längre) lösenord ('passphrase'). Slarvar man bort nyckeln, och/eller glömmer lösenordsfrasen, och inte har tillgång till den okrypterade originaldatan, ligger man pyrt till.

Nycklarna sparas i en "nyckelknippa" ('keyring'), en för hemliga, och en för öppna nycklar. GnuPG fungerar på en rad operativsystem. Allt stoff i detta dokument, och mycket mer därtill, finns att insupa på `http://gnupg.org`.

Här följer en kort översikt över hur det går till när person A skall kryptera data och skicka till person B:

- person A och person B genererar egna öppna och hemliga nycklar med `gpg` (om dom inte redan gjort det)
- person B skickar sin öppna nyckel till person A
- person A importerar person B:s öppna nyckel till sin `gpg`-nyckelknippa
- person A använder person B:s öppna nyckel för krypteringen av datan med `gpg`
- person A skickar datan till person B, som använder `gpg` för att dekryptera den

2 Installation

Hämta GnuPG i en passande variant från `http://gnupg.org`. Klicka på `Download` i menyn till vänster och gå till avsnittet `Binaries`. Öppna installationsprogrammet och följ instruktionerna.

Du kan också hämta ner det med Yum, Yast, Apt-get eller liknande, om du använder ett system med någon av dessa metoder för automatisk installation.

3 Nyckelgenerering

GnuPG installeras på Windows i katalogen `C:\Program\GNU\GnuPG` eller liknande. I katalogen finns programmet `gnu.exe`. Starta kommandotolken eller motsvarande för att köra GnuPG. För enkelhetens skulle är sökvägen till GnuPG borttagen i exemplen nedan, men troligen måste du ange hela sökvägen, exempelvis `C:\Program\GNU\GnuPG\gpg.exe`.

Generera ditt nyckelpar med kommandot

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

Här följer en skärmutskrift på hur det kan se ut (på Linux):

```
gpg --gen-key
gpg (GnuPG) 1.4.2; Copyright (C) 2005 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.
```

```
Please select what kind of key you want:
   (1) DSA and Elgamal (default)
   (2) DSA (sign only)
   (5) RSA (sign only)
Your selection?
```

Välj `1` (eller tryck return)

```
DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
```

Välj `2048` genom att trycka return.

```
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
```

Även här kan man gott välja förslaget—det går att själv bestämma att en nyckel inte längre skall vara giltig (sök exempelvis efter *revoke* i manualen).

```
Key does not expire at all
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name:
```

Fyll i förnamn och efternamn, epostadress samt en kommentar (i exemplet nedan är kommentaren `STTS`).

```
You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Nikolaj Lindberg
Email address: EN@EPOSTADRESS.se
Comment: STTS
You selected this USER-ID:
    "Nikolaj Lindberg (STTS) <EN@EPOSTADRESS.se>"
```

(Där `EN@EPOSTADRESS.se` ska vara användarens riktiga epostadress.)

Sedan kommer `gpg` fråga om lösenordsfrasen, som man skriver in två gånger. (Under tiden nycklarna genereras, föreslår `gpg` att man exempelvis skall röra pekdonet och använda tangentbordet, för att hjälpa slumpgenereringen på traven.)

```
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
.+++++++++++++++++++++++++++++.+++++++++++++++++++++++++++++++
++++++++++++++++++.++++++++++++++++++++.++++++++++.++++++++++
+++++++++>.++++++++++...+++++
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++++++++++++++++++++++++++..++++++++++..++++++++++.+++++.+++++.
++++++++++..++++++++++.....++++++++++...++++++++++++++++++++.++++++
++++..++++++++++.+++++.+++++.+++++>.+++++.+++++...................
..>..+++++..<+++++.........>..+++++.................
gpg: key 0F8E53F7 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   1024D/0F8E53F7 2005-12-28
      Key fingerprint = F086 79AE 4B79 E65F B85E  02A8 5C65 842A 0F8E 53F7
uid                  Nikolaj Lindberg (STTS) <EN@EPOSTADRESS.se>
sub   2048g/401D34B2 2005-12-28
```

De filer som genereras hamnar på unix/linux i katalogen `~/.gnupg`, och hamnar på Windows troligen i katalogen `C:\Documents and settings\UID\Application Data\gnupg` (byt ut `UID` mot ditt användarnamn). Dessa filer skall dock inte behöva hanteras manuellt, eftersom `gpg` har kommandon för det mesta man kan behöva göra.

4 Exportera öppen nyckel

Med kommandot `gpg --export [UID]`, där UID betyder användarnamn, kan man ”exportera” den öppna nyckeln, för att skicka den till de som vill kryptera data åt UID.

Följande skriver ut användare `nikolaj`s öppna nyckel till filen `nyckel`, i ASCII-format (`-a`):

```
gpg -a -o nyckel --export nikolaj
```

Nu kan alltså filen `nyckel` skickas till den som skall kryptera data med användare `nikolaj` som mottagare. För att detta skall fungera, skall den som krypterar datan först importera nyckeln.

5 Importera nyckel

För att lägga till någons öppna nyckel till nyckelknippan, används

```
gpg --import [Filnamn]
```

Filen `nyckel` i exemplet ovan, importeras alltså med

```
gpg --import nyckel
```

6 Kryptering

```
gpg -r [Recipient] -e [Data]
```

För att kryptera filen `hemligt.txt` med `nikolaj` som mottagare:

```
gpg -r nikolaj -e hemligt.txt
```

Den krypterade versionen av filen får namnet `hemligt.txt.gpg`.

6.1 Kryptering för flera mottagare

Man kan kryptera samma fil för flera mottagare. Det är bara att ange fler mottagare med `-r`. Här är ett exempel på hur man krypterar en fil med både `nikolaj` och `hanna` som mottagare:

```
gpg -r nikolaj -r hanna -e hemligt.txt
```

(Det kan vara smidigt att ta med sig själv som mottagare, om man vill ha möjligheten att dekryptera filer man skickat till andra.)

7 Dekryptering

Mottagaren får fram datan i klartext med kommandot

```
gpg (-d) [Data]
```

exempelvis

```
gpg hemligt.txt.gpg
```

8 Arbetsgång

Här följer ett exempel på hur arbetsgången kan se ut i ett projekt där STTS och en kund byter data:

1. STTS och Kunden genererar hemliga och öppna nycklar
2. Vi byter öppna nycklar

3. Kunden krypterar data, med hjälp av STTS öppna nyckel
4. Data levereras till Kundens konto på STTS internetserver, via `sftp` eller `scp`
5. STTS hämtar datan, dekrypterar på den lokala servern, som inte är tillgänglig utifrån, och utför arbetet
6. STTS krypterar resultatet av arbetet med kundens öppna nyckel, och lägger upp på internetservern, i Kundens konto
7. Kunden hämtar datan, och dekrypterar på egen server

(De två första stegen ovan skall endast behöva genomföras vid första leveransen—fortsättningsvis används samma nycklar.)

9 Övrigt

Användbara kommandon för att hantera sin nyckelknippa (läs mer i den riktiga manualen):

```
gpg --list-keys

gpg --list-secret-keys

gpg --edit-key [UID]
```

Det finns grafiska program för att hantera GnuPG i Windows, exempelvis gpg4win[1]. STTS har inte provat det, men det verkar bra. Med gpg4win slipper man köra GnuPG från kommandotolken. När man högerklickar på en fil kan man välja att kryptera och dekryptera filer, och det finns ett grafiskt gränssnitt för att hantera nycklar.

Om du inte redan har ett bra SFTP/SCP-program finns FileZilla[2], som fungerar på de flesta operativsystem. För Windows finns även WinSCP[3]. För Linux/Unix finns också `sftp`- och `scp`-kommandon.

[1] www.gpg4win.org
[2] filezilla-project.org/
[3] www.winscp.net