

GPG in Linux

Chari Karipidis
2TinG

May 20, 2012

Contents

1	Introductie	3
2	GPG	4
2.1	Geschiedenis	4
2.2	Wat is GPG?	5
2.3	GPG in command-line interface (CLI)	5
2.3.1	Voorbeelden van vaak gebruikte commando's	6
2.3.2	Voorbeelden van vaak gebruikte opties	6
2.3.3	Voorbeelden van het gebruik	6
2.4	Werking en uitvoer van GPG	7
2.5	Frontend GPG programma's	8
2.6	Werking van een GPG Frontend	8
3	Belangrijke woorden	13
4	Referenties	14

List of Figures

1	uitvoer in terminal	7
2	Cryptophane	9
3	Gajim	9
4	GnuPG Shell	10
5	GPA	10
6	KGpg	11
7	Seahorse	11
8	Wija	12

List of Tables

1	vaak gebruikte commando's	6
2	vaak gebruikte opties	6
3	voorbeelden	6
4	Gui Frontends	8
5	Belangrijke woorden	13

1 **Introductie**

In dit document wordt het gebruik van GPG(GnuPG) nader verklaard.

Benadering De geschiedenis, werking en uitvoer wordt uitgewerkt in Sectie 2 GPG.

Sectie 3 Belangrijke woorden, geeft een overzicht van belangrijke woorden in het document.

2 GPG

2.1 Geschiedenis

Er is altijd wel een probleem met boodschappen verzenden en ontvangen, zonder dat men deze kunnen onderscheppen en lezen. Hier zijn handige uitvindingen voor ontworpen, die helpen bij dit probleem.

Scytale In de tijd van de Romeinen had men een manier nodig om berichten te versturen naar geallieerde troepen. Verzender en ontvanger waren in het bezit van een 'Scytale' van ieder dezelfde grootte. Dit voorwerp was een soort van cilinder. Hier werd een riem over gewikkeld en een boodschap op geschreven.

Bij het verwijderen van de riem, was deze tekst onleesbaar zonder behulp van de Scytale. De letters waren namelijk door mekaar. Bij ontvangst van de riem bij de troepen, wikkelde ze de riem over de Scytale die zij bezitte en was het zo mogelijk, de boodschap te lezen.

Dit was een soort van encryptie. Ervoor zorgen dat een onderschepper, de boodschap niet kan lezen.

Caesar methode Een andere encryptie-methode was de Caesar methode. Deze bestond uit een zin hervormen m.b.v. het alfabet. Dit klinkt natuurlijk zeer logisch. Het alfabet wordt namelijk gebruikt om zinnen te schrijven.

Maar na het schrijven van de nodige boodschap, wordt er een 'sleutel' gekozen. Deze sleutel is afgesproken cijfer tussen 1 en 26, tussen beide partijen.

Belangrijk is dat de cijfers overeenkomen met een letter uit het alfabet. Als het gekozen cijfer, 6 is. Wordt het alfabet 6 maal naar links verschoven. A wordt dan F en B wordt dan G, enz...

De ontvanger krijgt dan een wirwar van letters en kan deze ontcijferen door het alfabet terug te vormen voor het 6 maal naar rechts te verschuiven.

Tegenwoordig worden loopjongens niet meer gebruikt. Men is mee geevolueerd naar de toekomst.

Technology is nu de heerser over het verzenden van boodschappen. Mailen, accounts aanmaken, bestanden opslaan, enz... Gebeurt iedere dag. Dit moet dan ook beveiligd worden.

Een manier voor encryptie is GPG.

2.2 Wat is GPG?

GPG of GnuPG staat voor; Gnu Privacy Guard. Zoals de naam al voorstelt, is het om de privacy van gebruikers te beschermen. Dit doormiddel van encryptie van; boodschappen die verzonden moeten worden zoals mails, data encrypteren, 'sleutelhangers', enz...

GnuPG is een commando voor de terminal, Te zien in Subsectie 2.3, maar er zijn dergelijke frontend programma's om deze in een gui te kunnen gebruiken. Te zien in Subsectie 2.5

2.3 GPG in command-line interface (CLI)

Zoals ieder ander command, heeft GPG ook zijn nodige syntax.

gpg[- - *homedirname*][- - *optionsfile*][*options*]*command*[*args*]

Het command 'GPG' heeft een enorm aantal aan opties. In *man**gpg* worden de opties weergegeven, met de nodige uitleg.

In subsecties 2.3.1, 2.3.2, 2.3.3, worden voorbeelden weergegeven van welke er het meest gebruikt worden.

2.3.1 Voorbeelden van vaak gebruikte commando's

<i>-c</i>	symmetrische encryptie, vraagt voor passphrase.
<i>- - decrypt</i>	decryptie van geïncrypteerde bestanden.
<i>- - encrypt</i>	Encryptie van data. Wordt gecombineerd met - sign.
<i>-sign</i>	maakt handtekening, wordt gecombineerd met - encrypt.
<i>- - encrypt - files</i>	encryptie van meerdere bestanden in 1 commando.
<i>- - decrypt - files</i>	decryptie van meerdere bestanden in 1 commando.

Table 1: vaak gebruikte commando's

2.3.2 Voorbeelden van vaak gebruikte opties

<i>-ofile</i>	schrijft output naar 'file'.
<i>- - default - keyname</i>	standaard waarde voor ID encryptie.
<i>-rname</i>	encryptie naar ontvanger 'name'.
<i>-v</i>	Verbose, geeft meer info tijdens het proces.
<i>-i</i>	Interactief, geeft prompts voor iedere stap.

Table 2: vaak gebruikte opties

2.3.3 Voorbeelden van het gebruik

<i>gpg -r Bobfile</i>	handteken en encrypt voor Bob.
<i>gpg - -clearsignfile</i>	maakt een lege handtekening.
<i>gpg - -fingerprintuser_ID</i>	laat vingerafdruk zien.
<i>gpg - -verifypgpfile</i>	verifieert pgpfile.
<i>gpg - -list - keysuser_ID</i>	laat sleutels zien.

Table 3: voorbeelden

2.4 Werking en uitvoer van GPG

Het bestand 'GPGtest' in de directory '/home/GPG/testfiles' Moet geïncrypteerd worden.

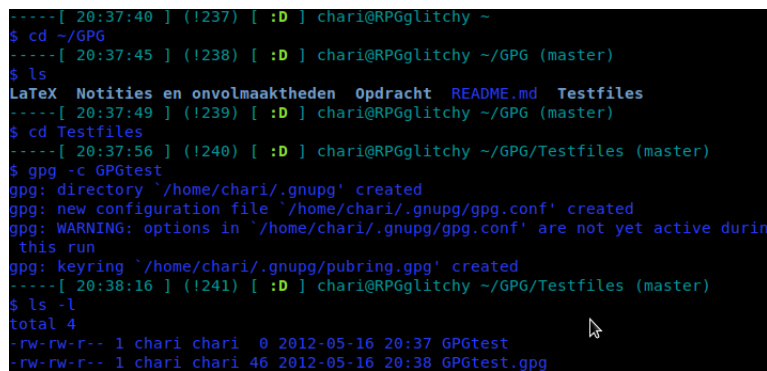
Bij gebruik van GPG is de werking als volgt:

```
cd /home/GPG
gpg -c test
```

Er wordt een passphrase gevraagd voordat er kan worden verdergegaan.

Dit is een wachtwoord dat bij eerste gebruik wordt aangemaakt.

Bij de weergave van de inhoud van deze Directory is er te zien dat het bestand 'GPGtest' geïncrypteerd is naar 'GPGtest.gpg'

A terminal window showing the execution of GPG commands. The user navigates to /home/GPG, lists files, then to /home/GPG/Testfiles. They run 'gpg -c GPGtest', which prompts for a passphrase and creates a keyring and configuration file. A warning message is shown. Finally, they run 'ls -l' showing the creation of 'GPGtest.gpg'.

```
-----[ 20:37:40 ] (1237) [ :D ] chari@RPGglitchy ~
$ cd ~/GPG
-----[ 20:37:45 ] (1238) [ :D ] chari@RPGglitchy ~/GPG (master)
$ ls
LaTeX  Notities en onvolmaaktheden  Opdracht  README.md  Testfiles
-----[ 20:37:49 ] (1239) [ :D ] chari@RPGglitchy ~/GPG (master)
$ cd Testfiles
-----[ 20:37:56 ] (1240) [ :D ] chari@RPGglitchy ~/GPG/Testfiles (master)
$ gpg -c GPGtest
gpg: directory `/home/chari/.gnupg' created
gpg: new configuration file `/home/chari/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/chari/.gnupg/gpg.conf' are not yet active during
this run
gpg: keyring `/home/chari/.gnupg/pubring.gpg' created
-----[ 20:38:16 ] (1241) [ :D ] chari@RPGglitchy ~/GPG/Testfiles (master)
$ ls -l
total 4
-rw-rw-r-- 1 chari chari  0 2012-05-16 20:37 GPGtest
-rw-rw-r-- 1 chari chari 46 2012-05-16 20:38 GPGtest.gpg
```

Figure 1: uitvoer in terminal

2.5 Frontend GPG programma's

Cryptophane	Een applicatie voor Windows.
Gajim	Een Jabber client voor GNOME.
GnuPG Shell	Een cross-platform, grafische Frontend voor GnuPG.
GPA	De standaard Frontend voor GPG.
KGpg	GnuPG voor KDE.
Seahorse	GnuPG voor GNOME.
Wija	Een cross-platform jabber client (MacOsX, Linux, Windows)

Table 4: Gui Frontends

2.6 Werking van een GPG Frontend

Een grafische applicatie voor het gebruik met GPG houdt heel vaak in dat een bestand geladen wordt en de gebruiker dan kan kiezen om te encrypteren of decrypteren.

Als de Frontend gebruik maakt van een messaging applicatie (Jabber-client), Houdt dit meestal in dat de gebruiker een handtekening instelt in de GUI, zodat deze gebruikt wordt bij het encrypteren van de boodschappen.

Als de GUI een sleutelhanger-functie heeft, zal er een instelling van een 'Passphrase' Voorzien zijn bij de initieele stappen. Deze passphrase zorgt ervoor dat de sleutelhanger ontgrendelt kan worden voor gebruik. Als er dan een wachtwoord wordt ingesteld op een website of applicatie, Zal er worden gevraagd deze op te slaan in je sleutelhanger. Als de gebruiker dit toestaat, wordt deze geencrypteerd toegevoegd aan je sleutelhanger.

Cryptophane Deze wordt gebruikt om te encrypteren, decrypteren, handtekenen, beheer van sleutelhangen en een command-line interface voor GnuPG.

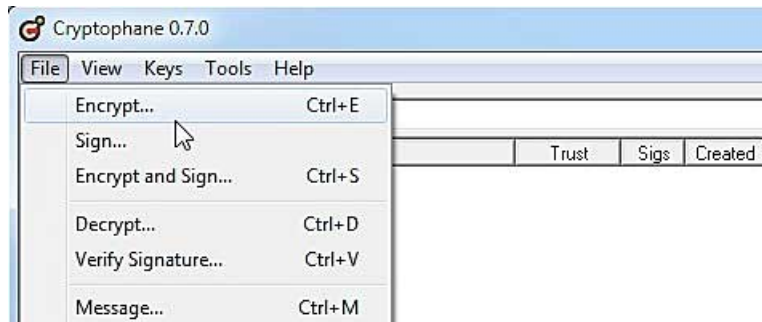


Figure 2: Cryptophane

Gajim Gajim is een Jabber-client. Een Jabber-client is een messaging applicatie. Omdat Gajim werkt met GnuPG, zullen de berichten die verzonden worden met Gajim, Geencrypteerd worden.

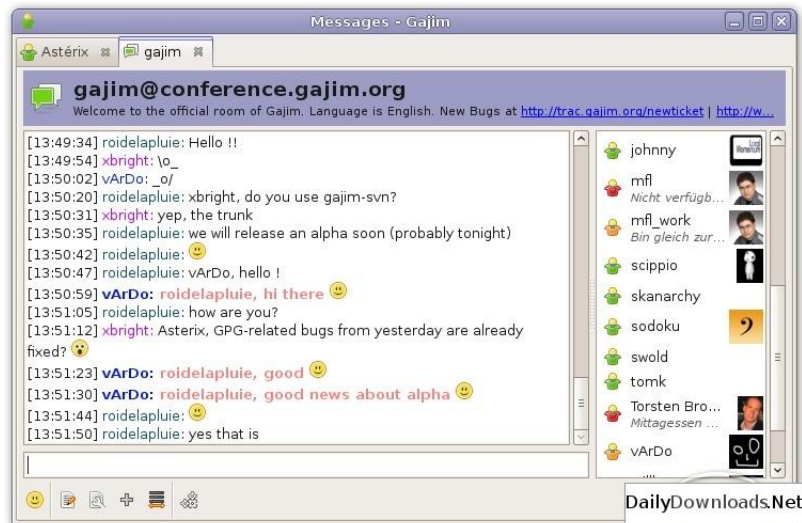


Figure 3: Gajim

GPGshell Een grafische frontend voor iedere platform. Met deze GUI is het mogelijk sleutels bij te houden en te encrypteren.

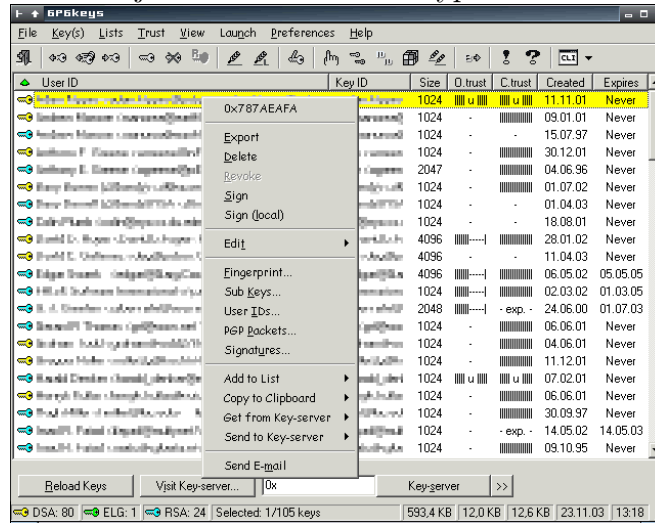


Figure 4: GnuPG Shell

GPA GPA probeert de standaard frontend te zijn voor GPG. www.gnupg.org Host GPA.

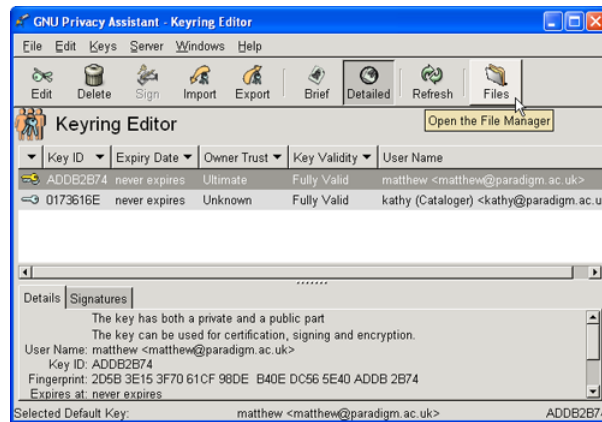


Figure 5: GPA

KGpg Met KGpg kan je bestanden en mails encrypteren en dycrepteren om je informatie veilig te houden. Het is een gratis en open-source frontend.

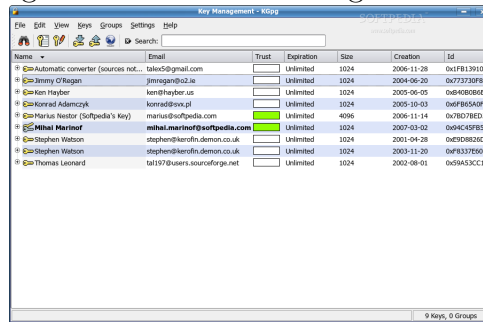


Figure 6: KGpg

Seahorse Seahorse is een GUI voor GNOME. Het is ook geïntegreerd in Nautilus, gedit en andere applicaties voor encryptie uit te voeren. Je kan met Seahorse; PGP en SSH sleutels maken en beheren, publiceren en terughalen van sleutels op de servers, een passphrase cachen, sleutelhanger backuppen, enz...

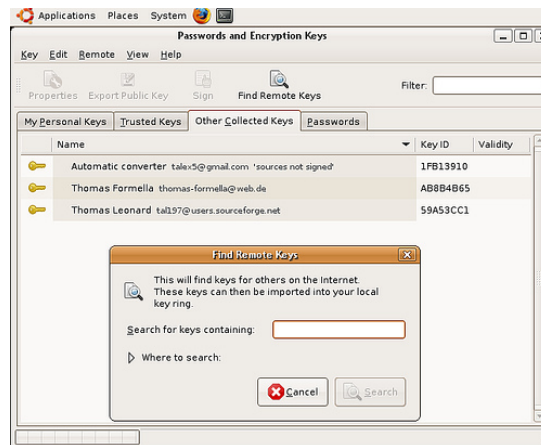


Figure 7: Seahorse

Wija Een Jabber-client zoals Gajim, maar geschreven in java en beschikbaar voor ieder platform. Het heeft een ingebouwde sleutelhanger beheersysteem. Het kan ook zeer gemakkelijk boodschappen encrypteren en decrypteren voor gewone gesprekken of multi-user gesprekken. Het is ook mogelijk de boodschappen te handtekenen.



Figure 8: Wija

3 Belangrijke woorden

Jabber-client		$h = here, t = top$
Cross-platform		
Multi-user		
Passphrase		
[!ht]		

Table 5: Belangrijke woorden

4 Referenties

http : //www.jumaros.de/rsoft/index.html

http : //www.gnupg.org/related_software/frontends.en.html

http : //www.google.be

http : //utils.kde.org/projects/kgpg/

http : //projects.gnome.org/seahorse/

*http : //tex.stackexchange.com/questions/8652/what - does - t - and -
ht - mean*

Geschreven door [?]