Gnu Privacy Guard (GnuPG) Mini Howto (Dutch)

Brenno J.S.A.A.F. de Winter (English) <brenno@dewinter.com> Michael Fischer v. Mollard (German) <fischer@math.uni-goettingen.de> Arjen Baart (Dutch) <arjen@andromeda.nl> Versie 0.1.3: 17 Mei, 2002

Dit document legt in het kort uit hoe je de GNU Privacy Guard (GnuPG) kunt gebruiken. GnuPG is een Open Source encryptie systeem, compatibel met OpenPGP. om het vrije gebruik van dit programma te waarborgen is het gebruik van het RSA en andere gepatenteerde algoritmes vermeden. Het originele document is geschreven door Michael Fischer v. Mollar in het Duits. De tekst is vertaald en op een aantal punten aangepast.

Contents

1 Concepten

Opmerking voor gedrukte versies: *Door een technische onvolkomenheid in verband met een dubbel streepje, zoals gebruikt in de lange opties van GnuPG, wordt dit in de lopende tekst als een enkel streepje weergegeven. Lees daarom bij lange opties altijd een* ***dubbel*** *streepje.*

1.1 Publieke Sleutel Encryptie

Klassieke encryptie methoden gebruiken slechts een (1) sleutel voor de encryptie. De verzender versleutelt de boodschap met deze ene sleutel. Om de boodschap weer te ontsleutelen heeft de ontvanger de zelfde sleutel nodig. De sleutel moet dan op een dusdanige wijze aan de ontvanger zijn gegeven dat anderen niet de gelegenheid hebben gehad om de sleutel te pakken te krijgen. Als een derde partij ook de sleutel heeft, is deze methode van encryptie onbruikbaar.

Dit probleem wordt opgelost door het gebruik van Publieke Sleutels. In het Publieke Sleutel concept worden twee sleutels gebruikt. Een sleutel is de Publieke Sleutel die op allerlei manieren verspreid wordt en die iedereen in zijn bezit mag hebben. De andere sleutel is de Prive Sleutel. Deze sleutel is geheim en mag niet verspreid worden. Alleen de eigenaar is in het bezit van de Prive Sleutel. In een goed ontworpen encryptie systeem kan de geheime sleutel niet van de publieke sleutel afgeleid worden. De verzender versleutelt nu de boodschap met de publieke sleutel van de ontvanger. Het ontsleutelen is dan alleen mogelijk met de gehieme sleutel van de ontvanger.

Cruciaal in dit concept is dat de geheime sleutel ook werkelijk geheim blijft en niet in het bezit komt van iemand anders dan de eigenaar van deze sleutel. JE MAG DEZE SLEUTEL NOOIT OVER HET INTERNET VERSTUREN. Het is dan ook niet aan te bevelen om GnuPG met een `telnet` verbinding te gebruiken.

1.2 Digitale Handtekeningen

Om te bewijzen dat een boodschap daadwerkelijk is verzonden door degene die beweert de boodschap verzonden te hebben is het concept van de Digital Handtekening uitgevonden. Zoals de naam suggereert, is een boodschap door de verzender ondertekend met een digitale handtekening. Met deze handtekening kun je de autenticiteit van de boodschap controleren. Hiermee is het risico voor Trojaanse paarden (een boodschap die beweert een oplossing voor een bepaald probleem te beiden maar in werkelijkheid een virus bevat of anderszins schade aanricht) te reduceren. Ook kan geverifieerd worden of informatie van een legitieme bron vandaan komt en als waarheid aangenomen kan worden.

Een digitale handtekening wordt gemaakt door een combinatie van de geheime sleutel en de tekst. De boodschap kan geverifieerd worden met de publieke sleutel van de verzender. Niet alleen wordt gecontroleerd of de verzender correct is, ook de inhoud van de boodschap wordt gecontroleerd. Je bent dan niet alleen zeker van de verzender, maar je ziet ook of de informatie onderweg niet veranderd is.

1.3 Web van vertrouwen

Een zwak punt in de Publieke Sleutel methoden is de verspreiding van de publieke sleutels. Iemand zou een publieke sleutel met een valse gebruiker identificatie in omloop kunnen brengen. Als een boodschap met deze sleutel gemaakt wordt, dan kan de indringer die boodschap decoderen en lezen. Als de indringer vervolgens de boodschap doorgeeft, versleuteld met de echte publieke sleutel van de bedoelde ontvanger, is deze aanval redelijk onopvallend.

De oplossing in PGP (en daarmee automatisch in GnuPG), bestaat uit het signeren van sleutels. Een publieke sleutel kan gesigneerd worden door andere personen. Een dergelijke handtekening bevestigt dat de sleutel, gebruikt door de UID (Gebruiker Identifikatie) inderdaad toebehoort aan de persoon waar deze beweert aan toe te behoren. Het is dan aan de gebruiker van GnuPG om aan te geven hoever het vertrouwen in die handtekening gaat. Je kunt een sleutel als vertrouwenswaardig beschouwen wanneer je de verzender van de sleutel vertrouwt en zeker weet dat de sleutel inderdaad bij die persoon hoort. Alleen wanneer je de sleutel van de persoon die de sleutel gesigneerd heeft vertrouwt, kun je de handtekening vertrouwen. Om absolute zekerheid te hebben dat een sleutel correct is, moet je de 'fingerprint' via een betrouwbare weg controleren voordat je een sleutel als vertrouwenswaardig beschouwt.

1.4 Grenzen aan beveiliging

Wanneer je informatie vertrouwelijk wilt houden is er meer nodig dan alleen een versleutel algoritme te kiezen. Je moet dan ook nadenken over de systeem beveiliging in het algemeen. Over het algemeen is PGP als veilig te beschouwen en op dit moment zijn er geen incidenten bekend van het kraken van PGP. Dat betekent echter niet dat alle versleuteling absoluut veilig is (de NSA, bij voorbeeld, zal het niet aan de grote klok hangen als ze op een of andere manier PGP gekraakt hebben). Maar ook al zou PGP volledig onkraakbaar zijn, dan zijn er nog andere methoden denkbaar om de beveiliging te doorbreken. In februari 1999 is er een Trojaans paard gevonden wat zocht naar geheime PGP sleutels op de harddisk en ze met FTP weg stuurde. Als de wachtwoorden onhandig gekozen zijn, kan een geheime sleutel vrij eenvoudig achterhaald worden.

Een andere technische mogelijkheid is een Trojaans paard dat toetsaanslagen doorstuurt. Ook is het mogelijk, alhoewel zeer moeilijk, om de inhoud van het beeldscherm door te geven aan iemand die afluistert. Het is dan niet eens nodig om gecodeerde boodschappen te kraken. Tegen al deze risico's is het nodig om een goed beveiligingsplan te hebben en ook uit te voeren.

Het is niet de bedoeling om onrust te zaaien, maar wel om duidelijk te maken dat er veel moet gebeuren om een goede beveiliging van gegevens te realizeren. Het is belangrijk om op te merken dat encryptie slechts een enkele stap is om de beveiliging te verhogen maar niet de totale oplossing kan vormen. Trojaanse paarden zoals die verschenen met het Melissa virus in maart 1999 bewezen dat veel bedrijven daar niet op voorbereid zijn.

2 Installatie

2.1 Bronnen voor GnuPG.

De officiele download site is:

GnuPG Homepage `http://www.gnupg.org/download.html` . Daar zijn ook links naar mirror sites te vinden.

Vanwege juridische beperkingen is het niet teogestaan om GnuPG te downloaden van servers in de USA. De USA legt export beperkingen op aan cryptografische software. Daarom is PGP altijd beschikbaar in een internationale en een nationale (voor de USA) versie. Voor de internationale versie van de software is de source code geexporteerd in afgedrukte vorm, als een boek. Deze versie is in Europa (Oslo) gescand. Meer informatie hierover is te vinden op

Internationale PGP Homepage `http://www.pgpi.com` . De internationale versie mag wel in de USA geimporteerd worden maar mag daarna niet meer geexporteerd worden.

Als je al een versie van PGP of GnuPG geinstalleerd hebt, doe je er natuurlijk verstandig aan de handtekening van de file te controleren (zie 5 (Handtekeningen)).

2.2 Configuratie

GnuPG is verkrijgbaar als een Debian pakket, een RPM pakket (Redhat Package Manager) of in source code. Met de nieuwste Redhat distributies wordt GnuPG meegeleverd. Gebruik dit commando om te controleren of GnuPG op je systeem is geinstalleerd:

```
rpm -q gnupg
```

De paketten zijn geinstalleerd als binaire files voor Linux. Wanneer je GnuPG voor een ander platform nodig hebt, zul je het zelf moeten compileren. Het wordt dan natuurlijk gewaardeerd als je installatie methoden voor een ander platform aan het publiek beschikbaar stelt.

Aangezien de ontwikkeling voor het leeuwendeel op Linux (x86) plaats vindt, is de vertaling naar andere systemen vaak wat problematisch. De actuele lijst van operating systems die GnuPG ondersteunen is te vinden op de *GnuPG Homepage* . Hieronder is een methode beschreven om GnuPG vanuit een source tar-ball te installeren. Deze methode is redelijk platform onafhankelijk.

Pak de tar-ball uit met het commando:

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

Ga na het uitpakken naar de directory waar de source code in staat. Type dan:

```
./configure
```

Hierbij zou niets bijzonders moeten gebeuren. Met

```
./configure --help
```

kun je de beschikbare opties voor het compileren bekijken. Als er problemen optreden die betrekking hebben op de internationalisatie (GET text), kun je een versie gebruiken die met de source code meegeleverd wordt door middel van de `-with-included-gettext` optie of dit uitschakelen door de `-disable-NLS` optie te gebruiken.

2.3 Compileren

Hierna kunnen we de boel compileren door

```
make
```

in te typen. Dit zou zonder problemen moeten werken. Mochten zich toch problemen voordoen, onderneem dan de volgende stappen (in de aangegeven volgorde): Probeer eerst zelf het probleem op te lossen (uiteraard door de beschikbare documentatie te raadplegen). Onderzoek dan of je probleem niet een bekende bug is (lees de BUGS file op http://www.gnupg.org). Vraag het vervolgens aan een bekende. Als je met deze stappen het probleem niet opgelost krijgt, stel dan een vraag op de GnuPG mailing list (zie 7 (Informatiebronnen)). Als het probleem te maken heeft met directory paden, probeer dan een `make clean`, daarna `configure` en probeer opnieuw te compileren. Werkt dat ook niet, dan is het tijd om paniek te zaaien.

2.4 Installatie

Type nu:

```
make install
```

Om het programma en de man-pages naar de installatie directory te copieren. Als je de installatie directory bij ./configure niet veranderd hebt is `/usr/local/share/gnupg/` de directory waar GnuPG geinstalleerd wordt. Je kunt deze directory vinden in de file `options.skel`.

Je kunt GnuPG als suid root draaien, zodat het programma draait met alle rechten van de superuser. Hiermee sluit je de mogelijkheid uit dat bepaalde delen van het programma extern opgeslagen worden en door iedereen gelezen kunnen worden. Het is mij echter niet mogelijk om dit risiko in te schatten. Echter, als je het programma met suid root draait moet je alert blijven op het gevaar van Trojaanse paarden. Een Trojaans paard wat met de rechten van de superuser draait kan in het gehele systeem schade aanrichten. Als je om deze of een andere reden er voor kiest om GnuPG niet als root te draaien, kun je de waarschuwing onderdrukken door de instelling `no-secmem-warning` in `~/.gnupg/options` te zetten.

3 Sleutels gebruiken

3.1 Een Sleutel aanmaken

Met

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

Wordt een nieuw sleutelpaar gemaakt (sleutelpaar: geheime en publieke sleutel). De eerste vraag is welk alogritme gebruikt wordt. Je kunt meer lezen over de algoritmes in

PGP DH vs. RSA FAQ `http://www.hertreg.ac.uk/ss/pgpfaq.html` of in 7.4 (Applied Cryptography). Je kunt gemakkelijk (en eigenlijk zou je dat ook moeten doen) DSA/ ElGamal gebruiken. Deze is niet gepatenteerd.

De volgende vraag is de lengte van de sleutel. Dit is afhankelijk van je eigen wensen. Je moet een keuze maken tussen veiligheid en rekentijd. Wanneer de sleutel langer is, is het risico dat de boodschap gekraakt wordt minder groot. Echter, een langere sleutel vergt ook meer rekentijd. Als rekentijd een punt is moet je nog steeds in gedachten houden dat je de sleutel een langere tijd gaat gebruiken. We weten allemaal dat de snelheid van computers toeneemt omdat processors steeds sneller en sneller worden. Houdt dit dus in gedachten. De minimale sleutellengte die GnuPG gebruikt is 768 bits. Hoewel, veel mensen vinden dat je eigenlijk een sleutel lengte van 2048 bits zou moeten gebruiken, het maximum wat GnuPG momenteeel aan kan. Voor DSA is 1024 bits de standaard lengte. Wanneer de veiligheid de hoogste prioriteit heeft en rekentijd minder zwaar weegt moet je natuurlijk de grootst mogelijke sleutellengte kiezen.

Het systeem vraagt nu om je naam, een omschrijving en je e-mail adres in te voeren. De code wordt op basis van deze gegevens berekend. Naderhand kun je deze items nog wijzigen. Zie 3.5 (Sleutels Administreren).

Tenslotte moet je een wachtwoord invoeren (eigenlijk is het meer een 'wachtzin', aangezien er ook spaties in voor mogen komen). Dit wachtwoord wordt gebruikt om de funkties die bij je geheime sleutel horen te kunnen gebruiken. Een goed wachtwoord voldoet aan het volgende:

- het is lang,
- het bevat speciale (niet alfanumerieke) tekens,
- het is iets speciaals (niet een naam),
- het is moeilijk te raden (dus GEEN namen, geboortedatums, telefoon nummers, bankrekening nummers, namen van kinderen en dergelijke)

Door soms HoOfdlETters te gebruiken en sOMs NieT kun je nog meer veiligheid inbouwen. Wanneer je je wachtwoord bedenkt, zorg er voor dat je het NOOIT MEER VERGEET, aangezien je boodschappen dan onleesbaar zijn en je geheime sleutel dus onbruikbaar is. Het kan verstandig zijn om een soort certificaat met deze informatie te maken (uiteraard wel voorzichtig zijn dat niemand je wachtwoord in handen krijgt). Zie 3.4 (Intrekken).

Als alles ingevoerd is begint het systeem de sleutels te genereren. Dit kan een tijdje duren. Gedurende die tijd wordt er veel toevallige (random) informatie verzameld. Door in de tussentijd op een ander scherm en met andere programma's door te werken help je het systeem goede random informatie te verzamelen. Je zult begrijpen dat er nooit de zelfde sleutel twee keer gegenereerd wordt. Iedere keer is de sleutel weer anders. Wanneer je nu een sleutel aanmaakt, en over 5 minuten weer met exact dezelfde invoer, krijg je twee verschillende sleutels. Nu begrijp je dus ook waarom je je wachtwoord niet moet vergeten.

3.2 Sleutels Exporteren

Het commando om de sleutel van een gebruiker te exporteren is:

```
gpg --export [UID]
```

Als er geen UID aangegeven is worden alle aanwezige sleutels geexporteerd. Normaliter gaat de uitvoer naar `stdout`. Met de `-o` optie is dit naar een file te sturen. Het kan aan te raden zijn de `-a` optie te gebruiken om de sleutel in 7-bits ASCII te schrijven inplaats van in een binaire file.

Door publieke sleutels te exporteren verbreedt je je horizon. Anderen kunnen je dan op een beveiligde manier benaderen. Je kunt dit doen door je publieke sleutel op je homepage te publiceren, met `finger`, met een sleutel server zoals http://www.pca.dfn.de/dfnpca/pgpkserv/ of iedere andere methode die je kunt bedenken.

3.3 Sleutels Importeren

Wanneer je iemands publieke sleutel (of diverse publieke sleutels) hebt ontvangen, moet je ze aan je sleutel database toevoegen om ze te kunnen gebruiken. Het commando om een sleutel aan je database toe te voegen ziet er zo uit:

```
gpg --import [Filename]
```

Als je de Filename er niet bij zet wordt de informatie van `stdin` gelezen.

3.4 Sleutels Intrekken

Er zijn verscheidene redenen denkbaar waarom je een bestaande sleutel zou willen intrekken. De gehieme sleutel kan bij voorbeeld gestolen zijn of in verkeerde handen gevallen, the UID is veranderd of de sleutel is niet lang genoeg meer, etc. In al deze gevallen is het commando om de sleutel in te trekken:

```
gpg --gen-revoke
```

Dit creeert een intrek-certificaat. *Om dit te doen heb je een gehieme sleutel nodig*, anders zou iedereen je certificaat kunnen intrekken. Dit heeft een nadeel. Als ik het wachtwoord niet ken is de sleutel onbruikbaar. Maar ik kan de sleutel niet intrekken ! Om dit probleem te voorkomen is het wijs een intrek-certificaat te maken wanneer je een sleutelpaar maakt. Als je dat doet, bewaar het op een veilige plaats ! Zorg er wel voor dat dit certificaat niet in verkeerde handen valt !! Je wilt natuurlijk voorkomen dat iemand anders het intrek-certificaat op jouw sleutel toepast en de sleutel onbruikbaar maakt.

3.5 Sleutels Administreren

Bij het GnuPG systeem hoort een file die dienst doet als een soort database. In deze file wordt allerlei informatie over de sleutels opgeslagen. Onder andere de *Ownertrust* waarden. Voor meer informatie hierover, lees 3.6 (Sleutels signeren)). Met

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

worden alle aanwezige sleutels getoond. Om ook de signaturen te zien, type:

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

(zie 3.6 (Sleutels signeren) voor nadere informatie). Om de vingerafdrukken te zien, type:

```
gpg --fingerprint
```

Met vingerafdrukken ("Fingerprints") kun je je er van verzekeren dat iemand werkelijk de persoon is die hij of zij beweert te zijn (bij voorbeeld met een telefoongesprek). Dit commando geeft een lijst van kleine getallen.

Om de geheime sleutels te laten zien type je:

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

Merk op dat het tonen van vingerafdrukken en signaturen van prive sleutels geen enkel nut heeft.

Om een publieke sleutel te verwijderen type je:

```
gpg --delete-key UID
```

Om een geheieme sleutel te verwijderen type je:

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

Er is nog een belangrijk commando om met sleutels om te gaan.

```
gpg --edit-key UID
```

Hiermee kun je onder andere de afloopdatum aanpassen, een vingerafdruk toevoegen en een sleutel signeren.

3.6 Sleutels signeren

Zoals reeds vermeld in de inleiding, er is een significante Achilleshiel in het systeem. Dat is de authenticiteit van publieke sleutels. Als je een verkeerde publieke sleutel hebt, kun je wel dag zeggen tegen de waarde van de versleuteling. Om dit risico het hoofd te bieden is er de mogelijkheid om sleutels te signeren. Je plaatst dan je handtekening over de sleutel, zodat je absoluut zeker weet dat het de juiste sleutel is. Dit geeft de situatie waarin de handtekening bevestigt dan de *user ID* die in de sleutel genoemd is daadwerkelijk de eigenaar van de sleutel is. Met deze zekerheid kun je beginnen met encrypten.

Gebruik makend van het `gpg -edit-key UID` commando voor de sleutel die gesigneerd moet worden, kun je de sleutel signeren met het `sign` commando.

Je moet alleen een sleutel signeren als zijnde authentiek wanneer je ABSOLUUT ZEKER bent van de authenticiteit van de sleutel !!!!!. Dus als je zeker weet dat je de sleutel persoonlijk overhandigd hebt gekregen (zoals op een *key signing party*) of je hebt de sleutel op een andere manier verkregen en de sleutel hebt gecontroleerd (bij voorbeeld telefonisch) met behulp van de vingerafdruk. Je moet nooit een sleutel signeren gebaseerd op welke aanname dan ook.

Gebaseerd om de beschikbare signaturen en "ownertrusts" bepaalt GnuPG de validiteit van sleutels. Ownertrust is een waarde waarmee je aangeeft tot op welk niveau je een bepaalde sleutel vertrouwt. De mogelijke waarden zijn

- 1 = Onbekend
- 2 = Geen vertrouwen
- 3 = Marginaal vertrouwen
- 4 = Volledig vertrouwen

Als je de signatuur niet vertrouwt kun je dat aangeven en daarmee de signatuur negeren. Vertrouwensinformatie wordt niet in dezelfde file opgeslagen als de sleutels, maar in een aparte file.

4 Encrypt en decrypt

Nadat alles geinstalleerd is en geconfigureerd zoals we willen, kunne we beginnen met encrypten en decrypten.

Tijdens het encrypten en decrypten is het mogelijk om meer dan een prive sleutel in gebruik te hebben. In dat geval moet je de aktieve sleutel kiezen. Dit kan door de optie `-u UID` of de optie `-local-user UID` te gebruiken. Dit zorgt er voor dat de default sleutel vervangen wordt door de aangegeven sleutel.

Als je de ontvanger wilt veranderen kan dat door de optie `-r` of de optie `-recipient` te gebruiken.

4.1 Encrypten

Het commando om te encrypten is

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

of

```
gpg --encrypt Recipient [Data]
```

Om het risico te voorkomen dat iemand anders beweert jou te zijn, is het handig om alles wat je encrypt ook te signeren, zie 5 (handtekeningen).

4.2 Decrypten

Het commando voor decrypten is:

```
gpg [-d] [Data]
```

of

```
gpg [--decrypt] [Data]
```

Ook hier wordt `stdout` gebruikt, maar met de `-o` optie kun je de uitvoer naar een file redigeren.

5 Signeren en handtekeningen controleren

Om data te signeren met je eigen handtekening, gebruik je het commando:

```
gpg -s (or --sign) [Data]
```

Hierbij vindt ook een compressie plaats. Dat houdt in dat het resultaat niet leesbaar is. Als je een leesbaar resultaat wilt, gebruik je:

```
gpg --clearsign [Data]
```

Dit geeft een leesbaar resultaat. Verder doet dit hetzelfde (data signeren).

Met

```
gpg -b (or --detach-sign) [Data]
```

kun je een handtekeing in een aparte file maken. Het is aan te bevelen deze optie te gebruiken bij het signeren van binaire files, zoals bij voorbeeld archieven. Ook de `-armot` optie kan hier bijzonder bruikbaar zijn.

Vrij vaak zie je dat data zowel gecrypt als gesigneerd is. De volledige instruktie ziet er als volgt uit:

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

De funktie van de optie `-u` (`-local-user`) en `-r` (`-recipient`) werken zoals hierboven beschreven.

Wanneer gecrypte data ook gesigneerd is, wordt de handtekening automatisch gecontroleerd wanneer de data gedecrypt wordt. Je kunt de handtekening van gesigneerde data controleren met het commando:

```
gpg [--verify] [Data]
```

Dit werkt natuurlijk alleen als je de publieke sleutel van de verzender hebt.

6 Front ends

Om het leven met betrekking tot GnuPG een stuk gemakkelijker te maken is er een ruime keuze aan programma's die GnuPG encryptie gebruiken of ondersteunen. Er zijn grafische front ends waarmee je niet veel meer nodig hebt dan een klik op de muis voor het beheer van je sleutels en veel Email programma's (MUAs - Mail User Agents) waarmee je naadloos de Email kunt encrypten en signeren. Een vrijwel complete lijst van front ends is te zien op de *GnuPG Frontends* pagina. We laten er in dit hoofdstuk een aantal de revue passeren.

6.1 Grafische interfaces

6.1.1 GPA

GPA , de *GNU Privacy Assistant* is een grafische gebruiker interface voor de GNU Privacy Guard (GnuPG). Dit is het standaard grafische front end, gemaakt binnen het GnuPG project. Met GPA kun je je sleutelbos bekijken, sleutels importeren en exporteren, sleutels aanmaken, de eigenschappen van sleutels aanpassen en documenten encrypten, signeren en decrypten. Installeren van GPA is eenvoudig. De tarball downloaden, uitpakken en dan de gebruikelijke

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

Start het programma door

```
gpa
```

in te typen.

6.1.2 Seahorse

Seahorse

is een GNOME front end voor GnuPG. Het is te gebruiken voor signeren, encrypten, verifieren en decrypten van tekst en binaire files. De tekst kun je uit het clipboard halen of direkt schrijven in de ingebouwde editor. Seahorse is ook een sleutelbeheerder waarmee je vrijwel alle eigenschappen kan bewerken van de sleutels in je sleutelbos. Je kunt Seahorse installeren vanuit een Debian pakket (RPMs zijn momenteel niet beschikbaar) of vanuit een source tarball. Installeren vanaf source werkt net als bij ieder ander pakket. Downloaden, uitpakken, configure en make install. De installatie zet Seahorse in `/usr/local` en maakt een menu item is het Gnome 'Toepassingen' menu.

6.1.3 Geheimnis

Geheimnis is een front end voor GnuPG die gebaseerd is op KDE. Het zou alle funkties moeten hebben die GPA of Seahorse ook hebben. Echter, toen ik het programma probeerde te installeren, liep ik vast op een bibliotheek die niet aanwezig bleek te zijn (/usr/lib/libfam.la).

6.2 Email programma's

De meeste populaire Email programma's (of MUAs) ondersteunen GnuPG. Hierbij zitten minimaal de volgende:

- Mozilla

- Pine
- Kmail
- Eudora
- Mutt
- exmh

Er zijn er waarschijnlijk meer; het is nauwelijks mogelijk om ze allemaal uit te proberen.

Door GnuPG ondersteuning in je mail programma te gebruiken kun je boodschappen die met je publieke sleutel gecrypt zijn decrypten, boodschappen signeren zodat de ontvangende partij er zeker van kan zijn dat jij de auteur bent en je Email encrypten met de publieke sleutels van de geadresseerden.

6.2.1 Mozilla en Enigmail

Mozilla heeft van zichzelf geen understeuing voor GnuPG. Om GnuPG encryptie in Mozilla te gebruiken moet je een plugin installeren, zoals *EnigMail* . Enigmail is een "plugin" voor Mozilla/Netscape Mail waarmee gebruikers de mogelijkheid hebben om de authenticatie en encryptie faciliteiten van GPG en PGP software te gebruiken. Enigmail kan verstuurde mail encrypten en signeren, en ook ontvangen mail decrypten en controleren. Het kan ook publieke sleutels importeren en exporteren.

Installeren van EnigMail op een RedHat Linux systeem is bijzonder gemakkelijk, aangezien hier RPMs voor beschikbaar zijn. Er zijn twee paketten nodig die beide geinstalleerd moeten worden: `mozilla-enigmail-0.39-3.i386.rpm` en `mozilla-ipc-0.99-0_rh7x.rpm`. Start Mozilla (of Netscape 6.x) na het installeren van deze RPMs opnieuw en je kunt GnuPG in je Email gebruiken. Merk echter op dat EnigMail alleen de tekst van je boodschap versleutelt. Bijgevoegde files (attachments) worden door EnigMail niet gecrypt; files die je wilt bijvoegen moet je apart versleutelen. Op andere systemen moet je EnigMail waarschijnlijk vanaf source code installeren.

6.2.2 Kmail

Kmail, het standaard Email programma voor KDE heeft geintegreerde ondersteuning voor GnuPG en PGP encryptie. Om de zaak op te zetten zodat je boodschappen kunt signeren en decrypten moet je je GnuPG user ID invullen in de 'Identity' sectie van de Kmail configuratie. Wanneer je een nieuwe boodschap verstuurt, wordt de boodschap normaliter niet gesigneerd of versleuteld. Je moet hiervoor de 'Sign message' en 'Encrypt message' knoppen indrukken.

7 Bronnen van informatie

7.1 GnuPG

- De *GnuPG Homepage* `http://www.gnupg.org`
- De GnuPG Mailing list, inclusief archieven en beschrijvingen, op de *GnuPG Homepage* .
- De informatie die bij het GnuPG project is bijgesloten (aangepast tot versie 0.9.2), hoewel niet uitputtend. En niet te vergeten:

  ```
  gpg --help
  ```

  Dit is zeer waardevolle informatie.

7.2 PGP

PGP is het oudere en nog steed wijd verspreide cryptografische programma. Door de jaren heen is hier veel over geschreven. Dit kan beschouwd worden als een zeer waardevolle bron van informatie. Veel van deze informatie is zo algemeen dat het ook voor GnuPG toegepast kan worden. Kijk eens naar de volgende URLs voor deze documenten:

- De *Internationale PGP Homepage* `http://www.pgpi.com`
- De *PGP DH vs. RSA FAQ* `http://www.hertreg.ac.uk/ss/pgpfaq.html` bevat informatie over de verschillen tussen deze twee algoritmen. Dit zijn de algoritmen die door GnuPG gebruikt worden.

7.3 Keyservers

- *Keyserver.net* `http://www.keyserver.net`
- `http://wwwkeys.eu.pgp.net`

7.4 Boeken

- B. Schneier, "Applied Cryptography, Second Edition", Wiley, 1996 Deutsche Ausgabe unter dem Titel "Angewandte Kryptographie", Addison-Wesley, 1996

8 Over dit document

Copyright © 1999 Brenno J.S.A.A.F. de Winter (Engelse versie) Copyright © 1999 Michael Fischer v. Mollard (originele Duitse versie) Copyright © 2002 Arjen Baart (Nederlandse versie)

Dit document is vrije documentatie en je kunt het herdistribueren en/of aanpassen onder de voorwaarden van de GNU Library General Public License als gepubliceerd door de Free Software Foundation; versie 2 van deze License, of een latere versie.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

8.1 Versies

Originele Duitse versie: Versie 0.1 was de eerste versie in Duits

Versie 0.1.0 (Engels) 30 April 1999, Koninginnedag.

- Deze versie is de vertaling van de Duitse versie in het Engels met een paar aanpassingen.

Versie 0.1.1 (Duits)

- Nieuwe paragraaf "Grenzen aan beveiliging"
- Uitleg van handtekeningen verbeterd

- Wijzigingen na commentaar van Werner Koch (bedankt!)

Versie 0.1.2 (Engels) 3 April, 2002

- Een aantal typefouten gecorrigeerd
- Nieuw hoofdstuk over front ends

Versie 0.1.3 (Nederlands) 17 Mei, 2002

- Deze versie is een vertaling van de Engelse versie in het Nederlands.

Alle wijzigingen zijn gedocumenteerd in een diff file:

dieses Dokument `http://www.stud.uni-goettingen.de/~s070674/GnuPGMiniHowto/`

Voor de Engelse of Nederlandse versie: Alle opmerkingen over dit document kunnen gestuurd worden naar Brenno J.S.A.A.F. de Winter (*brenno@dewinter.com*). of Arjen Baart (*arjen@andromeda.nl*). Je opmerkingen helpen ons een beter document te schrijven en worden zeer gewaardeerd.

Voor de Duitse versie: Anregungen, Kritik, Verbesserungen und Erweiterungen einfach an Michael Fischer v. Mollard (*fischer@math.uni-goettingen.de*) senden, damit dieses Dokument weiter verbessert werden kann.