Pieter Meulemeester

PGP

===========================

1. Maak voor jezelf minstens drie sleutelparen aan

(DSA+ElGamal, DSA en RSA) met een verschillend userID;

hiervoor gebruik je best iets in de aard van "voornaam

familienaam". Kies een verschillende geldigheidsperiode

voor deze sleutels.

Vraag 1: waartoe dient de passphrase?

Vraag 2: waarom wordt er bij het aanmaken van een sleutel

soms naar extra toetsaanslagen gevraagd en soms ook niet?

==========================================================

vraag 1: deze dient om de private sleutel te beschermen ->

wordt gebruikt om de private sleutel te encrypteren

vraag 2: Om ervoor te zorgen dat de random generator genoeg

entropie zou hebben. op die manier zijn de random getallen

ao willekeurig mogelijk.

sleutelparen aanmaken:

commando: gpg --gen-key

->toont keuzemenu algoritmen

1. DSA+ElGamal

---------------

-bij keuze algoritme: kiezen voor 2) DSA and Elgamal

-keysize kiezen - enter

-geldigheidsduur, bvb 5 dagen -> 5 ingeven

-keuze bevestigen -> y ingeven

-ID instellen

-echte naam: Tim Dheedene1

-email: timdhee1@hotmail.com

-eventueel commentaar ingeven

- keuze van ID bevestigen -> o ingeven

-passphrase ingeven

-> boodschap op het scherm:

we need to generate a lot of random bytes. it is a good

idea to perform some other action (type keyboard, move

mouse, utilize the disks) during the prime generation this

gives the random number generator a better chance to gain

enough entropy.

2. DSA

-------

-bij keuze algoritme: kiezen voor 3) DSA (sign only)

-keysize kiezen - enter

-geldigheidsduur, bvb 1w -> 1 week

-keuze bevestigen -> y ingeven

-ID instellen

-echte naam: Tim Dheedene2

-email: timdhee2@hotmail.com

-eventueel commentaar ingeven

- keuze van ID bevestigen -> o ingeven

-passphrase ingeven

3. RSA

-------

-bij keuze algoritme: kiezen voor 4) RSA (sign only)

-keysize kiezen - enter

-geldigheidsduur, bvb 1m -> 1 maand

-keuze bevestigen -> y ingeven

-ID instellen

-echte naam: Tim Dheedene2

-email: timdhee2@hotmail.com

-eventueel commentaar ingeven

- keuze van ID bevestigen -> o ingeven

-passphrase ingeven

controle: overzicht alle keys:

commando: gpg --list-keys

-> keys worden op scherm getoond met extra info

2. Exporteer al je publieke sleutels naar meerdere bestand-

en die je aan andere gebruikers kan aanbieden.

Exporteer ze in 2 verschillende formaten, waarvan er één

geschikt is voor transport via e-mail of publicatie op een

webserver.

Vraag: wat is het formaat van deze bestanden?

==========================================================

exporteren:

gpg --export -o exportkeys

dit bestand is een binair bestand.

door cat exportkeys te doen, komen vreemde tekens te

voorschijn.

gpg --export -a > exportasci.txt

dit bestand is in ASCII (ASCII armored output).

de opdracht cat xportasci.txt toont leesbare tekens op het

scherm.

3. Wissel dergelijke bestanden uit met minstens 4 andere

gebruikers. Dit kan via publieke directory, usbstick,

e-mail , ...

We gaan er op dit ogenblik van uit dat sleutels die op

deze manier zijn verkregen, volledig te vertrouwen zijn.

===========================================================

Ik heb thuis een paar sleutels uitgewisseld tussen 2

gebruikers op 2 virtuele machines. op beide 'toestellen'

werd opgave 1 uitgevoerd (met andere waarden).

4. Voeg deze verkregen publieke sleutels toe aan je

publieke sleutelhanger.

Verifieer of ze wel zijn opgenomen.

Verwijder een van de sleutels uit de sleutelhanger.

Controleer. Voeg hem er nadien weer bij.

===========================================================

importeren: gpg --import bestandsnaam

-> de sleutels van de gebruiker op de andere virtuele

machine worden opgenomen in de keychain.

De opdracht toont het aantal geimporteerde sleutels.

dit aantal klopt met het aantal sleutels dat er waren in

het bestand.

controleren of ze opgenomen zijn:

gpg --list-keys

-> deze opdracht toont het overzicht van alle aanwezige

sleutels;

de 3 die al aanwezig waren en de 3 die werden

geïmporteerd

1 sleutel verwijderen:

bij het importeren wordt er een ID getoond:

gpg: key XXXXXXXX

deze ID wordt ook getoond bij het opvragen van een

overzicht van alle aanwezige sleutels:

pub YYYYY/XXXXXXXX datum

(XXXXXXXX is de ID)

om te verwijderen:

gpg --delete-key XXXXXXXX

-> de opdracht vraagt de bevestiging van de gebruiker

om de key van de keyring te verwijderen

-> controle:

de opdracht gpg --list-keys geeft teug de aanwezige

sleutels weer, de verwijderde sleutel is wordt niet

getoond, en is dus verwijderd -> test geslaagd

terug toevoegen:

gewoon terug importeren, dan wordt de sleutel terug

toegevoegd. De opdracht vermeldt het aantal verwerkte

sleutels, het aantal geimporteerde sleutels en het

aantal sleutels die onveranderd zijn gebleven. Er is te

zien dat er 1 sleutel werd geimporteerd, namelijk

de verwijderde sleutel. terug een overzicht opvragen met

gpg --list-keys toont dat de verwijderde sleutel

inderdaad terug werd toegevoegd. test geslaagd

5. Editeer een van je sleutels: verander de passeerzin.

=========================================================

het commando om dit te doen:

gpg --edit-key XXXXXXXX

waarbij XXXXXXXX dezelfde betekenis heeft als bij de vorige

vraag.

deze opdracht geeft een secundaire prompt weer. Met 'help'

wordt een overzicht getoond van de mogelijke opdrachten.

--> passwd: change passphrase

we moeten de huidige passphrase ingeven, daarna kunnen we

een nieuwe passphrase ingeven.

om de secundaire prompt te verlaten: quit opdracht

op einde 'save changes?' -> y ingeven

6. Check je sleutels en de handtekeningen die er bij horen.

Verifieer de fingerprint van je sleutels bij je

correspondenten.

Zorg er voor dat al je sleutels in je sleutelhanger

volledig vertrouwd worden, indien dit nog niet het geval

zou zijn.

Teken al je eigen sleutels en die van de personen van wie

je de sleutel vertrouwt.

==========================================================

fingerprint verifieren:

met commando gpg --edit-key XXXXXXXX , XXXXXXXX heeft

terug dezelfde betekenis als in vraag 4.

in de secundaire promt het commando fpr gebruiken.

dit toont de details van de gekozen sleutel (ID en de

primaire key fingerprint)

eigen sleutels tekenen:

gpg --edit-key XXXXXXXX

-> sign

-> are you sure? -> y ingeven

-> passphrase ingeven

-> save changes? -> y ingeven

sleutels vertrouwen:

gpg --edit-key YYYYYYYY

-> trust

-> opties van niveau worden getoond -> optie 5 kiezen

7. Maak vijf korte tekstbestanden aan om ze nadien aan

andere gebruikers over te maken.

- het eerste bestand encrypteer je dmv conventionele

encryptie op twee manieren:

de eerste keer met het default algortime (welk is dat?),

de tweede keer geef je zelf het algoritme op (bv. AES256)

- het tweede bestand encrypteer je zodat alleen één van je

correspondenten uit je sleutelhanger het bestand kan

decrypteren.

Vraag: met welk algoritme is de encryptie gebeurd?

- het derde bestand encrypteer je zodat dit door minstens

twee gebruikers kan ontcijferd en gelezen worden.

Zorg dat de encryptie met een ander dan het default

algoritme gebeurt.

- het vierde bestand encrypteer je zodat de ontvanger hem

niet zonder meer kan opslaan op zijn harddisk.

- het vijfde bestand encrypteer je conventioneel maar het

resultaat is in radix64.

========================================================

het eerste bestand

-------------------

conventioneel encrypteren: gpg -c bestand1.txt

cat bestand1.txt.gpg -> inhoud is onleesbaar

CAST5 = het default algoritme bij conventionele encryptie

(gevonden bij 'man gpg' bij optie -c)

een zelfgekozen algoritme:

gpg --cipher-algo AES256 -c bestand1.txt

-> vraagt om nieuwe naam --> nieuwe naam ingeven

het tweede bestand

-------------------

optie -e gebruiken voor deze opdracht

gpg -e bestand2.txt

-> vraagt om een userID -> van in --list-key

Het algoritme dat hier gebruikt wordt is het algoritme dat

hoort bij de sleutel.

(Hier DSA and Elgamal gekozen)

het derde bestand

---------------------

hetzelfde als het tweede bestand, maar meer recipients

kiezen op dezelfde manier

(voor beide DSA and Elgamal gekozen)

het vierde bestand

----------------------

niet meer kunnen opslaan op de harde schijf :

--for-your-eyes-only optie gebruiken

hierbij net zoals de vorige keren recipients aanduiden

het vijfde bestand

-------------------

gpg --symmetric -a bestand5.txt

--> optie -a gebruiken

8. Veeg de originele plaintextbestanden uit.

Probeer nu de plaintext te herstellen uitgaande van de

ciphertext van elk bestand.

Controleer wat er al dan niet nog kan gebeuren.

=======================================================

bestand 1:

gpg --decrypt bestand1.txt.gpg

-> vraagt passphrase

na het invullen van de passphrase wordt de inhoud getoond

bestand 2:

gpg --decrypt bestand2.txt.gpg

-> secret key not availible

we hebben de geheime sleutel van de andere persoon niet

bestand 3

gpg --decrypt bestand3.txt.gpg

vraagt passphrase (ik was ook recipient)

moest ik geen recipient zijn, dan zou ik het bericht niet

kunnen encrypteren. Ik heb dan de geheime sleutel van de

andere recipient(s) nodig.

bestand 4

gpg --decrypt bestand4.txt.gpg

kan niet decrypteren, 'secret key not available'

bestand 5

gpg --decrypt bestand5\_symm.txt

vraagt de passphrase, toont daarna de inhoud op het scherm

9. Wissel uit met collega's

================================

uitgewisseld op de virtuele machines

10. Decrypteer alle ontvangen bestanden. Beschrijf in elk

van de gevallen wat er gebeurt.

Vraag: wat is het verschil tussen het al dan niet gebruiken

van de optie -d. Test uit.

===========================================================

bestand 1

gpg --decrypt bestand1.txt.gpg

-> vraagt passphrase

(gedaan met conventionele encryptie)

geeft inhoud correct weer

het bestand dat met AES256 werd geencrypteerd:

gpg --decrypt bestand1\_AES256.txt

-> vraagt passphrase

->geeft inhoud correct weer

(sleutel was DSA and Elgamal)

bestand 2

gpg --decrypt bestand2.txt.gpg

-> vraagt passphrase

-> passphrase correct ingeven

--> inhoud wordt correct getoond

(sleutel was DSA and Elgamal)

bestand 3

gpg --decrypt bestand2.txt.gpg

-> vraagt passphrase

-> passphrase correct ingeven

--> inhoud wordt correct getoond

(sleutel was DSA and Elgamal)

bestand 4

Dit zou normaal onmogelijk moeten zijn om op te slaan.

gpg --decrypt bestand4.txt.gpg > testbestand

vraagt passphrase -> passphrase invullen

we krijgen een foutboodschap

'Note: sender requested "four-your-eyes-only"'

--> test geslaagd

bestand 5

gpg --decrypt bestand5.txt.asc

vraagt passphrase -> passphrase invullen

--> inhoud wordt correct weergegeven

rol optie -d:

het bestand wordt gedecrypteerd en geschreven naar de

standaarduitvoer. als het bestand getekend is, wordt de

handtekening ook geverifieerd. het weigert bestanden die

niet beginnen met een geencrypteerd bericht.

11. Maak drie nieuwe tekstbestanden aan. Behandel ze als

volgt:

- teken het eerste bestand en encrypteer het voor een

van je correspondenten;

- plaats een handtekening op het tweede bestand;

- maak van het derde bestand een gehandtekende file die

bruikbaar is voor e-mail.

=======================================================

bestand 1

---------

gpg -s -e 11\_1.txt

-> vraagt passphrase en recipient

-> optie -s plaatst de handtekening

-> kies een sleutel voor de ontvanger (hier DSA+ElGamal)

bestand 2

----------

gpg -s 11\_2.txt

vraagt passphrase

bestand 3

---------

gpg -s -a 11\_3.txt

vraagt opnieuw de passphrase

optie -a zorgt voor ASCII output

13. Maak van een nieuw tekstbestand een apart

handtekeningcertificaat. Zorg er voor dat deze tekst ook

door 2 of 3 andere gebruikers wordt getekend.

Wissel de tekst en de handtekenigen uit en controleer.

Test uit wat er gebeurt indien je het originele

tekstbestand verwijderd hebt.

===========================================================

gpg -b vraag13.txt

optie -b zorgt voor de aparte handtekening, door de

handtekening te scheiden van de data, kunnen meerdere

personen het bestand 'tekenen'.

om te verifieren:

gpg --verify vraag13.txt.sig

verifieren geslaagd

Als het oorspronkelijke bestand verwijderd is, dan kan je

niet meer verifieren,

want de originele data is weg.

14. Maak van een tekstbestand een clear-signed document.

Wat is de inhoud?

Verstuur het via e-mail naar een correspondent.

Verifieer de handtekening van het ontvangen bestand en

decrypteer.

===========================================================

optie --clearsign gebruiken:

gpg --clearsign vraag14.txt

--> vraagt passphrase

--> vraag14.txt.asc is een ASCII bestand

(een handtekening met leesbare tekens)

verifieren:

gpg --verify vraag14.txt.asc

resultaat: good signature

decrypteren:

gpg --decrypt vraag14.txt.asc

--> toont de inhoud op het scherm, samen met de toestand

van de handtekening