

Velkommen til

PGP tutorial og keysigning workshop

The Camp - Juli 2005

Henrik Lund Kramshøj

hk@security6.net

<http://www.security6.net>

og

Flemming Jacobsen

fj@batmule.dk

Indbyrdes præsentation

Kort introduktion af deltagerne - 3 min.

- Navn
- Forventninger til dette møde
- Hvad jeg skal bruge det her til
- ...

Målet med dette møde

At introducere PGP og GnuPG

At inspirere jer til at bruge kryptering til blandt andet e-mail

At udføre en keysigning - altså udveksle nøgler

Er kryptering interessant?

Sikkerhedsproblemer i netværk er mange

Trådløst udstyr er blevet meget billigt! - endnu flere kan sniffe trafik!

Man vil gerne udveksle hemmeligheder som password m.v. over e-mail

Emneområder

Introduktion - begreber og teknologierne

PGP historik Phil Zimmermann

Basale værktøjer PGP og GPG

Grafiske brugergrænseflader

Keysigning - hvordan

Keysigning - præsentation og identifikation

Film :-)

Kryptografi

Kryptografi er læren om, hvordan man kan kryptere data

Kryptografi benytter algoritmer som sammen med nøgler giver en cif-
fertext - der kun kan læses ved hjælp af den tilhørende nøgle

privat-nøgle kryptografi (eksempelvis AES) benyttes den samme
nøgle til kryptering og dekryptering

offentlig-nøgle kryptografi (eksempelvis RSA) benytter to separate
nøgler til kryptering og dekryptering

Kryptografiske principper

Algoritmerne er kendte

Nøglerne er hemmelige

Nøgler har en vis levetid - de skal skiftes ofte

Et succesfuldt angreb på en krypto-algoritme er enhver genvej som kræver mindre arbejde end en gennemgang af alle nøglerne

Nye algoritmer, programmer, protokoller m.v. skal gennemgås nøje!

Se evt. Snake Oil Warning Signs: Encryption Software to Avoid

<http://www.interhack.net/people/cmcurtin/snake-oil-faq.html>

DES, Triple DES og AES

AES

Advanced Encryption Standard

DES kryptering baseret på den IBM udviklede Lucifer algoritme har været benyttet gennem mange år.

Der er vedtaget en ny standard algoritme Advanced Encryption Standard (AES) som afløser Data Encryption Standard (DES)

Algoritmen hedder Rijndael og er udviklet af Joan Daemen og Vincent Rijmen.

Kilde: <http://csrc.nist.gov/encryption/aes/>
<http://www.esat.kuleuven.ac.be/~rijmen/rijndael/>

Formålet med kryptering

kryptering er den eneste måde at sikre:

fortrolighed

autenticitet / integritet

e-mail og forbindelser

Kryptering af e-mail

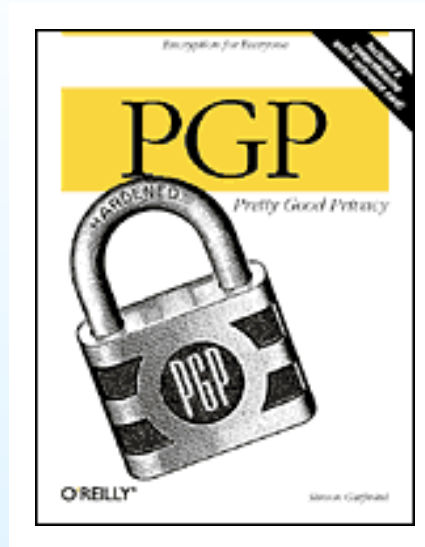
- Pretty Good Privacy - Phil Zimmermann
- PGP = mail sikkerhed

Kryptering af sessioner SSL/TLS

- Secure Sockets Layer SSL / Transport Layer Services TLS
- krypterer data der sendes mellem webservere og klienter
- SSL kan bruges generelt til mange typer sessioner, eksempelvis POP3S, IMAPS, SSH m.fl.

Sender I kreditkortnummeret til en webserver der kører uden https?

Basale tools - PGP



Pretty Good Privacy - PGP

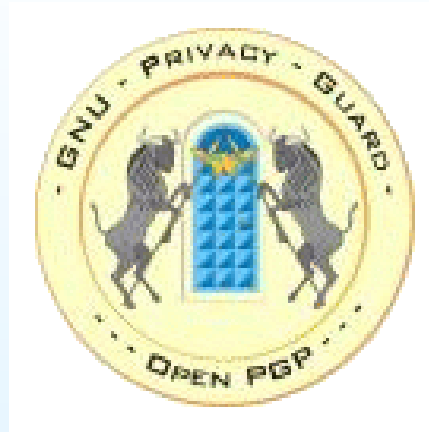
Oprindeligt udviklet af Phil Zimmermann

brug linket: <http://www.pgpi.net/>

kommercielt, men der findes en freeware version

Vist nok eksporteret fra USA på papir

Basale tools - GPG



Gnu Privacy Guard, forkortes GnuPG eller GPG

brug linket: <http://www.gnupg.org/>

Open Source med GPL licens. Findes også til Windows

GnuPG

```
$ gpg --version
gpg (GnuPG) 1.2.4
Copyright (C) 2003 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.
```

```
Home: ~/.gnupg
```

```
Supported algorithms:
```

```
Pubkey: RSA, RSA-E, RSA-S, ELG-E, DSA, ELG
```

```
Cipher: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH
```

```
Hash: MD5, SHA1, RIPEMD160, SHA256
```

```
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

```
$
```

GPGME

Et bibliotek til nem adgang til GPG fra applikationsprogrammer

GnuPG - send en mail fra kommandolinien

```
#!/bin/sh
#
# encrypt a message and send to hlk and CEM

PROGRAM=`basename $0 `

if [ $# -ne 1 ]; then
    echo " to encrypt a message I need a filename!"
    echo "Usage: $0 filename"
    echo "example $0 pentest-report"
    exit 0
fi

filename=$1
#hlk is D1EFBAA6
gpg -e -a -r D1EFBAA6 <$filename > $filename.pgp
$
```

.pgp filen kan nu vedhæftes en mail og sendes sikkert.

GnuPG - verifikation af downloads

```
$ cd /userdata/download/src/postfix/
$ ls -l *.sig
-rw-r--r--  1 hlk  admin  152 13 Sep  2003 postfix-2.0.16.tar.gz.sig
-rw-r--r--  1 hlk  admin  152  3 May 13:34 postfix-2.1.1.tar.gz.sig
$ gpg --verify postfix-2.1.1.tar.gz.sig
gpg: Signature made Mon May  3 19:34:08 2004 CEST using RSA key ID D5327CB9
gpg: Good signature from "wietse venema [wietse@porcupine.org]"
gpg:          aka "wietse venema <wietse@wzv.win.tue.nl>"
$
```

Generering af keys 1

Generering af key

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

- Vælg "DSA and Elgamal"
- Vælg passende keysize - 4096 skader næppe
- Vælg passende expiry - "no expire" vil virke for de fleste
- Brug din officielle mailadresse i forbindelse med dit navn, så Email klienter kan finde din key aytomatisk
- Brug en god passphrase.
En lang sætning som du kan huske, og som ikke kan gættes ud fra kendskab til dig.
- Når keyen genereres, så hjælp med at generere "randomness" i systemet. Det får genereringen til at gå hurtigere, og det giver en bedre key.

Generering af keys 2

Når keyen er genereret bliver der printet et kort sammendrag af indholdet

Dette kan også fås frem med:

```
$ gpg --fingerprint addr@domain.dk
pub      1024D/10019953 2005-05-07
         Key fingerprint = F643 81E1 0D8A 68A0 1024  C503 40B6 7A22 1001 9953
uid          Test Brugger <addr@domain.dk>
sub      4096g/47A33D01 2005-05-07
```

Du har nu en GnuPG key klar til at blive signeret

Er du **sikker** på at du kan huske din passphrase?

Opsætning af defaults

Vi sætter defaults der sikrer:

- Ingen brok over ulåste sider (at låse sider kræver root, dvs. SUID)
- Valg af default keyserver
- Valg af default key (hvis du har flere)
- Valg af karaktersæt

```
$ tail ~/.gnupg/gpg.conf  
no-secmem-warning  
keyserver hkp://pgp.mit.edu/  
default-key 47A33D01  
charset ISO-8859-1
```

Det gør livet lidt lettere

Upload af keys

For at andre kan signere din key skal de have en kopi af din publickey.

Dette kan enten gøres ved at du eksporterer den til en fil og sender filen via Email, diskette, scp, USBkey, etc.:

```
gpg -a --export addr@domain.dk
```

Eller det kan gøres ved at den uploades til keyserverne, hvorfra alle kan hente den:

```
gpg --send-keys 47A33D01
```

Download af keys

Download kan enten foretages via en webside med søgemuligheder

<http://pgp.mit.edu/>

Herfra kan en publickey gemmes i en fil, og importeres med:

```
gpg -import FILE
```

Download kan også foretages direkte hvis keyid kendes:

```
gpg --recv-keys DCC399C7
```

Signering af keys

Keys signeres med:

```
gpg --sign-key addr@domain.dk # Eller keyid
```

Husk at sikre at det nu også er den korrekte key i signerer

Kontroller med:

```
gpg --fingerprint addr@domain.dk
```

Spørgsmål?

Henrik Lund Kramshøj

hk@security6.net

<http://www.security6.net>

Flemming Jacobsen

fj@batmule.dk

I er altid velkomne til at sende spørgsmål på e-mail