Verschlüsseln mit dem Gnu Privacy Guard

Wie man seine Paranoia richtig pflegt

K. Rothemund

Technikabend der Rostocker Linux User Group (RoLUG)
21.4.2005

1. **Einleitung**
   - Warum verschlüsseln?
   - Von symmetrischen und asymmetrischen Schlüsseln
   - Anwendung von Kryptografie
2. **Die Praxis: GnuPG**
   - Erzeugen eines GnuPG -Schlüssels
   - Verschlüsseln und Signieren von Dateien
   - Frontends zu GnuPG
   - Verschlüsseln/Signieren von E-Mails
3. **Web of Trust**

Gliederung

1. **Einleitung**
   - Warum verschlüsseln?
   - Von symmetrischen und asymmetrischen Schlüsseln
   - Anwendung von Kryptografie
2. Die Praxis: GnuPG
   - Erzeugen eines GnuPG -Schlüssels
   - Verschlüsseln und Signieren von Dateien
   - Frontends zu GnuPG
   - Verschlüsseln/Signieren von E-Mails
3. Web of Trust

Warum verschlüsseln?

1. Privatsphäre - Ist ein Grundrecht!
2. Privatsphäre - Und ich muss mich nicht dafür rechtfertigen!
3. Privatsphäre - Auch, wenn ich nichts zu verbergen habe.

aber auch:

- Vertraulichkeit
- Integrität
- Authentizität

Lösung:

Kryptografie - Infos nicht für jeden zugänglich machen

Warum verschlüsseln?

1. Privatsphäre - Ist ein Grundrecht!
2. Privatsphäre - Und ich muss mich nicht dafür rechtfertigen!
3. Privatsphäre - Auch, wenn ich nichts zu verbergen habe.

aber auch:

- Vertraulichkeit
- Integrität
- Authentizität

Lösung:

Kryptografie - Infos nicht für jeden zugänglich machen

Warum verschlüsseln?

1. Privatsphäre - Ist ein Grundrecht!
2. Privatsphäre - Und ich muss mich nicht dafür rechtfertigen!
3. Privatsphäre - Auch, wenn ich nichts zu verbergen habe.

aber auch:

- Vertraulichkeit
- Integrität
- Authentizität

Lösung:

Kryptografie - Infos nicht für jeden zugänglich machen

Gliederung

1. **Einleitung**
   - Warum verschlüsseln?
   - Von symmetrischen und asymmetrischen Schlüsseln
   - Anwendung von Kryptografie
2. Die Praxis: GnuPG
   - Erzeugen eines GnuPG -Schlüssels
   - Verschlüsseln und Signieren von Dateien
   - Frontends zu GnuPG
   - Verschlüsseln/Signieren von E-Mails
3. Web of Trust

Von symmetrischen und asymmetrischen Schlüsseln

symmetrischer Schlüssel:

- nur ein Schlüssel
- ver- und entschlüsseln mit dem selben Schlüssel

Asymmetrischer Schlüssel:

- Schlüsselpaar: 2 verschiedene Schlüssel
- mit einem ver- und mit dem anderen entschlüsseln
- einen Teil des Schlüsselpaars ist öffentlich ("public key")
- den anderen geheim halten ("private key")

Von symmetrischen und asymmetrischen Schlüsseln

symmetrischer Schlüssel:

- nur ein Schlüssel
- ver- und entschlüsseln mit dem selben Schlüssel

Asymmetrischer Schlüssel:

- Schlüsselpaar: 2 verschiedene Schlüssel
- mit einem ver- und mit dem anderen entschlüsseln
- einen Teil des Schlüsselpaars ist öffentlich ("public key")
- den anderen geheim halten ("private key")

Theorie: RSA (asymmetrisch)

Man braucht:

p,q : große Primzahlen

n : Produkt der beiden $n = p \times q$

e : Kodierschlüssel (in der Nähe von $(p-1) \times (q-1)$)

d : Dekodierschlüssel $d = e^{-1} (\mod (p-1) \times (q-1))$

Kodieren und dekodieren:

Kodieren : $t = b^e \mod n$

Dekodieren : $b = t^d \mod n$

b: Botschaft(Klartext), t: chiffrierter Text

Theorie: RSA (asymmetrisch)

Man braucht:

p,q : große Primzahlen

n : Produkt der beiden $n = p \times q$

e : Kodierschlüssel (in der Nähe von $(p-1) \times (q-1)$)

d : Dekodierschlüssel $d = e^{-1} (\mod (p-1) \times (q-1))$

Kodieren und dekodieren:

Kodieren : $t = b^e \mod n$

Dekodieren : $b = t^d \mod n$

b: Botschaft(Klartext), t: chiffrierter Text

Wie sicher ist RSA?

Sicherheit von RSA hängt davon ab, dass

- Zahlen p und q geheim bleiben
  (aus p und q und dem Kodierschlüssel e läßt sich sofort der Decodierschlüssel d berechnen)
- Produkt n schwierig in Primfaktoren zu zerlegen ist
  (es kann dann wieder der Dekodierschlüssel d berechnet werden)
- es kein anderes Verfahren existiert, aus n und dem Kodierschlüssel e den Dekodierschlüssel d zu berechnen

Gliederung

1. Einleitung
   - Warum verschlüsseln?
   - Von symmetrischen und asymmetrischen Schlüsseln
   - Anwendung von Kryptografie
2. Die Praxis: GnuPG
   - Erzeugen eines GnuPG -Schlüssels
   - Verschlüsseln und Signieren von Dateien
   - Frontends zu GnuPG
   - Verschlüsseln/Signieren von E-Mails
3. Web of Trust

Anwendung von Kryptografie

Verschlüsseln eigener Dateien : Verschlüsselung mit symmetrischem und [ffentlichem Schlüssel

Sichere Datenübermittlung : Verschlüsseln mit öffentlichem Schlüssel des Empfängers (eventuell mehrere Empfänger)

Signieren von Dateien : bilde Prüfsumme mit privatem Schlüssel → jeder kann mit dem öffentlichen Schlüssel überprüfen, dass die Daten

1. unverändert, und
2. vom Unterzeichner sind

z.B. Sourcen (speziell von Kryptografie-Programmen) sollten signiert sein → sonst Vorsicht!

Signieren von Korrespondenz : z.B. E-Mail (Stichwort: Phishing)

Die Praxis: GnuPG

Bemerkungen

- gehe nicht auf Pretty Good Privacy (PGP) ein, da es mittlerweile kommerziell zumindest aber nicht frei ist.
  Grundzüge sind aber gleich (GnuPG besitzt alle Optionen von PGP).

- GnuPG ist Open Source: Vertrauen in Code und Algoritmen
- GnuPG unterstützt OpenPG-Standard → erweiterbar (einfache Einbindung neuer Veschlüsselungsalgorithmen)
- verschiedene Schlüssel-Typen
- verschiedene Verschlüsselungsverfahren
- ABER: GnuPG ist ein komplexes Programm mit vielen Möglichkeiten → sehr viele Optionen
- die wichtigste Option: `gpg -help`
- Hinweis: Manpages und Dokumentation

Die Praxis: GnuPG

Bemerkungen

- gehe nicht auf Pretty Good Privacy (PGP) ein, da es mittlerweile kommerziell zumindest aber nicht frei ist.
  Grundzüge sind aber gleich (GnuPG besitzt alle Optionen von PGP).

- GnuPG ist Open Source: Vertrauen in Code und Algoritmen
- GnuPG unterstützt OpenPG-Standard → erweiterbar (einfache Einbindung neuer Veschlüsselungsalgorithmen)
- verschiedene Schlüssel-Typen
- verschiedene Verschlüsselungsverfahren
- ABER: GnuPG ist ein komplexes Programm mit vielen Möglichkeiten → sehr viele Optionen
- die wichtigste Option: `gpg -help`
- Hinweis: Manpages und Dokumentation

Die Praxis: GnuPG

Bemerkungen

- gehe nicht auf Pretty Good Privacy (PGP) ein, da es mittlerweile kommerziell zumindest aber nicht frei ist.
  Grundzüge sind aber gleich (GnuPG besitzt alle Optionen von PGP).

- GnuPG ist Open Source: Vertrauen in Code und Algoritmen
- GnuPG unterstützt OpenPG-Standard → erweiterbar (einfache Einbindung neuer Veschlüsselungsalgorithmen)
- verschiedene Schlüssel-Typen
- verschiedene Verschlüsselungsverfahren
- ABER: GnuPG ist ein komplexes Programm mit vielen Möglichkeiten → sehr viele Optionen
- die wichtigste Option: `gpg -help`
- Hinweis: Manpages und Dokumentation

Wichtige Optionen und Befehle

Schlüssel-Handling

`-gen-key,-gen-revoke` : neues Schlüsselpaar/-widerrufszertifikat erzeugen

`-edit-key` : bearbeiten eines Schlüssels

`-fingerprint` : Fingerabdruck anzeigen

Dateien bearbeiten

`-sign` : Signieren einer Datei

`-encrypt,-decrypt` : Ver- und Entschlüsseln einer Datei

Schlüssel-Ring

`-list-keys,-list-sigs` : Liste der Schlüssel (mit Signaturen)

`-delete-keys,-delete-secret-keys` : Schlüssel aus Schlüsselbund entfernen

Wichtige Optionen und Befehle

Schlüssel-Handling

`-gen-key,-gen-revoke` : neues Schlüsselpaar/-widerrufszertifikat erzeugen

`-edit-key` : bearbeiten eines Schlüssels

`-fingerprint` : Fingerabdruck anzeigen

Dateien bearbeiten

`-sign` : Signieren einer Datei

`-encrypt,-decrypt` : Ver- und Entschlüsseln einer Datei

Schlüssel-Ring

`-list-keys,-list-sigs` : Liste der Schlüssel (mit Signaturen)

`-delete-keys,-delete-secret-keys` : Schlüssel aus Schlüsselbund entfernen

Wichtige Optionen und Befehle

Schlüssel-Handling

`-gen-key,-gen-revoke` : neues Schlüsselpaar/-widerrufszertifikat erzeugen

`-edit-key` : bearbeiten eines Schlüssels

`-fingerprint` : Fingerabdruck anzeigen

Dateien bearbeiten

`-sign` : Signieren einer Datei

`-encrypt,-decrypt` : Ver- und Entschlüsseln einer Datei

Schlüssel-Ring

`-list-keys,-list-sigs` : Liste der Schlüssel (mit Signaturen)

`-delete-keys,-delete-secret-keys` : Schlüssel aus Schlüsselbund entfernen

Wichtige Optionen und Befehle

Schlüsselaustausch

`-sign-key` : Schlüssel signieren

`-export,-import` : Schlüssel exportieren/importieren

`-send-keys,-recv-keys` : Schlüssel mit Schlüsselserver austauschen

`-refresh-keys` : auf Schlüsselserver aktualisieren

Optionen:

`-a,-armor` : Ausgabe mit ASCII-Hülle versehen

`-r,-recipient` : Adressat spezifizieren

`-u, -local-user` : Mit dieser User-ID signieren

`-o,-output` : Ausgabedatei explizit angeben

`-n,-dry-run` : Testlauf, keine Änderungen durchführen

`-openpgp` : strikt an den OpenPGP-Standard halten

`-pgp2` : PGP 2.x kompatibles Verhalten

Wichtige Optionen und Befehle

Schlüsselaustausch

`-sign-key` : Schlüssel signieren

`-export,-import` : Schlüssel exportieren/importieren

`-send-keys,-recv-keys` : Schlüssel mit Schlüsselserver austauschen

`-refresh-keys` : auf Schlüsselserver aktualisieren

Optionen:

`-a,-armor` : Ausgabe mit ASCII-Hülle versehen

`-r,-recipient` : Adressat spezifizieren

`-u, -local-user` : Mit dieser User-ID signieren

`-o,-output` : Ausgabedatei explizit angeben

`-n,-dry-run` : Testlauf, keine Änderungen durchführen

`-openpgp` : strikt an den OpenPGP-Standard halten

`-pgp2` : PGP 2.x kompatibles Verhalten

Gliederung

1 Einleitung
- Warum verschlüsseln?
- Von symmetrischen und asymmetrischen Schlüsseln
- Anwendung von Kryptografie

2 **Die Praxis: GnuPG**
- Erzeugen eines GnuPG -Schlüssels
- Verschlüsseln und Signieren von Dateien
- Frontends zu GnuPG
- Verschlüsseln/Signieren von E-Mails

3 Web of Trust

Erzeugen eines GnuPG -Schlüssels

```
karo@worf$ gpg -gen-key
Bitte wählen Sie, welche Art von Schlüssel Sie möchten:
   (1) DSA and Elgamal (default)
   (2) DSA (nur signieren/beglaubigen)
   (5) RSA (nur signieren/beglaubigen)
Ihre Auswahl? 1
DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) <RETURN>
Die verlangte Schlüssellänge beträgt 2048 Bit
Bitte wählen Sie, wie lange der Schlüssel gültig bleiben soll
         0 = Schlüssel verfällt nie
      <n>  = Schlüssel verfällt nach n Tagen
                    [...]
Wie lange bleibt der Schlüssel gültig? (0) <RETURN>
Key verfällt nie.
Is this correct? (y/N) y
```

Erzeugen eines GnuPG -Schlüssels

```
karo@worf$ gpg -gen-key
Bitte wählen Sie, welche Art von Schlüssel Sie möchten:
   (1) DSA and Elgamal (default)
   (2) DSA (nur signieren/beglaubigen)
   (5) RSA (nur signieren/beglaubigen)
Ihre Auswahl? 1
DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) <RETURN>
Die verlangte Schlüssellänge beträgt 2048 Bit
Bitte wählen Sie, wie lange der Schlüssel gültig bleiben soll
         0 = Schlüssel verfällt nie
      <n>  = Schlüssel verfällt nach n Tagen
                    [...]
Wie lange bleibt der Schlüssel gültig? (0) <RETURN>
Key verfällt nie.
Is this correct? (y/N) y
```

Erzeugen eines GnuPG -Schlüssels

```
karo@worf$ gpg -gen-key
Bitte wählen Sie, welche Art von Schlüssel Sie möchten:
   (1) DSA and Elgamal (default)
   (2) DSA (nur signieren/beglaubigen)
   (5) RSA (nur signieren/beglaubigen)
Ihre Auswahl? 1
DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) <RETURN>
Die verlangte Schlüssellänge beträgt 2048 Bit
Bitte wählen Sie, wie lange der Schlüssel gültig bleiben soll
         0 = Schlüssel verfällt nie
      <n>  = Schlüssel verfällt nach n Tagen
                    [...]
Wie lange bleibt der Schlüssel gültig? (0) <RETURN>
Key verfällt nie.
Is this correct? (y/N) y
```

Erzeugen eines GnuPG -Schlüssels

```
You need a user ID to identify your key; the software  [...]
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter)
  <heinrichh@duesseldorf.de>"
Ihr Name ("Vorname Nachname"): Blah Blahmann
E-Mail-Adresse: blah@blah.de
Kommentar: Nur zu Demozwecken
Sie haben diese User-ID gewählt:
    "Blah Blahmann (Nur zu Demozwecken) <blah@blah.de>"
Ändern: (N)ame, (K)ommentar, (E)-Mail oder (F)ertig/(B)eenden?f
Sie benötigen eine Passphrase, um den geheimen
                                        Schlüssel zu schützen.
Geben Sie die Passphrase ein: <PASSPHRASE>
Wir müssen eine ganze Menge Zufallswerte
  erzeugen. [...]
+++++...+++++++++++++++++++++++++++++
             [...]
+++++++.+++++.........+++++
```

Erzeugen eines GnuPG -Schlüssels

```
 You need a user ID to identify your key; the software  [...]
 from the Real Name, Comment and Email Address in this form:
     "Heinrich Heine (Der Dichter)
   <heinrichh@duesseldorf.de>"
 Ihr Name ("Vorname Nachname"): Blah Blahmann
 E-Mail-Adresse: blah@blah.de
 Kommentar: Nur zu Demozwecken
 Sie haben diese User-ID gewählt:
     "Blah Blahmann (Nur zu Demozwecken) <blah@blah.de>"
Ändern: (N)ame, (K)ommentar, (E)-Mail oder (F)ertig/(B)eenden?f
 Sie benötigen eine Passphrase, um den geheimen
                                        Schlüssel zu schützen.
 Geben Sie die Passphrase ein: <PASSPHRASE>
 Wir müssen eine ganze Menge Zufallswerte
   erzeugen. [...]
 +++++...+++++++++++++++++++++++++++++
              [...]
 +++++++.+++++.........+++++
```

Erzeugen eines GnuPG -Schlüssels

```
 You need a user ID to identify your key; the software  [...]
 from the Real Name, Comment and Email Address in this form:
     "Heinrich Heine (Der Dichter)
   <heinrichh@duesseldorf.de>"
 Ihr Name ("Vorname Nachname"): Blah Blahmann
 E-Mail-Adresse: blah@blah.de
 Kommentar: Nur zu Demozwecken
 Sie haben diese User-ID gewählt:
     "Blah Blahmann (Nur zu Demozwecken) <blah@blah.de>"
Ändern: (N)ame, (K)ommentar, (E)-Mail oder (F)ertig/(B)eenden?f
 Sie benötigen eine Passphrase, um den geheimen
                                        Schlüssel zu schützen.
 Geben Sie die Passphrase ein: <PASSPHRASE>
 Wir müssen eine ganze Menge Zufallswerte
   erzeugen. [...]
 +++++...+++++++++++++++++++++++++++++
             [...]
 +++++++.+++++.........+++++
```

Erzeugen eines GnuPG -Schlüssels

```
 You need a user ID to identify your key; the software  [...]
 from the Real Name, Comment and Email Address in this form:
     "Heinrich Heine (Der Dichter)
   <heinrichh@duesseldorf.de>"
 Ihr Name ("Vorname Nachname"): Blah Blahmann
 E-Mail-Adresse: blah@blah.de
 Kommentar: Nur zu Demozwecken
 Sie haben diese User-ID gewählt:
     "Blah Blahmann (Nur zu Demozwecken) <blah@blah.de>"
Ändern: (N)ame, (K)ommentar, (E)-Mail oder (F)ertig/(B)eenden?f
 Sie benötigen eine Passphrase, um den geheimen
                                        Schlüssel zu schützen.
 Geben Sie die Passphrase ein: <PASSPHRASE>
 Wir müssen eine ganze Menge Zufallswerte
   erzeugen. [...]
 +++++...+++++++++++++++++++++++++++++
              [...]
 +++++++.+++++.........+++++
```

Erzeugen eines GnuPG -Schlüssels

```
gpg: key F1E2D662 marked as ultimately trusted
Öffentlichen und geheimen Schlüssel erzeugt und signiert.

gpg: ''Trust-DB'' wird überprüft
                [...]
gpg: nächste ''Trust-DB''-Pflichtüberprüfung am 2005-07-01
pub  1024D/F1E2D662 2005-04-05
     Key fingerprint = 99CE 769B B8FB AAB5 61C2
                       9468 8638 3A33 F1E2 D662
uid        Blah Blahmann (Nur zu Demozwecken) <blah@blah.de>
sub  2048g/DD8734EF 2005-04-05
karo@worf$
```

Erzeugen eines GnuPG -Schlüssels

```
gpg: key F1E2D662 marked as ultimately trusted
Öffentlichen und geheimen Schlüssel erzeugt und signiert.

gpg: ''Trust-DB'' wird überprüft
                  [...]
gpg: nächste ''Trust-DB''-Pflichtüberprüfung am 2005-07-01
pub  1024D/F1E2D662 2005-04-05
     Key fingerprint = 99CE 769B B8FB AAB5 61C2
                       9468 8638 3A33 F1E2 D662
uid        Blah Blahmann (Nur zu Demozwecken) <blah@blah.de>
sub  2048g/DD8734EF 2005-04-05
karo@worf$
```

Erzeugen eines GnuPG -Schlüssels

```
gpg: key F1E2D662 marked as ultimately trusted
Öffentlichen und geheimen Schlüssel erzeugt und signiert.

gpg: ''Trust-DB'' wird überprüft
                  [...]
gpg: nächste ''Trust-DB''-Pflichtüberprüfung am 2005-07-01
pub  1024D/F1E2D662 2005-04-05
     Key fingerprint = 99CE 769B B8FB AAB5 61C2
                       9468 8638 3A33 F1E2 D662
uid        Blah Blahmann (Nur zu Demozwecken) <blah@blah.de>
sub  2048g/DD8734EF 2005-04-05
karo@worf$
```

Erzeugen eines GnuPG -Schlüssels

```
gpg: key F1E2D662 marked as ultimately trusted
Öffentlichen und geheimen Schlüssel erzeugt und signiert.

gpg: ''Trust-DB'' wird überprüft
                [...]
gpg: nächste ''Trust-DB''-Pflichtüberprüfung am 2005-07-01
pub  1024D/F1E2D662 2005-04-05
     Key fingerprint = 99CE 769B B8FB AAB5 61C2
                       9468 8638 3A33 F1E2 D662
uid        Blah Blahmann (Nur zu Demozwecken) <blah@blah.de>
sub  2048g/DD8734EF 2005-04-05
karo@worf$
```

Erzeugen eines GnuPG -Schlüssels

Bemerkungen

- Ergebnis: Schlüsselpaar (Secret und Public Key) im Schlüsselring unter (`$HOME/.gnupg/`)
- GnuPG signiert den neuen Schlüssel direkt
- Am besten: sofort Widerrufszertifikat (Revocation Certificate) erzeugen: `gpg -gen-revoke [KEY-ID]`
- Zur Sicherheit: den privaten Schlüssel (und das Revokation Certificate) sollte man (in ASCII) exportieren und ausdrucken
- geheime Schlüssel immer sicher aufbewahren
- Schlüssel ist zusätzlich durch Passphrase geschützt

Der Schlüsselring

Schlüsselring liegt unter `$HOME/.gnupg/`

```
karo@worf:~> ls -l .gnupg/
 drwxr-sr-x  3 karo  mafia     512 18 Apr 21:46 ./
 drwxr-xr-x  6 karo  mafia     512 23 Dez 18:48 ../
 -rw-r-r-  1 karo  mafia     117  3 Apr 23:18 gpa.conf
 -rw----  1 karo  mafia    1056 23 Dez 18:47 options
 -rw-r-r-  1 karo  mafia  132703  5 Apr 19:39 pubring.gpg
 -rw----  1 karo  mafia     600  8 Apr 21:57 random_seed
 -rw----  1 karo  mafia    9401  5 Apr 19:39 secring.gpg
 -rw----  1 karo  mafia    2840  5 Apr 19:39 trustdb.gpg
karo@worf:~>
```

beachte die Rechte von `secring.gpg` und `trustdb.gpg`

Der Schlüsselring

Schlüsselring liegt unter `$HOME/.gnupg/`

```
karo@worf:~> ls -l .gnupg/
 drwxr-sr-x  3 karo  mafia     512 18 Apr 21:46 ./
 drwxr-xr-x  6 karo  mafia     512 23 Dez 18:48 ../
 -rw-r-r-  1 karo  mafia     117  3 Apr 23:18 gpa.conf
 -rw----  1 karo  mafia    1056 23 Dez 18:47 options
 -rw-r-r-  1 karo  mafia  132703  5 Apr 19:39 pubring.gpg
 -rw----  1 karo  mafia     600  8 Apr 21:57 random_seed
 -rw----  1 karo  mafia    9401  5 Apr 19:39 secring.gpg
 -rw----  1 karo  mafia    2840  5 Apr 19:39 trustdb.gpg
karo@worf:~>
```

beachte die Rechte von `secring.gpg` und `trustdb.gpg`

Wie werden die Schlüssel verteilt?

Schlüssel-Verteilung:

- verteilt wird nur der öffentliche Schlüssel
- über Schlüsselserver (z.B. `http://www.dfn-pca.de/pgpkserv/`)
- auf der Homepages (als ASCII)
- mittels Dateien (ASCII und binär)
  - Datenträger: Diskette, USB-Stick etc.
  - elektronisch: z.B. E-Mail

Gliederung

1 Einleitung
- Warum verschlüsseln?
- Von symmetrischen und asymmetrischen Schlüsseln
- Anwendung von Kryptografie

2 **Die Praxis: GnuPG**
- Erzeugen eines GnuPG -Schlüssels
- Verschlüsseln und Signieren von Dateien
- Frontends zu GnuPG
- Verschlüsseln/Signieren von E-Mails

3 Web of Trust

Verschlüsseln und Signieren von Dateien

Kommandozeile

- Datei verschlüsseln: `gpg -e [file]`
  oder `gpg -encrypt [file]`
- Datei entschlüsseln: `gpg -d [file]`
  oder `gpg -decrypt [file]`
- Signieren einer Datei: `gpg -s [file]`
  oder `gpg -sign [file]`
  oder mit Klartextsignatur: `gpg -clearsign [file]`

Gliederung

1. Einleitung
   - Warum verschlüsseln?
   - Von symmetrischen und asymmetrischen Schlüsseln
   - Anwendung von Kryptografie
2. **Die Praxis: GnuPG**
   - Erzeugen eines GnuPG -Schlüssels
   - Verschlüsseln und Signieren von Dateien
   - Frontends zu GnuPG
   - Verschlüsseln/Signieren von E-Mails
3. Web of Trust

Frontends zu GnuPG

Kommandozeile

unhandlich und unübersichtlich zu bedienen; aber auch sehr vielfältig

Frontend

für die einfacheren und alltäglichen Aufgaben:

- Dateien ver- und entschlüsseln
- Signaturen erzeugen und prüfen
- Schlüsselringverwaltung (private Schlüssel, Import und Export von öffentlichen Schlüsseln)
- Schlüssel beglaubigen/signieren
- Trust-DB verwalten

nicht alles machbar (z.B. zus. User-IDs, neue Subschlüssel erzeugen)

Frontends zu GnuPG

Kommandozeile

unhandlich und unübersichtlich zu bedienen; aber auch sehr vielfältig

Frontend

für die einfacheren und alltäglichen Aufgaben:

- Dateien ver- und entschlüsseln
- Signaturen erzeugen und prüfen
- Schlüsselringverwaltung (private Schlüssel, Import und Export von öffentlichen Schlüsseln)
- Schlüssel beglaubigen/signieren
- Trust-DB verwalten

nicht alles machbar (z.B. zus. User-IDs, neue Subschlüssel erzeugen)

Frontends zu GnuPG

GPA - Gnu Privacy Guard

GNU Privacy Assistant - Keyring Editor

File Edit Keys Server Windows Help

Keyring Editor

Key ID	Expiry Date	Owner Trust	Key Validity	User Name
[illegible]	03/06/06	Full	Fully Valid	[illegible]
[illegible]	12/31/03	Unknown	Expired	[illegible]
[illegible]	09/26/03	Full	Expired	[illegible]
[illegible]	never expires	Unknown	Revoked	[Revoked] [illegible]
[illegible]	never expires	Unknown	Unknown	[illegible]
[illegible]	04/29/05	Unknown	Revoked	[Revoked] [illegible]
7BAA61AC	never expires	Ultimate	Fully Valid	Photor (I am Photor) <webmaster@photor.de>
[illegible]	never expires	Unknown	Fully Valid	[illegible]
[illegible]	never expires	Full	Fully Valid	[illegible]
094AE55F	never expires	Ultimate	Fully Valid	Photor (I am Photor) <photor@photor.de>
[illegible]	11/15/04	Unknown	Expired	[illegible]
[illegible]	03/29/04	Unknown	Expired	[illegible]

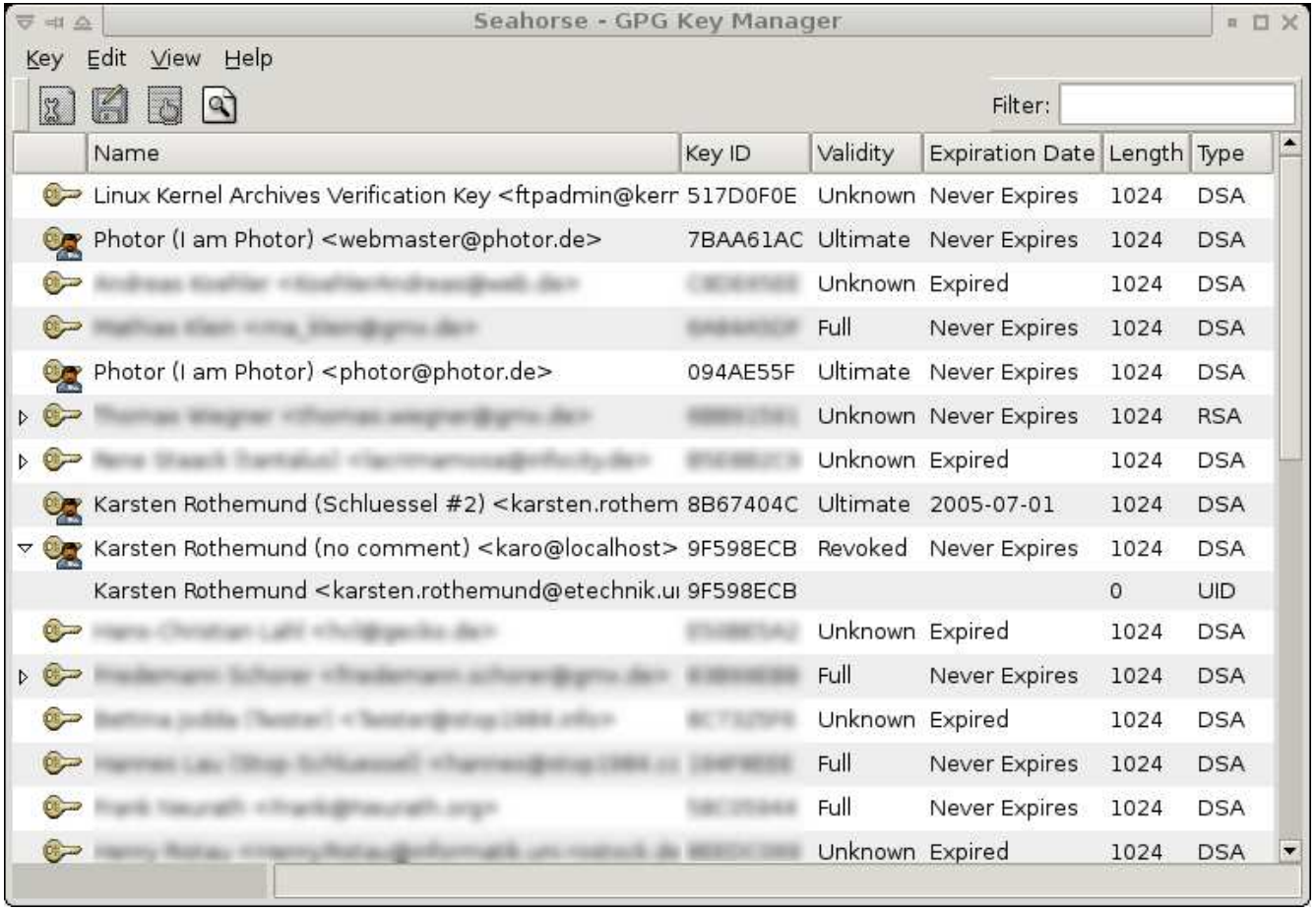
Details | Signatures | Subkeys

The key has both a private and a public part
The key can be used for certification, signing and encryption.
User Name: Photor (I am Photor) <photor@photor.de>
Key ID: 094AE55F
Fingerprint: C0BC 9AF3 A3E4 E55F 2AA3 715E 62A3 4C0A 094A E55F
Expires at: never expires
Owner Trust: Ultimate
Key Validity: Fully Valid
Key Type: DSA 1024 bits
Created at: 11/20/03

Selected Default Key: Photor (I am Photor) <photor@photor.de> 094AE55F

Frontends zu GnuPG

Seahorse - GPG Schlüsselveraltung



Gliederung

1 Einleitung
- Warum verschlüsseln?
- Von symmetrischen und asymmetrischen Schlüsseln
- Anwendung von Kryptografie

2 **Die Praxis: GnuPG**
- Erzeugen eines GnuPG -Schlüssels
- Verschlüsseln und Signieren von Dateien
- Frontends zu GnuPG
- Verschlüsseln/Signieren von E-Mails

3 Web of Trust

Verschlüsseln/Signieren von E-Mails

Programme und Interfaces

Mozilla/Thunderbird Enigmail, ein Plug-In (sowohl Linux/*BSD/UNIX als auch Windows)

Mutt (UNIX-Derivate) Aufruf von GnuPG , ist meist bereits vorkonfiguriert

KMail,Evolution (KDE): bereits integriert

Outlook und andere Windoze-Mailer: ???

Web of Trust

Problem

Wie kann ich sicher sein, dass der Schlüssel `0xF1E2D662` wirklich Blah Blahmann gehört?

Lösung: Netz des Vertrauens ("Web of Trust")

Grundlage : gegenseitiges Signieren der Schlüssel

Vorraussetzung : Schlüssel nur dann signieren, wenn die Identität des Inhabers sicher ist!

Ergebnis : Netz von Leuten, die sich ihr Vertrauen aussprechen.

Vorteil : keine zentrale Speicherung von geheimen Schlüsseln/Daten

Web of Trust

Problem

Wie kann ich sicher sein, dass der Schlüssel `0xF1E2D662` wirklich Blah Blahmann gehört?

Lösung: Netz des Vertrauens ("Web of Trust")

Grundlage : gegenseitiges Signieren der Schlüssel

Vorraussetzung : Schlüssel nur dann signieren, wenn die Identität des Inhabers sicher ist!

Ergebnis : Netz von Leuten, die sich ihr Vertrauen aussprechen.

Vorteil : keine zentrale Speicherung von geheimen Schlüsseln/Daten

Zum Schluss

Meine Daten

- Mail-Adresse: `photor@photor.de`
- GnuPG-Schlüssel: `0xF28CEDCD`
- Fingerprint:
  `C0BC 9AF3 A3E4 E55F 2AA3 715E 62A3 4C0A 094A E55F`