28.2.5 GPG – Exkurs

Eine Alternative zu den beschriebenen Verfahren, um den Inhalt einer Datei mit einem Passwort zu schützen besteht darin, eine Datei mit einem öffentlichen Schlüssel der *GnuPG-Verschlüsselung* zu verschlüsseln. Die Alternative bietet sich an, wenn man häufig Dateien – zum Beispiel Backup-Dateien von Datenbanken – oder Verzeichnis-Archive oder EMails zu verschlüsseln hat. Dann lohnt es, ein Schlüsselpaar aus privatem und öffentlichem Schlüssel mit GnuPG zu erzeugen, weil das Programm-Paket *gnupg* zum Beispiel bei allen aktuellen Ubuntu-Versionen vorinstalliert ist. Ein Blick in die Programmhilfe ist unentbehrlich, bevor Sie zum ersten Mal mit dem Programm GnuPG (gpg) arbeiten:

```
hans@linux:~$ gpg –help
```

Ein neues Schlüsselpaar aus privatem und dazu gehörendem öffentlichen Schlüssel können Sie ganz einfach im Terminal erzeugen, weil Sie durch den gesamten Vorgang geführt werden:

```
hans@linux:~$ gpg --gen-key
gpg (GnuPG) 1.4.11; Copyright (C) 2010 Free Software Foundation, Inc.
...
Bitte wählen Sie, welche Art von Schlüssel Sie möchten:
   (1) RSA und RSA (voreingestellt)
   (2) DSA und Elgamal
   (3) DSA (nur unterschreiben/beglaubigen)
   (4) RSA (nur signieren/beglaubigen)
Ihre Auswahl? 1
RSA-Schlüssel können zwischen 1024 und 4096 Bit lang sein.
Welche Schlüssellänge wünschen Sie? (2048)
...
```

28.2.5.1 Programm 'Seahorse'

Unter *http://wiki.ubuntuusers.de/Seahorse* finden Sie eine gute Beschreibung zur Nutzung des grafischen Frontends für das Verschlüsselungsprogramm 'GnuPG', wobei im ersten Teil nur die Erzeugung eines Schlüsselpaares im Mittelpunkt steht. Sie starten das Programm *Seahorse* über *Anwendungen* | *Zubehör* | *Passwörter und Verschlüsselung* oder durch Aufruf des Befehls 'seahorse' in einer Konsole. Folgen Sie den einzelnen Formularen nach dem Start mit Datei> Neu. Wählen Sie den Eintragstyp und geben Sie die geforderten Eingaben ein:

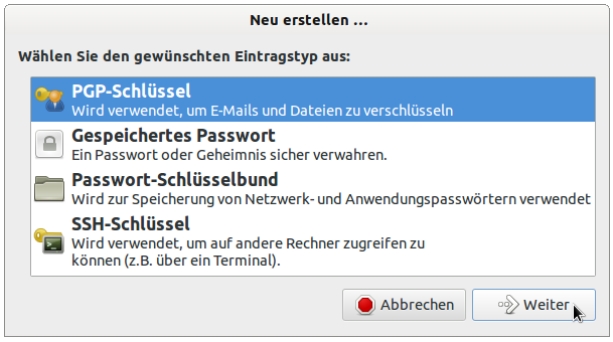

Abbildung 28.2.5.1.1: Wahl des Eintragstyps

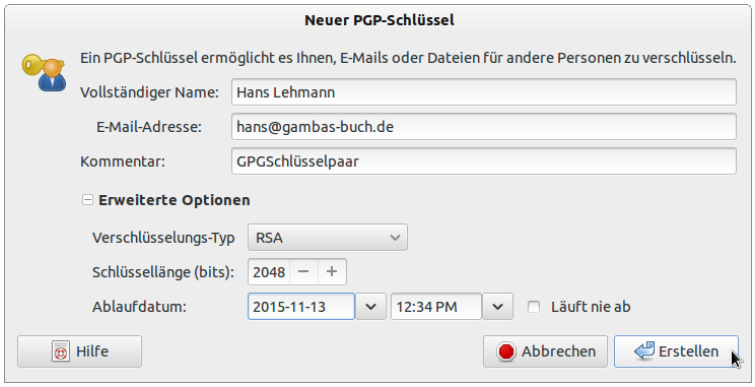

Abbildung 28.2.5.1.2: Eingabe der Schlüssel-Daten

- Ein Ablaufdatum sollte gesetzt werden.
- Die Option von "*Läuft nie ab"* sollte man vermeiden.
- Nach der Bestätigung 'Erstellen' wird ein Schlüssel(-Paar) erzeugt.

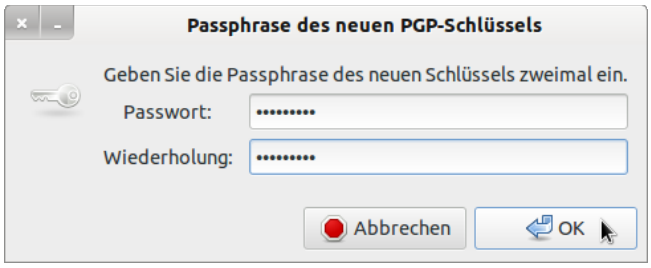

Abbildung 28.2.5.1.3: Eingabe des privaten Schlüssels

Zum Abschluss wird mit *Exportieren* der öffentliche Schlüssel in eine Text-Datei mit der Extension *.asc* geschrieben, die im *Home-Verzeichnis* abgespeichert wird. Den vorgeschlagenen Datei-Namen sollte man so ändern, dass keine Leerzeichen enthalten sind: hans_lehmann.asc oder hans.lehmann.asc.

Unter *Eigenschaften* kann man sich weitere Details zum generierten öffentlichen Schlüssel ansehen.

28.2.5.2 Verwendung des erzeugten Schlüsselpaares

- Der *öffentliche* Schlüssel wird an eine Person verschickt, die mit diesem Schlüssel eine Datei, ein Archiv oder eine EMail verschlüsseln kann oder er wird auf einem Key-Server deponiert.
- Der Inhalt der mit einem *öffentlichen* Schlüssel geschützten Dateien kann nur von der Person gelesen werden, die den *privaten* Schlüssel des Schlüsselpaares besitzt.

28.2.5.3 Verschlüsseln einer einzelnen Datei

Ausgangssituation:

- Die Personen E und S verabreden, dass S an E ausgewählte Dateien schickt, die S mit dem öffentlichen Schlüssel von E vorher verschlüsselt.
- Person E schickt an Person S den öffentlichen Schlüssel eines GPG-Schlüsselpaares.
- S importiert den öffentlichen Schlüssel – zum Beispiel mit Hilfe des Programms 'seahorse' und kennt damit genau die Informationen, um eine ausgewählte Datei mit dem öffentlichen Schlüssel von E zu verschlüsseln.

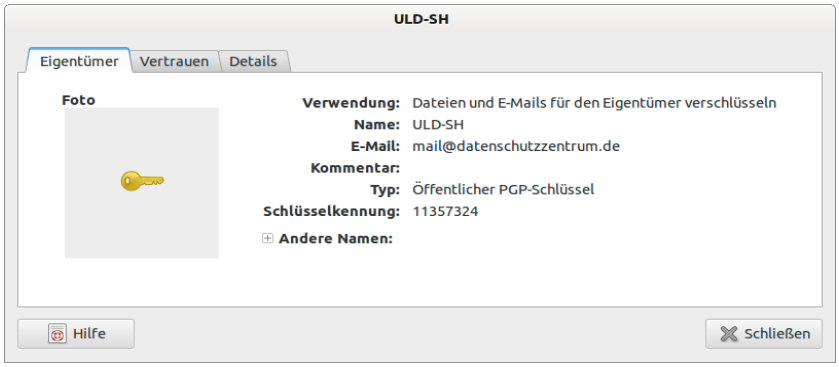

Abbildung 28.2.5.3.1: Eingabe des privaten Schlüssels

Zum Verschlüsseln wird für die *Demonstration* ein *öffentlicher* Schlüssel des Autors benutzt, weil dann auch das Entschlüsseln mit dem zum Schlüsselpaar gehörenden privaten Schlüssel schnell gezeigt werden kann:

Befehl in einer Konsole:

```
hans@linux:~$ gpg --encrypt -a --recipient hans@gambas-buch.de /home/hans/DBT/daten.csv
hans@linux:~$ gpg --encrypt --armor --recipient E06BA316 /home/hans/DBT/daten.csv
```

In der Angabe der EMail-Adresse steckt bereits der Verweis auf den (eigenen) öffentlichen Schlüssel, die Teil des öffentlichen Schlüssels ist! Sie können aber auch die Schlüssel-Kennung direkt eingeben. Die zu verschlüsselnde Datei ist *daten.csv*. Der Inhalt der generierten verschlüsselten Datei liegt im lesbaren ASCII-(Armor-)Format in der Datei daten.csv.asc.

28.2.5.4 Entschlüsseln einer einzelnen Datei

Für das Entschlüsseln der Datei *daten.csv.asc* wird der private Schlüssel des Autors benötigt und diesen besitzt nur einer – der Autor!

Nach dem Ausführen des folgenden Kommandos in einer Konsole wird man zur Eingabe des privaten Schlüssels (Passphrase) aufgefordert:

```
SYNTAX:
hans@linux:~$ gpg --decrypt --output Ziel-Datei-Pfad           Quell-Datei-Pfad
```

```
hans@linux:~$ gpg --decrypt --output /home/hans/DBT/daten.csv /home/hans/DBT/daten.csv.asc

Sie benötigen eine Passphrase, um den geheimen Schlüssel zu entsperren.
Benutzer: »Hans Lehmann (GPGSchlüsselpaar) <hans@gambas-buch.de>«
2048-Bit RSA Schlüssel, ID DE7B82D4, erzeugt 2014-11-13 (Hauptschlüssel-ID E06BA316)

gpg: verschlüsselt mit 2048-Bit RSA Schlüssel, ID DE7B82D4, erzeugt 2014-11-13
      "Hans Lehmann (GPGSchlüsselpaar) <hans@gambas-buch.de>"
hans@linux:~$
```

Abbildung 28.2.5.4.1: Eingabe des privaten Schlüssels

Anschließend findet man die Originaldatei unter /home/hans/DBT/daten.csv.

28.2.5.5 Beispiel 2

Verschlüsseln einer Datei mit dem öffentlichen Schlüssel:

```
hans@linux:~$ gpg --encrypt -a --recipient wer@ist.da /home/hans/DBT2/daten.csv
```

Beim Empfänger wird die Datei mit dem privaten Schlüssel entschlüsselt und die entschlüsselte Datei in einer Text-Datei mit einem *anderen* Dateinamen gesichert → *daten_backup.csv*:

```
Syntax: gpg --decrypt --output Ziel Quelle
```

```
hans@linux:~$ gpg --decrypt --output /home/hans/DBT2/daten_backup.csv /home/hans/DBT2/daten.csv.asc

Sie benötigen einen Passwortsatz, um den geheimen Schlüssel
für Nutzer: "Hans Lehmann (Datensicherheit) <wer@ist.da>" zu entsperren

2048-Bit ELG-E Schlüssel, ID 99028A2D, erstellt 2010-08-19 (Hauptschlüssel-ID 44337DE4)

gpg: verschlüsselt mit 2048-Bit ELG-E Schlüssel, ID 99028A2D, erzeugt 2010-08-19
     "Hans Lehmann (Datensicherheit) <wer@ist.da>"
hans@linux:~$
```

So sieht der (lesbare) Inhalt eines öffentlichen Schlüssels aus:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.11 (GNU/Linux)

mQENBFRkmCUBCADAJ1ZDmTPLdnLdRGcJtPfuPjZJKq/33ESaY/adGy4m2xGmsDeh
QrNGjqpK2WV5izHQxiUkECacjPqsgG7Y4DysKCbV4TB/gu3fWNNdLnIaaUYkDzqN
gSavC3HKqRM4Z0rth5U0nX2CAaTZ8fuRaOcHSne+ZKWQ3xuvMt+C0vZ7ANSZLTz3
2PtpNDQO1KagDC9KDDBUNbXXgpcpylK0PFRldlf/4Haf5aenLvHmLeZM+1AXjI90
Q/yvq5siZTkj3fHcgsnvMylq1eO+Vb+basMAcAVsma//XjiNaTrd8xqx3ssLDf/b
fl55r6i4QcNM1eMns8hcLHQMnVDAK+WA166NABEBAAG0NkhhbnMgTGVobWFubiAo
...
3M+TnKcuPV1TpNBkWwhlmdsLpt0lBQznhwCo5AHZQ03Kl3he9y/xRW303uYR7BAN
Amyw2wlG2PCu7yDrXQ7icpjkyqLrJ1j3SeF7GIvZNGhq5AHqmFaGikoRMvulqSa4
ZtJ8bH2cYTrKP0T5kjoYEuEkHVvD2KollobmFtaGlwEe/x0vI/ZHDGwGmEqcrdAv
H4JNO0kkntWZYGTIDcK6x5Q=
=BolE
-----END PGP PUBLIC KEY BLOCK-----
```