SelfLinux-0.13.1

Grundlagen

Autor: Mike Ashley ()
Formatierung: Matthias Hagedorn (*matthias.hagedorn@selflinux.org*)

Dieses Kapitel führt in die wesentlichen Funktionen des GNU Privacy Guard ein. Hier lernen Sie, wie man Schlüsselpaare erzeugt, Schlüssel austauscht und überprüft, Dokumente verschlüsselt, entschlüsselt und durch digitale Unterschriften authentifiziert.

Wie bereits in Kapitel concepts erwähnt, bedient sich GnuPG eines Public-Key-Verfahrens, um eine sichere Kommunikation zu gewährleisten. In einem solchen System hat jeder Benutzer ein Schlüsselpaar, bestehend aus einem geheimen Schlüssel und einem öffentlichen Schlüssel. Der geheime Schlüssel darf unter keinen Umständen jemand anderem zugänglich sein. Den öffentlichen Schlüssel sollte man für jeden, mit dem man kommunizieren möchte, zugänglich machen.

GnuPG benutzt ein erweitertes Schema, bei dem jeder Benutzer jeweils ein primäres Schlüsselpaar hat und optional weitere untergeordnete Schlüsselpaare haben kann. Das primäre und das untergeordnete Schlüsselpaar werden gebündelt, um die Schlüsselverwaltung zu erleichtern; das Bündel kann vereinfacht als ein Schlüsselpaar betrachtet werden.

Inhaltsverzeichnis

1 Erzeugen eines neuen Schlüsselpaares

Damit Sie GnuPG zum Verschlüsseln, Entschlüsseln oder Signieren einsetzen können, benötigen Sie ein Schlüsselpaar, das aus einem geheimen und einem öffentlichen Schlüssel besteht. Mit der Kommandozeilen-Option `--gen-key` können Sie ein neues primäres Schlüsselpaar erzeugen:

```
user@linux ~/ $ gpg --gen-key

gpg (GnuPG) 1.0.1; Copyright (C) 1999 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.

Bitte wählen Sie, welche Art von Schlüssel Sie möchten:
   (1) DSA und ElGamal (voreingestellt)
   (2) DSA (nur signieren/beglaubigen)
   (4) ElGamal (signieren/beglaubigen und verschlüsseln)
Ihre Auswahl?
```

Mit GnuPG können Sie verschiedene Typen von Schlüsselpaaren erzeugen, doch muß der primäre Schlüssel Unterschriften liefern können. Es gibt daher nur drei Optionen. Option 1 erzeugt wirklich zwei Schlüsselpaare, nämlich ein DSA-Schlüsselpaar, das nur zum Unterschreiben geeignet ist, und außerdem noch ein untergeordnetes ElGamal-Schlüsselpaar für die Verschlüsselung. Option 2 erzeugt nur das DSA-Schlüsselpaar. Option 4 [1] erzeugt ein einzelnes ElGamal-Schlüsselpaar, das sowohl zum Unterzeichnen als auch zum Verschlüsseln verwendbar ist. In allen Fällen ist es möglich, später noch weitere Unterschlüssel für die Verschlüsselung und Unterzeichnung hinzuzufügen. In der Regel sollten Sie hier die Standardoption auswählen.

Als nächstes wählen Sie die Schlüsselgröße. Bei einem DSA-Schlüssel muß diese zwischen 512 und 1024 Bits liegen, ein ElGamal-Schlüssel dagegen kann - zumindest theoretisch - eine beliebige Größe haben. Der GnuPG erfordert es allerdings, dass die Schlüssel nicht kleiner als 768 Bits sind. Wenn Option 1 mit einer Schlüsselgröße von mehr als 1024 Bit gewählt wurde, hat der ElGamal-Schlüssel die verlangte Größe, doch der DSA-Schlüssel wird maximal 1024 Bits haben.

```
Der DSA Schlüssel wird 1024 Bits haben.
Es wird ein neues ELG-E Schlüsselpaar erzeugt.
              kleinste Schlüssellänge ist  768 Bit
              standard Schlüssellänge ist 1024 Bit
      größte sinnvolle Schlüssellänge ist 2048 Bit
Welche Schlüssellänge wünschen Sie? (1024)
```

Je größer der Schlüssel ist, desto sicherer ist er gegen Brute-Force-Angriffe, doch sollte für die meisten Zwecke die Standard-Schlüsselgröße ausreichend sein, da es einfacher wäre, die Verschlüsselung zu umgehen, als sie zu knacken. Außerdem wird mit zunehmender Schlüsselgröße die Ver- und Entschlüsselung langsamer, und auch die Unterschrift wird länger. Einmal festgelegt, kann die Schlüsselgröße nicht nachträglich geändert werden.

Schließlich müssen Sie noch ein Verfallsdatum wählen. Wenn Option 1 gewählt wurde, gilt dieses Verfallsdatum sowohl für die ElGamal- als auch die DSA-Schlüsselpaare.

```
Bitte wählen Sie, wie lange der Schlüssel gültig bleiben soll.
         0 = Schlüssel verfällt nie
      <n>  = Schlüssel verfällt nach n Tagen
      <n>w = Schlüssel verfällt nach n Wochen
      <n>m = Schlüssel verfällt nach n Monaten
      <n>y = Schlüssel verfällt nach n Jahren
Der Schlüssel bleibt wie lange gültig? (0)
```

Für die meisten Fälle reicht ein Schlüssel ohne Verfallsdatum völlig aus. Allerdings sollte man das Verfallsdatum immer sorgfältig auswählen; denn, obwohl es sich auch noch nachträglich ändern läßt, kann es umständlich sein, das geänderte Verfallsdatum allen Ihren Kommunikationspartnern mitzuteilen.

Im nächsten Schritt müssen Sie eine Benutzer-ID (Benutzer-Kennung) angeben. Das dient dazu, den soeben erzeugten Schlüssel einer realen Person zuzuordnen.

```
Sie benötigen eine User-ID, um Ihren Schlüssel eindeutig zu machen; das
Programm baut diese User-ID aus Ihrem echten Namen, einem Kommentar und
Ihrer E-Mail-Adresse in dieser Form auf:
 "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>''

Ihr Name ("Vorname Nachname"):
```

Es wird zunächst nur eine Benutzer-ID erzeugt, doch können Sie später weitere Benutzer-IDs hinzufügen, wenn Sie den Schlüssel in verschiedenen Situationen benutzen wollen, also beispielsweise bei der Arbeit in Ihrer Firma oder für Ihre politische Arbeit. **Die Benutzer-ID sollten Sie mit aller Sorgfalt wählen, da Sie sie später nicht mehr ändern können.**

Damit Ihr geheimer Schlüssel nicht von anderen mißbraucht werden kann, wird er von GnuPG mit einem symmetrischen Verfahren verschlüsselt. Dazu geben Sie ein sogenanntes Mantra (einen Paßwort-Satz) ein, das Sie wiederum jedesmal benötigen, wenn Sie auf Ihren geheimen Schlüssel zugreifen.

```
Sie benötigen ein Mantra, um den geheimen Schlüssel zu schützen.

Geben Sie das Mantra ein:
```

Die Länge des Mantra ist theoretisch unbegrenzt. Sie sollten es mit Sorgfalt auswählen. Unter dem Gesichtspunkt der Sicherheit ist das Mantra einer der schwächsten Punkte im GnuPG (wie auch in anderen Verschlüsselungssystemen mit öffentlichen Schlüsseln), da es Ihr einziger Schutz ist, falls jemand in den Besitz Ihres privaten Schlüssels kommt.

Man sollte für das Mantra keine Wörter aus einem Wörterbuch oder Lexikon nehmen und nicht nur die Buchstaben des Alphabets, sondern auch Sonderzeichen verwenden. Je länger das Mantra ist, desto sicherer ist es, aber andererseits sollten Sie sich das Mantra auch gut merken können; nichts ist fataler als das Mantra auf einem Zettel oder in einer Datei zu notieren. Ein gut gewähltes Mantra ist entscheidend für Ihre Datensicherheit.

Es ist beispielsweise keine gute Idee, einen bekannten Ausspruch oder ein Zitat einer bekannten Persönlichkeit als Mantra zu nehmen. Das würde die Chance erhöhen, das Mantra zu erraten: ein Angreifer könnte einfach den Computer eine Zitatenliste durchprobieren lassen. Am besten denkt man sich einen unsinnigen Satz wie z.B: **Die**

Currywurst schmeckt nach Zimt und Zucker oder **Helmut Kohl ist bekanntermaßen Vegetarier** aus. Ihrer Phantasie sind hierbei keine Grenzen gesetzt. Wenn Sie auch noch ein paar Rechtschreibfehler und Sonderzeichen einbauen, ist ein Wörterbuchangriff praktisch unmöglich: **Dat Körriwurst schmöckt nach #imt und #ucker**. **Benutzen Sie auch auf keinen Fall eines der soeben aufgeführten Beispiele!!**.

1.1 Erzeugen einer Widerrufurkunde

Nach dem Erzeugen Ihres Schlüsselpaars sollten Sie sofort mit der Option`--gen-revoke` eine Widerrufurkunde für Ihre Schlüssel erzeugen. Wenn Sie Ihr Mantra vergessen oder wenn Ihr privater Schlüssel kompromittiert oder verloren gegangen ist, können Sie mit dieser Widerrufurkunde andere davon in Kenntnis setzen, dass der dazugehörige öffentliche Schlüssel nicht mehr benutzt werden sollte. Ein zurückgerufener öffentlicher Schlüssel kann noch benutzt werden, um Unterschriften zu prüfen, die Sie vor dem Widerruf abgegeben haben, er kann jedoch nicht benutzt werden, um künftige Mitteilungen an Sie zu verschlüsseln. Vorausgesetzt, Sie haben noch Zugang zu Ihrem widerrufenen geheimen Schlüssel, so können Sie selbstverständlich noch Daten entschlüsseln, die vor dem Widerruf für Sie verschlüsselt worden sind.

```
user@linux ~/ $ gpg --output revoke.asc --gen-revoke mykey

          [...]
```

wobei `mykey` entweder die Schlüssel-ID Ihres ersten Schlüsselpaares oder irgendein Teil einer dazugehörigen Benutzer-ID ist. Die erzeugte Widerrufurkunde wird in die Datei revoke.asc, bzw., wenn man die Option`--output` wegläßt, auf die Standard-Ausgabe geschrieben. Da die Widerrufurkunde kurz ist, ist es kein Problem, eine ausgedruckte Kopie der Widerrufurkunde irgendwo sicher aufzubewahren, z.B. in Ihrem Bankschließfach. Die Widerrufurkunde sollten Sie aber auf keinen Fall an Stellen aufbewahren, zu denen andere Personen Zugang haben, da im Prinzip jeder die Widerrufurkunde veröffentlichen und so den entsprechenden Schlüssel nutzlos machen könnte.

2 Austauschen von Schlüsseln

Um mit anderen zu kommunizieren, müssen Sie untereinander Ihre öffentlichen Schlüssel austauschen. Zum Auflisten der Schlüssel in Ihrem öffentlichen Schlüsselbund verwenden Sie die Kommandozeilen-Option `--list-keys`.

```
user@linux ~/ $ gpg --list-keys

/users/alice/.gnupg/pubring.gpg
-----------------------------------
pub  1024D/FB5797A9 2000-06-06 Alice (Rechtsanwältin) <alice@cyb.org>
sub  1024g/C8B3998F 2000-06-06
```

2.1 Exportieren eines öffentlichen Schlüssels

Um jemandem Ihren öffentlichen Schlüssel zu schicken, müssen Sie diesen zunächst exportieren. Hierzu benutzt man die Kommandozeilen-Option `--export`. Zur Indentifikation des zu exportierenden öffentlichen Schlüssels dient entweder die Schlüssel-ID oder irgendein Teil der Benutzer-ID.

```
user@linux ~/ $ gpg --output alice.gpg --export alice@cyb.org
```

Der Schlüssel wird in einem binären Format exportiert, doch kann dies unerwünscht sein, wenn Sie den Schlüssel per E-Mail verschicken oder auf einer WWW-Seite veröffentlichen wollen. GnuPG unterstützt daher die Kommandozeilen-Option `--armor` (**) die bewirkt, dass der Output im ASCII-Format ausgegeben wird. (Im Allgemeinen kann jeder Output von GnuPG - beispielsweise Schlüssel, verschlüsselte Dokumente oder Unterschriften - im ASCII-Format dargestellt werden, indem man die `--armor`-Option hinzufügt.)

```
user@linux ~/ $ gpg --armor --export alice@cyb.org

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.0.1 (GNU/Linux)
Comment: For info see http://www.gnupg.org

[...]
-----END PGP PUBLIC KEY BLOCK-----
```

2.2 Importieren eines öffentlichen Schlüssels

Ein öffentlicher Schlüssel kann zu Ihrem öffentlichen Schlüsselbund hinzugefügt werden, und zwar mit folgender Option: `--import`

```
user@linux ~/ $ gpg --import blake.gpg

gpg: Schlüssel B2690E6F: Öffentlicher Schlüssel importiert
gpg: Anzahl insgesamt bearbeiteter Schlüssel: 1
gpg:            importiert: 1

user@linux ~/ $ gpg --list-keys
```

```
-----------------------------------
pub  1024D/FB5797A9 2000-06-06 Alice (Rechtsanwältin) <alice@cyb.org>
sub  1024g/C8B3998F 2000-06-06

pub  1024D/B2690E6F 2000-06-06 Blake (Staatsanwalt) <blake@cyb.org>
sub  1024g/F251B862 2000-06-06
```

Wenn ein Schlüssel einmal importiert ist, sollte er auf Authentizität überprüft werden. GnuPG arbeitet mit einem wirksamen und flexiblen Vertrauensmodell, bei dem Sie nicht jeden Schlüssel persönlich zu authentifizieren brauchen, den Sie importieren. Einige Schlüssel können dies jedoch erfordern. Ein Schlüssel wird dadurch authentifiziert, dass Sie den Fingerabdruck des Schlüssels überpüfen und dann den Schlüssel unterschreiben, um seine Gültigkeit zu bestätigen. Der Fingerabdruck eines Schlüssels kann schnell mit der Befehlszeilen-Option `--fingerprint` geprüft werden, um aber den Schlüssel zu bestätigen, müssen Sie ihn editieren.

```
user@linux ~/ $ gpg --edit-key blake@cyb.org

pub  1024D/B2690E6F  created: 2000-06-06 expires: never      trust: -/q
sub  1024g/F251B862  created: 2000-06-06 expires: never
(1)  Blake (Staatsanwalt) <blake@cyb.org>

user@linux ~/ $ fpr

pub  1024D/B2690E6F 2000-06-06 Blake (Staatsanwalt) <blake@cyb.org>
             Fingerprint: 6A51 852C 7491 95B5 C5F0  731C 141F C008 B269
0E6F
```

Um den Fingerabdruck zu überprüfen, müssen Sie den Eigentümer des Schlüssels kontaktieren und die Fingerabdrücke vergleichen. Sie können persönlich oder per Telefon mit ihm sprechen oder auf beliebigem anderen Wege kommunizieren, solange nur garantiert ist, dass es sich um den rechtmäßigen Eigentümer handelt. Stimmen beide Fingerabdrücke überein, dann können Sie sicher sein, daß Sie eine echte Kopie des öffentlichen Schlüssels haben.

Nach dem Prüfen des Fingerabdrucks können Sie den Schlüssel unterschreiben, um ihn zu authentifizieren. Da die Schlüsselüberprüfung ein Schwachpunkt in der Kryptographie mit öffentlichem Schlüssel ist, sollten Sie **äußerste Sorgfalt** walten lassen und den Fingerabdruck eines Schlüssels **immer** gemeinsam mit dem Eigentümer prüfen, bevor Sie den Schlüssel unterschreiben.

```
user@linux ~/ $ sign

pub  1024D/B2690E6F  created: 2000-06-06 expires: never      trust: -/q
             Fingerprint: 6A51 852C 7491 95B5 C5F0  731C 141F C008 B269
0E6F

     Blake (Staatsanwalt) <blake@cyb.org>

Sind Sie wirklich sicher, dass Sie vorstehenden Schlüssel mit Ihrem
Schlüssel beglaubigen wollen: "Alice (Rechtsanwältin) <alice@cyb.org>"

Wirklich unterschreiben?
```

Sie können sich jederzeit vergewissern, welche Unterschrift Sie hinzugefügt haben. Jede Benutzer-ID auf dem Schlüssel hat dann sowohl eine oder mehrere Eigenbeglaubigungen als auch eine Unterschrift von jedem Benutzer, der den Schlüssel authentifiziert hat.

```
user@linux ~/ $ check

uid  Blake (Staatsanwalt) <blake@cyb.org>
sig!       B2690E6F 2000-06-06   [Eigenbeglaubigung]
sig!       FB5797A9 2000-06-06   Alice (Rechtsanwältin) <alice@cyb.org>
```

3 Ver- und Entschlüsseln von Dokumenten

Der öffentliche und der geheime Schlüssel haben jeweils eine spezifische Aufgabe beim Ver- und Entschlüsseln von Dokumenten. Das Public-Key-Verfahren kann man sich wie einen offenen Safe vorstellen. Wenn jemand ein Dokument unter Benutzung eines öffentlichen Schlüssels verschlüsselt, wird dieses Dokument in den Safe gelegt, der Safe geschlossen und das Kombinationsschloß mehrmals verdreht. Der entsprechende geheime Schlüssel ist die Kombination, mit der man den Safe wieder öffnen und das Dokument wieder herausholen kann. Mit anderen Worten, es kann nur die Person, die den geheimen Schlüssel hat, auf ein Dokument zugreifen, das unter Benutzung des dazugehörigen öffentlichen Schlüssels verschlüsselt worden ist.

Das Verfahren für das Ver- und Entschlüsseln von Dokumenten ist bei diesem Modell einfach: eine Nachricht an Alice verschlüsseln Sie unter Verwendung von Alices öffentlichem Schlüssel, und diese entschlüsselt sie mit ihrem geheimen Schlüssel. Umgekehrt geht es genauso: Alice verschlüsselt eine Nachricht an Sie mit Ihrem öffentlichen Schlüssel, welche Sie dann mit Ihrem geheimen Schlüssel entschlüsseln können.

Um ein Dokument zu verschlüsseln, benutzt man die Option `--encrypt`. Dazu müssen Sie die öffentlichen Schlüssel der vorgesehenen Empfänger haben. Sollten Sie auf der Kommandozeile den Namen der zu verschlüsselnden Datei nicht angeben, werden die zu verschlüsselnden Daten von der Standard-Eingabe gelesen. Das verschlüsselte Resultat wird auf die Standard-Ausgabe oder in die Datei, die durch die Option `--output` spezifiziert ist, geschrieben. Das Dokument wird darüberhinaus auch noch komprimiert.

```
user@linux ~/ $ gpg --output doc.gpg --encrypt --recipient blake@cyb.org
doc
```

Mit der Option `--recipient` wird der öffentliche Schlüssel spezifiziert, mit dem das Dokument verschlüsselt werden soll. Entschlüsseln läßt sich das so verschlüsselte Dokument jedoch nur von jemandem mit dem dazugehörigen geheimen Schlüssel. Das bedeutet konsequenterweise aber auch, dass Sie selbst ein so verschlüsseltes Dokument nur wieder entschlüsseln können, wenn Sie Ihren eigenen öffentlichen Schlüssel in die Empfängerliste aufgenommen haben.

Zum Entschlüsseln einer Nachricht wird die Option `--decrypt` benutzt. Sie benötigen dazu den geheimen Schlüssel, für den die Nachricht verschlüsselt wurde und das Mantra, mit dem der geheime Schlüssel geschützt ist.

```
user@linux ~/ $ gpg --output doc --decrypt doc.gpg

Sie benötigen ein Mantra, um den geheimen Schlüssel zu entsperren.
Benutzer: "Blake (Staatsanwalt) <blake@cyb.org>"
1024-Bit ELG-E Schlüssel, ID F251B862, erzeugt 2000-06-06
(Hauptschlüssel-ID B2
690E6F)

Geben Sie das Mantra ein:
```

Mit GnuPG können Sie aber auch ohne Anwendung eines Public-Key-Verfahrens Dokumente verschlüsseln und stattdessen ein symmetrisches Verfahren benutzen. Der Schlüssel für die symmetrische Verschlüsselung wird aus einem Paßwort - besser noch, einem Paßwort-Satz - generiert, das **auf gar keinen Fall** dem Mantra zum Schutz Ihres privaten Schlüssels entsprechen sollte. Je länger das gewählte Paßwort ist, desto sicherer ist der Schlüssel. Wenn Sie diesen symmetrischen Schlüssel an jemanden weitergeben, sollten Sie dazu einen sicheren Weg wählen. Ein Dokument läßt sich so durch Benutzung der Option `--symmetric` verschlüsseln.

```
user@linux ~/ $ gpg --output doc.gpg --symmetric doc
 Geben Sie das Mantra ein:
```

Symmetrische Verfahren empfehlen sich beispielsweise, wenn Sie die verschlüsselten Daten nicht weiter geben möchten, das Problem der Paßwortübergabe also entfällt. Ein mögliches Anwendungsbeispiel wäre, dass Sie alte E-Mails oder alte Datensätze aus Ihrer Umsatzstatisk auf ihrer Festplatte oder einer CDROM archivieren und vor fremden Zugriffen schützen möchten. Oder Sie können auch ganze Verzeichnisse oder Festplatten verschlüsseln.

4 Digitale Signaturen

Eine digitale Unterschrift oder Signatur ist am ehesten mit einem Siegel zu vergleichen. Mit dem Siegel wird die Integrität eines Dokumentes bestätigt, das sich in einem Umschlag befindet, und ermöglicht, dass sich eine nachträgliche Manipulation feststellen läßt. Wenn das Dokument nachfolgend in irgendeiner Weise verändert wird, ergibt die Prüfung der Signatur ein negatives Ergebnis. Außerdem ermöglicht die Signatur eine zweifelsfreie Zuordnung des Absenders. Eine digitale Unterschrift kann so demselben Zweck wie eine handgeschriebene Unterschrift dienen mit dem zusätzlichen Vorteil, eine Handhabe gegen Verfälschung zu bieten. Die GnuPG-Quelltextdistribution ist [eg] so unterschrieben, dass die Benutzer nachprüfen können, dass der Quelltext nachträglich nicht verändert worden ist und auch wirklich vom GnuPG-Team stammt.

Die rechtliche Verbindlichkeit von digitalen Unterschriften ist von Land zu Land verschieden. In Deutschland ist das Signaturgesetz augenblicklich einer Novellierung unterworfen. Weitere Informationen und Quellenverweise finden Sie in Kapitel 4.

Bei der Erzeugung und Prüfung von Unterschriften benutzt man das öffentlich/geheime Schlüsselpaar anders als bei der Ver- und Entschlüsselung. Die Unterschrift wird hier mit dem geheimen Schlüssel des Unterzeichnenden erzeugt und dann jeweils mit dem entsprechenden öffentlichen Schlüssel geprüft. So würde beispielsweise Alice ihren geheimen Schlüssel benutzen, um ihren letzten Beitrag für eine Zeitschrift zu signieren. Der Redakteur, der Alices Artikel bearbeitet, benutzt dann ihren öffentlichen Schlüssel, um die Unterschrift zu prüfen und so sicherzustellen, daß der Beitrag wirklich von Alice selbst stammt und auch nicht nachträglich verändert worden ist.

Als Konsequenz aus der Verwendung digitaler Signaturen ergibt sich, dass sich kaum abstreiten läßt, dass man eine digitale Unterschrift geleistet hat, da dies ja bedeuten würde, dass der geheime Schlüssel kompromittiert wurde.

Die Kommandozeilen-Option `--sign` wird zum Erzeugen einer digitalen Unterschrift verwendet. Mit der Option `--output` legen Sie fest, in welche Datei das signierte Dokument geschrieben wird.

```
user@linux ~/ $ gpg --output doc.sig --sign doc

Sie benötigen ein Mantra, um den geheimen Schlüssel zu entsperren.
Benutzer: "Alice (Rechtsanwältin) <alice@cyb.org>"
1024-bit DSA Schlüssel, 1024D/FB5797A9, erzeugt 2000-06-06

Geben Sie das Mantra ein:
```

Das Dokument wird vor dem Unterschreiben komprimiert und die Ausgabe erfolgt im binären Format.

Haben Sie ein unterschriebenes Dokument erhalten, können Sie die Unterschrift prüfen, und zwar optional ohne oder mit Entnahme des unterschriebenen Originaldokumentes. Zur bloßen Überprüfung der Unterschrift benutzen Sie die Option `--verify`. Wenn Sie außerdem das unterzeichnete Dokument entnehmen wollen, verwenden Sie die Option `--decrypt`.

```
user@linux ~/ $ gpg --output doc --decrypt doc.sig

gpg: Unterschrift vom Die 06 Jun 2000 17:19:52 CEST, DSA Schlüssel ID
FB5797A9
gpg: Korrekte Unterschrift von "Alice (Rechtsanwältin) <alice@cyb.org>"
```

4.1 Dokumente mit Klartextsignatur

In Fällen, in denen es unerwünscht ist, das Dokument beim Unterschreiben zu komprimieren, benutzt man die Option `--clearsign`. Das bewirkt, dass eine in ASCII dargestellte Signatur das Dokument wie ein Briefumschlag umgibt, das Dokument an sich aber nicht verändert wird. Der Vorteil dieses Verfahrens ist, dass der Empfänger das Dokument auch ohne Prüfung der Signatur lesen kann.

```
user@linux ~/ $ gpg --clearsign doc

Sie benötigen ein Mantra, um den geheimen Schlüssel zu entsperren.
Benutzer: "Alice (Rechtsanwältin) <alice@cyb.org>"
1024-Bit DSA Schlüssel, ID FB5797A9, erzeugt 2000-06-06

Geben Sie das Mantra ein:
```

GnuPG markiert dann im Klartext den Anfang des signierten Dokuments und hängt am Ende einen Block mit der eigentlichen OpenPGP-Signatur an.

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Hier steht irgend ein
von GnuPG signierter Text
[...]
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.0.1 (GNU/Linux)
Comment: For info see http://www.gnupg.org

iD8DBQE5Pf40WyoKbftXl6kRAsWJAJ4hj7FzPX8M9MWZav9u6yjbHXWGKwCfSiKA
wTaJ/lfY1ETv3R/uJrtGTbI=
=BDOH
-----END PGP SIGNATURE-----
```

4.2 Abgetrennte Signatur

Der Nachteil bei signierten Dokumenten ist, dass der Empfänger das Originaldokument aus der unterschriebenen Version erst wiederherstellen muß bzw. bei einem im Klartext unterschriebenen Dokument dieses gegebenenfalls noch editieren muß. Deshalb gibt es als Drittes noch die Möglichkeit, Dokumente mit abgetrennter Unterschrift zu signieren. Dazu verwendet man die Option `--detach-sig`. Die Signatur wird dann in einer separaten Datei abgelegt. Das eigentliche Dokument bleibt unverändert.

```
user@linux ~/ $ gpg --output doc.sig --detach-sig doc

Sie benötigen ein Mantra, um den geheimen Schlüssel zu entsperren.
Benutzer: "Alice (Rechtsanwältin) <alice@cyb.org>"
1024-Bit DSA Schlüssel, ID FB5797A9, erzeugt 2000-06-06

Geben Sie das Mantra ein:
```

Um die Signatur zu prüfen, benötigen Sie sowohl das eigentliche Dokument als auch die abgetrennte Unterschrift. Die Option `--verify` kann zum Prüfen der Signatur benutzt werden.

```
user@linux ~/ $ gpg --verify doc.sig doc

gpg: Unterschrift vom Die 06 Jun 2000 17:34:43 CEST, DSA Schlüssel ID
FB5797A9
gpg: Korrekte Unterschrift von "Alice (Rechtsanwältin) <alice@cyb.org>"
```

5 Fußnoten

5.1 Option 4

Mit der Option 3 läßt sich ein ElGamal-Schlüsselpaar erzeugen, mit dem Sie keine Unterschriften leisten können.

5.2 Kommandozeilen-Option --armor

Viele Kommandozeilen-Optionen, die häufig benutzt werden, können auch in einer Konfigurationsdatei definiert werden.