SelfLinux-0.10.0

Schlüsselverwaltung

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

Schlüsselverfälschungen sind ein nicht zu unterschätzender Unsicherheitsfaktor bei der Public-Key-Kryptographie. Ein Angreifer kann beispielsweise die Schlüsselbunde eines Benutzers manipulieren oder sich einen öffentlichen Schlüssel mit einer vorgetäuschten Identität erzeugen und ihn an andere zum Herunterladen und Benutzen schicken. Wenn z.B. Chloe unbemerkt die Nachrichten, welche Alice an Blake sendet, lesen will, dann könnte sie folgendermaßen vorgehen: zuerst erzeugt sie ein neues Schlüsselpaar mit einer gefälschten Benutzer-ID. Dann ersetzt sie Alices Kopie von Blakes öffentlichem Schlüssel durch den neuen Schlüssel. Anschließend fängt sie die Nachrichten ab, die Alice an Blake sendet. Diese Nachrichten kann sie dann mit dem neuen geheimen Schlüssel entschlüsseln. Dann verschlüsselt sie die Nachricht wieder, aber diesmal mit dem echten öffentlichen Schlüssel von Blake und schickt sie weiter an Blake. Chloe kann jetzt - ohne dass jemand etwas bemerkt - alle von Alice an Blake geschickten Nachrichten mitlesen.

Eine gute Schlüsselverwaltung ist entscheidend für die Integrität Ihrer eigenen Schlüsselbunde, wie auch der Schlüsselbunde anderer Benutzer. Der Kern der Schlüsselverwaltung von GnuPG ist das **Signieren von Schlüsseln** und verfolgt zwei Hauptzwecke: es erlaubt Ihnen, Verfälschungen an Ihrem Schlüsselbund zu entdecken, und es ermöglicht Ihnen, die Zugehörigkeit eines Schlüssels zu der von der jeweiligen Benutzer-ID genannten Person zu überprüfen. Schlüsselunterschriften werden in einem **Web of Trust** genannten Schema benutzt, um die Authentisierung auch auf Schlüssel auszudehnen, die nicht direkt von Ihnen selbst, sondern von anderen Personen, denen Sie zutrauen, Schlüssel nur nach sorgfältiger Prüfung zu signieren, signiert worden sind. Durch eine gewissenhafte Schlüsselverwaltung können Sie Schlüsselverfälschungen als einen praktischen Angriff auf ihre sichere und vertrauliche Kommunikation abwehren.

Inhaltsverzeichnis

1 Verwaltung Ihres Schlüsselpaares

Ein Schlüsselpaar besteht aus einem öffentlichen und einem geheimen Schlüssel und einem Satz von Benutzer-IDs, um die Schlüssel einer realen Person zuzuordnen. Jeder dieser Bestandteile enthält Informationen über sich selbst. Bei einem öffentlichen Schlüssel sind dies seine ID sowie Angaben darüber, wann er erzeugt worden ist, wann seine Gültigkeit abläuft usw. Bei der Benutzer-ID sind das der Name der realen Person, die durch die ID identifiziert wird, eine optionale Bemerkung sowie eine E-mail-Adresse. Der geheime Schlüssel enthält dagegen keine Informationen über die Benutzer-ID.

Wenn Sie Informationen über ein Schlüsselpaar sehen möchten, dann rufen Sie am besten mit der Kommandozeilen-Option `--edit-key`> den Schlüsseleditor auf. Zum Beispiel:

```
user@linux ~/ # gpg --edit-key chloe@cyb.org

Geheimer Schlüssel ist vorhanden.

pub  1024D/1B087D04  created: 2000-06-07 expires: never      trust: -/u
sub  2048g/6A3E902A  created: 2000-06-07 expires: never
sub  1792G/7D5D4DAE  created: 2000-06-07 expires: 2002-06-07
sub   960D/C0A27DBE  created: 2000-06-07 expires: 2002-06-07
(1)  Chloe (Journalistin) <chloe@cyb.org>
(2)  Chloe (Freie Autorin) <chloe@tel.net>


Befehl>
```

Zusammen mit dem öffentlichen Schlüssel wird angezeigt, ob der geheime Schlüssel verfügbar ist oder nicht. Alle Informationen über die Bestandteile des öffentlichen Schlüssels werden dann aufgelistet. Die erste Spalte gibt den Typ des Schlüssels an. Das Schlüsselwort `pub` identifiziert den öffentlichen Hauptschlüssel und das Schlüsselwort `sub` identifiziert einen untergeordneten öffentlichen Schlüssel (Subkey). Die zweite Spalte gibt Länge, Typ und ID des Schlüssels an. Dabei steht `D` für DSA-Schlüssel, `g` für einen nur zur Verschlüsselung geeigneten ElGamal-Schlüssel und `G` für einen ElGamal-Schlüssel, der sowohl zur Verschlüsselung als auch zum Unterschreiben verwendet werden kann. Das Datum der Erzeugung und das Verfallsdatum wird in den Spalten drei und vier angegeben. Die Benutzer-IDs werden nach den Schlüsseln angegeben.

Es stehen noch weitere Befehle zu Verfügung, um zusätzliche Informationen über die Schlüssel zu erhalten. Der Befehl `toggle` schaltet zwischen den öffentlichen und den geheimen Komponenten eines Schlüsselpaares um, wenn tatsächlich beides zur Verfügung steht.

```
user@linux ~/ # toggle

sec  1024D/1B087D04  created: 2000-06-07 expires: never
sbb  2048g/6A3E902A  created: 2000-06-07 expires: never
sbb  1792G/7D5D4DAE  created: 2000-06-07 expires: 2002-06-07
sbb   960D/C0A27DBE  created: 2000-06-07 expires: 2002-06-07
(1)  Chloe (Journalistin) <chloe@cyb.org>
(2)  Chloe (Freie Autorin) <chloe@tel.net>
```

Die Information ist ähnlich der Auflistung für die Komponente des öffentlichen Schlüssels. Das Schlüsselwort `sec` identifiziert den geheimen Hauptschlüssel und das Schlüsselwort `ssb` identifiziert die geheimen Subkeys. Die Benutzer-IDs vom öffentlichen Schlüssel werden der Bequemlichkeit halber auch aufgelistet.

1.1 Schlüssel-Integrität

Wenn Sie Ihren öffentlichen Schlüssel weitergeben, so geben Sie damit die öffentlichen Komponenten Ihres Hauptschlüssels und Ihrer Subkeys ebenso wie Ihre Benutzer-IDs weiter. Wenn Sie diese Informationen jedoch ungeschützt weitergeben, so besteht ein Sicherheitsrisiko, da es für einen potentiellen Angreifer möglich ist, den Schlüssel zu verfälschen. Der öffentliche Schlüssel kann durch Hinzufügen oder Ersetzen von Schlüsseln oder von Benutzer-IDs modifiziert werden. Der Angreifer könnte beispielsweise durch Verfälschen der E-Mail-Adresse einer Benutzer-ID die E-Mail an sich selbst umleiten. Durch Veränderung der öffentlichen Schlüssel wäre der Angreifer auch in der Lage, die zu ihm umgeleiteten Nachrichten zu entschlüsseln.

Die Benutzung digitaler Signaturen ist die Lösung für dieses Problem. Indem man den öffentlichen Schlüssel sowie die Benutzer-IDs mit seinem geheimen Schlüssel unterzeichnet, lassen sich Verfälschungen daran leicht feststellen. Dieser Vorgang wird Eigenbeglaubigung genannt; ein öffentlicher Schlüssel, der eigenbeglaubigte Benutzer-IDs enthält, wird `Zertifikat` genannt.

Ein Beispiel: Chloe hat zwei Benutzer-IDs und drei untergeordnete öffentliche Schlüssel bzw. Subkeys. Die Unterschriften auf den Benutzer-IDs können mit dem Befehl `check` im Schlüsseleditior geprüft werden.

```
user@linux ~/ # gpg --edit-key chloe

geheimer Schlüssel ist vorhanden.

pub  1024D/1B087D04  created: 2000-06-07 expires: never      trust: -/u
sub  2048g/6A3E902A  created: 2000-06-07 expires: never
sub  1792G/7D5D4DAE  created: 2000-06-07 expires: 2002-06-07
sub   960D/C0A27DBE  created: 2000-06-07 expires: 2002-06-07
(1)  Chloe (Journalistin) <chloe@cyb.org>
(2)  Chloe (Freie Autorin) <chloe@tel.net>


Befehl> check

uid  Chloe (Journalistin) <chloe@cyb.org>
sig!       1B087D04 2000-06-07   [Eigenbeglaubigung]
uid  Chloe (Freie Autorin) <chloe@tel.net>
sig!       1B087D04 2000-06-07   [Eigenbeglaubigung]
```

Wie erwartet, wird für jede Unterschrift der primäre Schlüssel mit der Schlüssel-ID `0x26B6AAE1` genommen. Die Eigenbeglaubigungen auf den Subkeys sind in dem öffentlichen Schlüssel enthalten, doch werden sie vom Schlüsseleditor nicht gezeigt.

1.2 Editieren von Schlüsseln

Zu Ihrem ursprünglichen Schlüsselpaar können Sie später sowohl neue Subkeys als auch neue Benutzer-IDs hinzufügen. Eine neue Benutzer-ID wird durch Verwendung des Befehls `adduid` erzeugt. Dabei werden Sie wieder nach Ihrem wirklichem Namen, E-Mail-Adresse und einer optionalen Bemerkung gefragt. Ein Subkey wird durch Verwendung des Befehls `addkey` hinzugefügt und kann von beliebigem Typ sein. Das ist so ähnlich, wie Sie es vom Erzeugen Ihres anfänglichen Schlüsselpaares kennen. Wenn Sie einen neuen Subkey oder eine neue Benutzer-ID erzeugen, so werden diese mit Ihrem geheimen Schlüssel eigenbeglaubigt; deshalb müssen Sie auch Ihr Mantra eingeben, wenn der Schlüssel erzeugt wird.

Zusätzliche Benutzer-IDs sind nützlich, wenn Sie für verschiedene Zwecke verschiedene IDs benötigen. So

wollen Sie vielleicht eine Benutzer-ID für Ihre Arbeit, eine für Ihre politische Tätigkeit und eine weitere für private Korrespondenz haben. Ihre Mitarbeiter und Geschäftspartner, Politische Mitstreiter und Freunde werden Sie dann jeweils unter einer anderen ID kennen.

Zusätzliche Subkeys sind ebenfalls nützlich. Die zu Ihrem primären öffentlichen Schlüssel gehörigen Benutzer-IDs werden von den Leuten, mit denen Sie kommunizieren, authentisiert, deshalb erfordert eine Änderung des primären Schlüssels eine nochmalige Bestätigung. Wenn Sie mit vielen Leuten kommunizieren, kann dies schwierig und zeitaufwendig sein. Andererseits ist es gut, von Zeit zu Zeit die Subkeys für die Verschlüsselung zu ändern. Wenn ein Schlüssel kompromittiert wurde, ist die Sicherheit aller mit diesem Schlüssel verschlüsselten Daten gefährdet. Durch das Ändern der Schlüssel erreichen Sie jedoch, dass in der Zukunft zu verschlüsselnde Daten nicht auch noch gefährdet werden.

Subkeys und Benutzer-IDs können auch gelöscht werden. Dazu müssen Sie diese zunächst im Schlüsseleditor auswählen, indem Sie die Befehle `key` bzw. `uid` benutzen. So wählt beispielsweise der Befehl key 2 den zweiten Subkey aus; ein nochmaliger Aufruf des Befehls key 2 macht diese Auswahl wieder rückgängig. Wird key ohne Argument aufgerufen, wird die komplette Auswahl an Subkeys wieder aufgehoben. Das gleiche gilt für den Befehl uid. Wenn Sie die zu löschenden Benutzer-IDs ausgewählt haben, werden diese mit dem Befehl `deluid` aus Ihrem Schlüssel entfernt. Ebenso löscht der Befehl `delkey` alle ausgewählten Subkeys aus Ihren öffentlichen und geheimen Schlüsseln.

Für die lokale Schlüsselverwaltung ist das Löschen von Schlüssel-Komponenten ein geeignetes Mittel, um die öffentlichen Schlüssel anderer von unnötigem Ballast frei zu halten. Hingegen sollten Sie normalerweise keine Benutzer-IDs und Subkeys von Ihrem eigenen Schlüssel entfernen, da Sie so die Weiterverbreitung dieses Schlüssels verkomplizieren. Wenn ein anderer GnuPG-Benutzer Ihren aktuellen öffentlichen Schlüssel importiert, wird dieser standardmäßig mit dessen alter Kopie Ihres öffentlichen Schlüssels zusammengeführt. Dadurch werden effektiv alle Komponenten wieder hergestellt, die Sie gelöscht haben. Um den Schlüssel wirklich zu aktualisieren, müßte der Benutzer zuerst die alte Version Ihres Schlüssels löschen und dann die neue Version importieren. Dies bringt eine zusätzliche Belastung für Ihre Kommunikationspartner mit sich. Es ist daher auch keine gute Idee, Ihren aktualisierten Schlüssel zu einem Key-Server zu schicken. Zum Aktualisieren Ihres eigenen Schlüssels ist es folglich besser, die jeweiligen Schlüsselkomponenten zu widerrufen, statt sie zu löschen.

1.3 Widerrufen von Schlüssel-Komponenten

Um einen Subkey zu widerrufen, wählen Sie ihn im Schlüsseleditor aus, dann können Sie ihn mit dem Befehl `revkey` widerrufen. Der Schlüssel wird dadurch widerrufen, dass man dem Schlüssel eine Widerruf-Unterschrift hinzufügt. Anders als bei der Kommandozeilen-Option `--gen-revoke` tritt der Widerruf sofort in Kraft.

```
key 2


pub  1024D/1B087D04  created: 2000-06-07 expires: never      trust: -/u
sub  2048g/6A3E902A  created: 2000-06-07 expires: never
sub* 1792G/7D5D4DAE  created: 2000-06-07 expires: 2002-06-07
sub   960D/6E82436B  created: 2000-06-07 expires: 2002-06-07
(1)  Chloe (Journalistin) <chloe@cyb.org>
(2)  Chloe (Freie Autorin) <chloe@tel.net>


Befehl> revkey

Möchten Sie diesen Schlüssel wirklich wiederrufen? j
```

```
Sie benötigen ein Mantra, um den geheimen Schlüssel zu entsperren.
Benutzer: "Chloe (Journalistin) <chloe@cyb.org>"
1024-Bit DSA Schlüssel, ID 1B087D04, erzeugt 2000-06-07


pub  1024D/1B087D04  created: 2000-06-07 expires: never      trust: -/u
sub  2048g/6A3E902A  created: 2000-06-07 expires: never
sub  1792G/7D5D4DAE  created: 2000-06-07 expires: 2002-06-07
rev! subkey has been revoked: 2000-06-07
sub   960D/6E82436B  created: 2000-06-07 expires: 2002-06-07
(1)  Chloe (Journalistin) <chloe@cyb.org>
(2)  Chloe (Freie Autorin) <chloe@tel.net>
```

Beim Widerrufen einer Benutzer-ID wird anders verfahren. Durch Unterschriften auf einer Benutzer-ID wird bestätigt, dass der Eigentümer des Schlüssels tatsächlich identisch mit der in der Benutzer-ID genannten Person ist. In der Theorie beschreibt eine Benutzer-ID eine Person für immer, da diese Person sich nie ändert. In der Praxis können sich jedoch Elemente der Benutzer-ID, so z.B. die E-Mail-Adresse oder eine Bemerkung, mit der Zeit verändern und so die Benutzer-ID unbrauchbar machen.

Die Spezifikation von OpenPGP unterstützt den Widerruf einer Benutzer-ID nicht. Man kann sich aber dadurch helfen, dass man seine Eigenbeglaubigung für die entsprechende Benutzer-ID widerruft. Aus den **zuvor** beschriebenen Sicherheitsgründen werden die Korrespondenzpartner keiner Benutzer-ID ohne gültige Eigenbeglaubigung trauen, GnuPG lehnt den Import eines solchen Schlüssels sogar gänzlich ab.

Eine Unterschrift wird unter Verwendung des Befehls `revsig`. widerrufen. Da Sie eine beliebige Zahl von Benutzer-IDs unterschrieben haben können, verlangt der Schlüsseleditor von Ihnen für jede Unterschrift eine Entscheidung, ob sie widerrufen werden soll oder nicht.

```
Befehl> revsig

Sie haben folgende User-IDs beglaubigt:
     Chloe (Journalistin) <chloe@cyb.org>
   beglaubigt durch 1B087D04 um 2000-06-07
   beglaubigt durch 1B087D04 um 2000-06-07
User-ID: ``Chloe (Journalistin) <chloe@cyb.org>''
unterschrieben mit Ihrem Schlüssel 1B087D04 um 2000-06-07
Ein Widerrufszertifikat für diese Unterschrift erzeugen (j/N)n
User-ID: ``Chloe (Freie Autorin) <chloe@tel.net>''
unterschrieben mit Ihrem Schlüssel 1B087D04 um 2000-06-07
Ein Widerrufszertifikat für diese Unterschrift erzeugen (j/N)j
Es werden nun folgende Beglaubigungen entfernt:
     Chloe (Freie Autorin) <chloe@tel.net>
   beglaubiigt durch 1B087D04 um 2000-06-07
Wirklich ein Unterschrift-Widerrufszertifikat erzeugen? (j/N) j

Sie benötigen ein Mantra, um den geheimen Schlüssel zu entsperren.
Benutzer: ``Chloe (Journalistin) <chloe@cyb.org>''
1024-Bit DSA Schlüssel, ID 1B087D04, erzeugt 2000-06-07


pub  1024D/1B087D04  created: 2000-06-07 expires: never      trust: -/u
sub  2048g/6A3E902A  created: 2000-06-07 expires: never
sub  1792G/7D5D4DAE  created: 2000-06-07 expires: 2002-06-07
rev! subkey has been revoked: 2000-06-07
sub   960D/6E82436B  created: 2000-06-07 expires: 2002-06-07
```

```
(1)  Chloe (Journalistin) <chloe@cyb.org>
(2)  Chloe (Freie Autorin) <chloe@tel.net>
```

Eine widerrufene Benutzer-ID wird durch die Widerrufs-Signatur auf der Benutzer-ID angezeigt, wenn die Unterschriften auf den Benutzer-IDs des Schlüssels aufgelistet werden.

```
Befehl check

uid  Chloe (Journalistin) <chloe@cyb.org>
sig!       1B087D04 2000-06-07   [Eigenbeglaubigung]
uid  Chloe (Freie Autorin) <chloe@tel.net>
rev!       1B087D04 2000-06-07   [Widerruf]
sig!       1B087D04 2000-06-07   [Eigenbeglaubigung]
```

Ein Widerruf sowohl der Subkeys als auch der Eigenbeglaubigung auf Benutzer-IDs fügt dem Schlüssel eine Widerrufs-Signatur hinzu. Da also nur etwas hinzugefügt und nichts gelöscht wird, ist ein Widerruf für andere stets sichtbar, wenn Ihr aktueller öffentlicher Schlüssel weitergegeben und mit anderen älteren Kopien davon zusammengeführt wird. Der Widerruf garantiert deshalb, dass jeder die aktuelle Kopie Ihres öffentlichen Schlüssels haben kann.

1.4 Aktualisieren des Verfallsdatums

Das Verfallsdatum eines Schlüssels kann mit dem Befehl `expire` im Schlüsseleditor aktualisiert werden. Wenn kein Schlüssel ausgewählt ist, wird das Verfallsdatum des primären Schlüssels aktualisiert, ansonsten das des jeweils ausgewählten Subkeys.

Das Verfallsdatum eines Schlüssels ist mit der Eigenbeglaubigung des Schlüssels verbunden. Es wird dadurch aktualisiert, dass man die alte Eigenbeglaubigung löscht und eine neue hinzufügt. Da die Korrespondenzpartner die alte Eigenbeglaubigung noch nicht gelöscht haben, werden sie eine zusätzliche Eigenbeglaubigung auf dem Schlüssel sehen, wenn sie ihre Kopie Ihres Schlüssels aktualisieren. Die jüngste Eigenbeglaubigung hat jedoch jeweils Vorrang, und so werden alle Korrespondenzpartner unzweideutig die Verfallsdaten Ihrer Schlüssel kennen.

2 Authentisieren anderer Schlüssel

Wie in Kapitel intro bereits bereits ausführlich besprochen, wird der öffentliche Schlüssel eines Korrespondenzpartners dadurch authentisiert, dass Sie persönlich den Fingerabdruck seines Schlüssels prüfen und dann seinen öffentlichen Schlüssel mit Ihrem geheimen Schlüssel unterschreiben. Durch das persönliche Prüfen des Fingerabdrucks können Sie sicher sein, dass der Schlüssel wirklich ihm gehört. Da Sie den Schlüssel unterschrieben haben, können Sie sicher sein, jede Verfälschung an ihm in der Zukunft zu entdecken. Leider ist dieses Verfahren umständlich, wenn Sie entweder eine große Zahl von Schlüsseln authentisieren müssen oder wenn Sie mit Leuten kommunizieren, welche Sie nicht persönlich kennen.

GnuPG geht dieses Problem mit einem Mechanismus an, der allgemein als **Web of Trust** bezeichnet wird. Im Web of Trust wird die Verantwortlichkeit für das Authentisieren öffentlicher Schlüssel an Personen übertragen, denen Sie zutrauen, bei der Authentisierung von Schlüsseln die nötige Sorgfalt walten zu lassen. Nehmen Sie zum Beispiel folgendes an:

* Alice hat Blakes Schlüssel unterschrieben und
* Blake hat Chloes Schlüssel und Dharmas Schlüssel unterschrieben.

Wenn Alice Blake hinsichtlich der ordnungsgemäßen Authentisierung von Schlüsseln vertraut, dann kann sie davon ausgehen, dass Chloes und Dharmas Schlüssel gültig sind, ohne dass sie diese persönlich prüfen muß. Sie benutzt einfach ihre authentisierte Kopie von Blakes öffentlichem Schlüssel, um zu prüfen, dass Blakes Unterschriften auf den öffentlichen Schlüsseln von Chloe und Dharma echt sind. Im allgemeinen wird, wenn Alice bei allen Partnern völlig darauf vertraut, dass diese die von ihnen unterschriebenen Schlüssel richtig authentisieren, auch jeder mit einem gültigen Schlüssel unterschriebene Schlüssel als gültig betrachtet. Der Ausgangspunkt ist Alices Schlüssel, dessen Gültigkeit vorausgesetzt wird.

2.1 Vertrauen in den Eigentümer eines Schlüssels

Vertrauen ist in der Praxis natürlich immer subjektiv. So ist beispielsweise Blakes Schlüssel für Alice gültig, da sie ihn selbst unterschrieben hat, aber vielleicht traut sie Blake kein richtiges Authentisieren der von ihm unterschriebenen Schlüssel zu. In diesem Fall könnte sie die Gültigkeit von Chloes und Dharmas Schlüssel bezweifeln, da sich diese nur auf Blakes Unterschrift stützt. Das Web of Trust trägt diesem Umstand Rechnung, indem es jedem öffentlichen Schlüssel in Ihrem Schlüsselbund eine Angabe darüber zuordnet, inwieweit Sie dem Eigentümer des Schlüssels dahingehend vertrauen, dass er Schlüssel erst nach gründlicher Prüfung authentisiert. Es gibt vier Vertrauensstufen:

Unbekannt

Es ist nichts über die Fähigkeit des Eigentümers bekannt, Schlüssel vor dem Signieren zu authentisieren. Alle Schlüssel in Ihrem öffentlichen Schlüsselbund, die Ihnen nicht gehören, fallen zunächst unter diese Vertrauensstufe.

Kein Vertrauen

Der Eigentümer ist dafür bekannt, andere Schlüssel nicht korrekt zu unterschreiben.

Teilweises Vertrauen

Der Eigentümer versteht die Implikationen des Unterschreibens von Schlüsseln und authentisiert Schlüssel richtig, bevor er sie unterschreibt.

Volles Vertrauen

> Der Eigentümer hat ein ausgezeichnetes Verständnis hinsichtlich des Unterschreibens von Schlüsseln, und seine Unterschrift auf einem Schlüssel wäre so gut wie Ihre eigene.

Das Vertrauensmaß eines Schlüssels ist etwas, das Sie alleine dem Schlüssel zuordnen, und es wird als private Information betrachtet. Es wird nicht mit dem Schlüssel verpackt, wenn dieser exportiert wird; es wird sogar getrennt von Ihren Schlüsselbunden in einer gesonderten Trustdatenbank (trustdb.gpg) gespeichert.

Der GnuPG-Schlüsseleditor kann benutzt werden, um das Maß Ihres Vertrauens in den Eigentümer eines Schlüssels anzugeben. Der Befehl lautet `trust` (Andererseits fragt GnuPG auch nach, wenn es die Information braucht und noch kein Vertrauensmaß angegeben wurde). In diesem Beispiel gibt Alice das Maß ihres Vertrauens zu Blake an und aktualisiert dann entsprechend die Trustdatenbank, um neu zu ermitteln, welche Schlüssel auf der Basis ihrer neuen Einstufung von Blake gültig sind.

```
user@linux ~/ # gpg --edit-key blake

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


Befehl> trust

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

Bitte entscheiden Sie, inwieweit Sie diesem User zutrauen,
den Schlüssel eines anderen Users korrekt zu prüfen (Vergleich mit
Lichtbildausweisen, Vergleich der Fingerabdrücke aus unterschiedlichen
Quellen ...)?

 1 = Weiß nicht so recht
 2 = Kein Vertrauen
 3 = Ich vertraue ihm normalerweise
 4 = Ich vertraue ihm vollständig
 s = Bitte weitere Informationen anzeigen
 m = Zurück zum Menü


Ihre Auswahl? 3


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


Befehl> quit
```

Das Vertrauen in den Schlüssel-Eigentümer und in die Gültigkeit des Schlüssels wird rechts neben dem Schlüssel angezeigt. An erster Stelle wird das Vertrauen in den Eigentümer angezeigt, dann das Vertrauen in die Gültigkeit des Schlüssels. Die vier Vertrauensstufen werden folgendermaßen abgekürzt:

* Unbekannt (`q`),
* kein Vertrauen (`n`),
* teilweises Vertrauen (`m`) und
* volles Vertrauen (`f`)

In diesem Fall ist Blakes Schlüssel voll gültig, da Alice ihn selbst unterschrieben hat. Anfangs fallen Blakes Schlüssel für sie unter die Vertrauensstufe **Unbekannt**, doch sie entscheidet sich dafür, ihn unter **Teilweises Vertrauen** einzustufen.

2.2 Authentisieren von Schlüsseln im Web of Trust

Das Web of Trust ist ein flexibleres und komfortableres Verfahren zur Authentisierung eines Schlüssels. Früher wurde ein Schlüssel nur dann als gültig betrachtet, wenn er von Ihnen persönlich unterzeichnet war. Nach diesem Verfahren wird jetzt auch ein Schlüssel **K** als gültig betrachtet, wenn er die folgenden zwei Bedingungen erfüllt:

1. Schlüssel **K** ist von genügend gültigen Schlüsseln unterschrieben, das heißt, dass er entweder
   * von **Ihnen persönlich** oder
   * von **einem Schlüssel vollen Vertrauens** oder
   * von **drei Schlüsseln teilweisen Vertrauens** unterschrieben wurde.
2. Der Pfad unterschriebener Schlüssel, der vom Schlüssel **K** zurück zu Ihrem eigenen Schlüssel führt, besteht aus **maximal fünf Schritten**.

Die Pfadlänge, die Anzahl der erforderlichen Schlüssel Ihres teilweisen Vertrauens und die erforderliche Anzahl der Schlüssel Ihres vollen Vertrauens können Ihrer jeweiligen Vertrauensstufe angepaßt werden. Die oben angegebenen Zahlen sind die von GnuPG benutzten Standardwerte.

`wot-examples` zeigt ein Web of Trust, das seinen Ausgangspunkt bei Alice hat. Das Diagramm zeigt anschaulich, wer wessen Schlüssel unterschrieben hat und welche Schlüssel Alice aufgrund ihres Vertrauens in die anderen Mitglieder des Web of Trust als gültig betrachtet. In diesem Beispiel wird angenommen, dass zwei Schlüssel teilweisen Vertrauens oder ein Schlüssel vollen Vertrauens benötigt werden, um einen anderen Schlüssel zu authentisieren. Die maximale Pfadlänge beträgt drei Schritte.

Übersicht, wer wessen Schlüssel unterschrieben hat

Vertrauen		**Gültigkeit**	
teilweise	völlig	teilweise	völlig
	Dharma		Blake, Chloe, Dharma, Francis
Blake, Dharma	Francis		Blake, Chloe, Dharma
Chloe, Dharma		Chloe, Francis	Blake, Dharma
	Blake, Chloe,Elena		Blake, Chloe, Elena, Francis

Beim Berechnen der gültigen Schlüssel in dem Beispiel gilt folgendes: Blakes und Dharmas Schlüssel werden immer als voll gültig betrachtet, da sie direkt von Alice unterschrieben worden sind. Die Gültigkeit der anderen Schlüssel hängt vom Vertrauen ab. Im ersten Fall genießt Dharma volles Vertrauen, woraufhin die Schlüssel von Chloe und Francis als gültig betrachtet werden. Im zweiten Beispiel genießen Blake und Dharma nur teilweises Vertrauen. Da nun zwei Schlüssel teilweisen Vertrauens nötig sind, um einen Schlüssel voll zu authentisieren, wird der Schlüssel von Chloe als voll gültig, der von Francis aber nur als teilweise gültig betrachtet. Falls Chloe

und Dharma nur teilweises Vertrauen genießen, wird Chloes Schlüssel nur teilweise gültig sein, wähend Dharmas Schlüssel voll gültig ist. Der Schlüssel von Francis jedoch wird ebenfalls nur als teilweise gültig betrachtet, da nur ein voll gültiger Schlüssel zur Authentisierung anderer Schlüssel benutzt werden kann, und Dharmas Schlüssel der einzige voll gültige Schlüssel ist, der zum Unterschreiben des Schlüssels von Francis benutzt worden ist. Wenn teilweises Vertrauen in Blakes Schlüssel hinzukommt, kann Chloes Schlüssel voll gültig werden und kann dann zur vollen Authentisierung des Schlüssels von Francis und zur teilweisen Authentisierung des Schlüssels von Elena benutzt werden. Wenn schließlich Blake, Chloe und Elena volles Vertrauen genießen, reicht dies noch nicht aus, um den Schlüssel von Geoff zu authentisieren, da die maximal zulässige Länge des Zertifizierungspfades aus drei Schritten bestehen soll, die Pfadlänge von Geoff zurück zu Alice jedoch vier Schritte beträgt.

Das Web of Trust ermöglicht es Ihnen, GnuPG genau Ihren Vorstellungen von Sicherheit anzupassen. Sie könnten beispielsweise auf mehreren kurzen Pfaden von Ihrem Schlüssel aus zu einem anderen Schlüssel **K** bestehen, um diesem zu vertrauen. Vielleicht entscheiden Sie sich aber auch für längere Pfade oder sogar nur einen Pfad von Ihrem Schlüssel zu dem anderen Schlüssel **K**. Wenn Sie mehrfache kurze Pfade voraussetzen, so ist das eine starke Garantie dafür, dass Schlüssel **K** demjenigen gehört, von dem Sie dies annehmen. Der Preis dafür ist natürlich, dass die Authentisierung von Schlüsseln schwieriger ist, da Sie persönlich mehr Schlüssel unterschreiben müssen, als wenn Sie weniger und dafür längere Pfade akzeptieren.

3 Weitergabe von Schlüsseln

Im Idealfall wird ein Schlüssel durch persönliche Übergabe an Ihre Korrespondenzpartner weitergegeben. In der Praxis werden jedoch Schlüssel oft per E-Mail oder irgendein anderes elektronisches Kommunikationsmittel weitergegeben. Die Weitergabe per E-Mail ist durchaus annehmbar, wenn Sie nur einige wenige Korrespondenzpartner haben. Wenn Sie viele Korrespondenzpartner haben, könnten Sie beispielsweise Ihre(n) öffentlichen Schlüssel auf Ihrer Homepage im Web publizieren. Das setzt jedoch voraus, daß Ihre Korrespondenzpartner auch wissen, wo sie Ihre(n) Schlüssel finden können.

Um dieses Problem zu lösen, gibt es Key-Server, die öffentliche Schlüssel sammeln und weitergeben. Ein bei dem Server eingegangener öffentlicher Schlüssel wird entweder der Datenbank des Servers hinzugefügt oder mit Ihrem eventuell schon vorhandenen Schlüssel zusammengeführt. Wenn eine Anfrage nach einem Schlüssel beim Server eingeht, durchsucht dieser seine Datenbank und sendet den angeforderten öffentlichen Schlüssel zurück, wenn er ihn gefunden hat.

Ein Schlüssel-Server ist auch sinnvoll, wenn viele Leute häufig die Schlüssel anderer Leute unterschreiben. Ohne einen Schlüssel-Server würde Blake, wenn er Alices Schlüssel unterschreibt, an Alice eine Kopie ihres von ihm unterschriebenen Schlüssels schicken, so dass Alice den so aktualisierten Schlüssel ihrem Schlüsselbund hinzufügen und ihn auch an alle ihre Korrespondenzpartner weitergeben könnte. Mit dieser Mühe genügen Alice und Blake weitgehend ihrer Verantwortung gegenüber der Allgemeinheit durch den Aufbau engmaschiger Vertrauensnetze und helfen so, die Sicherheit von GPG zu verbessern. Dies ist jedoch sehr lästig, wenn das Unterschreiben von Schlüsseln häufig vorkommt.

Durch die Benutzung eines Schlüssel-Servers wird das etwas leichter. Wenn nun Blake Alices Schlüssel unterschreibt, so schickt er den unterschriebenen Schlüssel an den Schlüssel-Server, welcher dann Blakes Unterschrift seiner Kopie von Alices Schlüssel hinzufügt. Personen, die daran interessiert sind, ihre Kopie von Alices Schlüssel zu aktualisieren, wenden sich dann selbständig an den Schlüssel-Server, um sich den aktualisierten Schlüssel zu holen. Alice braucht sich mit der Weitergabe überhaupt nicht zu befassen und kann Unterschriften auf ihrem Schlüssel wie jeder andere auch einfach durch Anfrage bei einem Schlüssel-Server holen.

Ein oder mehr Schlüssel können unter Verwendung der Kommandozeilen-Option `--send-keys` an den Key-Server geschickt werden. Die Option erwartet eine Schlüssel-ID oder Benutzer-ID als Argument und schickt die so spezifizierten Schlüssel an den Key-Server. Der Key-Server, an den die Schlüssel geschickt werden sollen, wird durch die Kommandozeilen-Option `--keyserver` spezifiziert. In ähnlicher Weise wird die Option `--recv-keys` benutzt, um Schlüssel von einem Key-Server zu holen, doch müssen Sie hier den Schlüssel mit einer Schlüssel-ID spezifizieren. Im folgenden Beispiel aktualisiert Alice ihren öffentlichen Schlüssel mit neuen Unterschriften vom Key-Server blackhole.pca.dfn.de und schickt dann ihre Kopie von Blakes öffentlichem Schlüssel ebenfalls dorthin, um alle neuen Unterschriften, die sie hinzugefügt hat, weiterzugeben.

```
user@linux ~/ # gpg --keyserver wwwkeys.de.pgp.net --recv-key FB5797A9

gpg: Schlüssels FB5797A9 von wwwkeys.de.pgp.net wird angefordert ...
gpg: Schlüssel FB5797A9: 1 neue Signatur
gpg: Anzahl insgesamt bearbeiteter Schlüssel: 1
gpg:         neue Signaturen: 1

user@linux ~/ # gpg --keyserver wwwkeys.de.pgp.net --send-key
blake@cyb.org

gpg: Senden an `wwwkeys.de.pgp.net' erfolgreich (status=200)
```

Weltweit gibt es eine Vielzahl bekannter Key-Server. Die größeren Key-Server synchronisieren sich wechselseitig. Am Besten benutzen Sie einen gut erreichbaren Key-Server im Internet und tauschen dann regelmäßig über diesen Schlüssel aus. Eine kleine Auswahl gängiger Key-Server finden Sie im Anhang app-netres des Buches.

4 Fußnoten

GnuPG überfrachtet das Wort **Vertrauen**, indem sowohl **Vertrauen in einen Eigentümer** als auch **Vertrauen in einen Schlüssel** gemeint sein kann. Dies kann Verwirrung stiften. Manchmal wird das Vertrauen in einen Eigentümer zur klareren Unterscheidung als **Ownertrust** bezeichnet. In diesem Handbuch ist jedoch der Begriff **Vertrauen** durchweg in der Bedeutung **Vertrauen in den Eigentümer eines Schlüssels** benutzt worden, und der Begriff **Gültigkeit** bezieht sich darauf, dass ein Schlüssel der mit der Schlüssel-ID verknüpften Person gehört.