

# E-Mail Verschlüsselung mit GnuPG und Mozilla Thunderbird

German Privacy Foundation e.V.

6. Juli 2011

## Zusammenfassung

Der Einsatz kryptographischer Methoden ist insbesondere für die Kommunikation via E-Mail sinnvoll. Das europäische Verbraucheramt in Kiel, das BSI sowie der Bundesdatenschutzbeauftragte empfehlen die breite Nutzung folgender Methoden:

- **Signieren** von E-Mails: Eine vom Absender erstellte Signatur ermöglicht es dem Empfänger, die Identität des Absenders zu prüfen und gewährleistet, dass die E-Mail nicht verändert wurde.
- **Verschlüsseln** von E-Mails: Es wird die Vertraulichkeit der Kommunikation gewährleistet. Eine Nachricht kann nur vom Empfänger geöffnet und gelesen werden.

Mit OpenPGP und S/MIME haben sich zwei Standards für diese Aufgaben etabliert. Diese Anleitung beschreibt die Nutzung von GnuPG mit Mozilla Thunderbird.

## Inhaltsverzeichnis

<b>1</b>	<b>Asymetrische Verschlüsselung</b>	<b>2</b>
<b>2</b>	<b>GnuPG und Thunderbird</b>	<b>2</b>
2.1	Installation von GnuPG	2
2.2	Installation der Enigmail-Erweiterung	3
2.3	Schlüsselverwaltung	5
2.4	Signieren und Verschlüsseln erstellter E-Mails	7
2.5	Verschlüsselung in Webformularen	8
2.6	GnuPG SmartCard nutzen	8
2.7	Web des Vertrauens	13
2.8	Schlüssel zurückrufen	15
<b>3</b>	<b>Eine Bemerkung zum Abschluß</b>	<b>15</b>

# 1 Asymetrische Verschlüsselung

OpenPGP nutzt eine asymetrische Verschlüsselung. Das Verfahren ermöglicht es, eine öffentliche Komponente des Schlüssels, den sogenannten Public Key, auf einfache Art möglichst weit zu verbreiten und allen Partnern zur Verfügung zu stellen. Der geheime Teil des Schlüssels ist sicher zu verwahren.

Damit ist man nicht auf einen vertraulichen Kanal zum Schlüsseltausch angewiesen, da der öffentliche Schlüssel ohne seinen geheimen Gegenpart wertlos ist. Wie die beiden Keys genutzt werden, sollen folgende Beispiele beschreiben:

- Jeder Anwender generiert ein Schlüsselpaar bestehend aus einem geheimen und einem öffentlichen Schlüssel. Während der geheime Schlüssel sorgfältig geschützt nur dem Anwender selbst zur Verfügung stehen sollte, ist der öffentliche Schlüssel an alle Kommunikationspartner zu verteilen.
- Wenn der Anwender Anton eine signierte E-Mail an die Anwenderin Beatrice senden will, erstellt er eine Signatur mit *seinem geheimen Schlüssel*. Die Anwenderin Beatrice kann mit dem *öffentlichen Schlüssel von Anton* die Nachricht verifizieren.
- Wenn Beatrice eine verschlüsselte Nachricht an Anton senden will, nutzt sie den *öffentlichen Schlüssel von Anton*, um die Nachricht zu chiffrieren. Nur Anton kann diese E-Mail mit seinem geheimen Schlüssel dechiffrieren und lesen.

## 2 GnuPG und Thunderbird

Die folgende Anleitung erläutert den Einsatz von **GnuPG** in Kombination mit **Thunderbird**, dem E-Mail Client der Mozilla Foundation. Alle Komponenten stehen für Linux, Mac OS und WINDOWS kostenfrei zur Verfügung:

### 2.1 Installation von GnuPG

WINDOWS-User finden Binärpakete mit grafischer Installationsroutine auf der Website <http://gnupg.org/download/index.en.html> weiter unten im Abschnitt *Binaries*. Das Setup-Archiv ist nach dem Download mit den Rechten des Administrators zu starten.

Wir empfehlen das Paket **GnuPG-Pack**, welches einige zusätzliche Tools enthält wie WinPT Tray, GPGrelay, Enigmail und GPGSX. Mit GPGol soll auch Outlook die Verschlüsselung nutzen können. Das Paket steht unter <http://home.arcor.de/roseindorf/> zum Download bereit.

Nach dem Download ist das ZIP-Archiv zu entpacken. Das Setup-Programm (EXE-Datei) ist zu starten und den Anweisungen zu folgen. Nach Bestätigung der Lizenzbedingungen usw. kann man die zu installierenden Komponenten

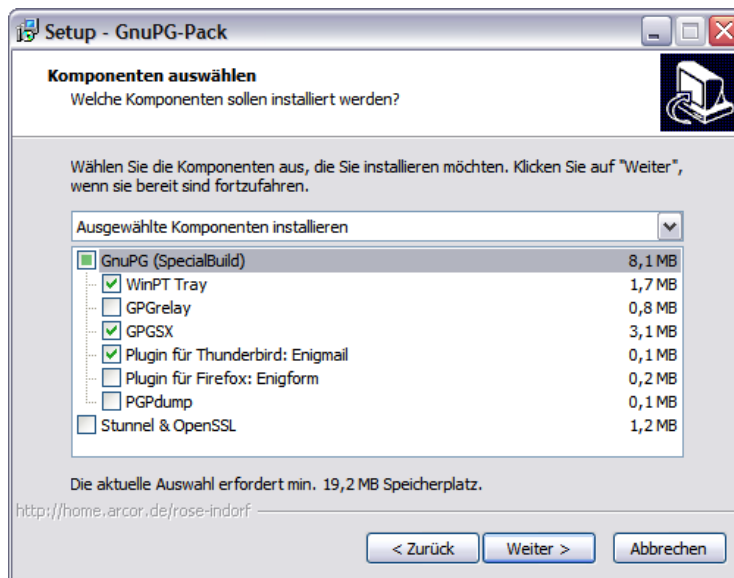


Abbildung 1: Komponenten des GnuPG-Pack zur Installation auswählen

auswählen (Bild 1).

Ob man WinPT für die Verschlüsselung via Zwischenablage benötigt, kann man selbst entscheiden. Für Thunderbird benötigt man das Enigmail Add-on. Außerdem sollte man die Erweiterung für den Explorer GPGSX mit installieren. Das vereinfacht die Ver- und Entschlüsselung von Dateien mit einem Klick im Explorer.

In einem weiteren Schritt werden verschiedene Zusatzfeatures angeboten. Statt der vorgeschlagenen Verschlüsselung der GPG-Schlüsselringe mit dem Windows Encrypt Filesystem empfehlen wir die komplette Verschlüsselung der Festplatte. Damit kann diese Option deaktiviert werden.

## 2.2 Installation der Enigmail-Erweiterung

Enigmail ist eine Erweiterung für Thunderbird, welche den Einsatz von GnuPG im täglichen E-Mail-Chaos vereinfacht. Die aktuelle deutsche Version steht unter <https://addons.mozilla.org/de/thunderbird/addon/71> zum Download zur Verfügung:

Nutzer der Mozilla-Browser sollten nicht mit der linken Maustaste auf den Download-Link klicken. Statt dessen ist mit der rechten Maustaste auf den Downloadlink zu klicken und im Kontextmenü der Punkt *Ziel speichern unter* zu wählen.

Nach dem Download ist Thunderbird zu starten und der Dialog zur Verwaltung der Erweiterungen über den Menüpunkt *Extras / Erweiterungen* zu öff-



Abbildung 2: Weitere Features vom GnuPG-Pack

nen. Hier wählt man den Button *Installieren* und in dem sich öffnenden Dateidialog das gespeicherte Plug-In (.xpi).

Nach Installation von Enigmail muss Thunderbird neu gestartet werden. Es wird der **Assistent** zur Einrichtung von Enigmail ausgeführt, der folgende Schritte durchläuft:

1. Abfrage, für welche Konten die Funktionen zur Verschlüsselung aktiviert werden sollen (in der Regel alle).
2. Abfrage, ob gesendete E-Mails standardmäßig signiert und verschlüsselt werden sollen. Um unbedarfte Anwender nicht zu verwirren, kann man das Signieren deaktivieren.
3. Optimierung der Einstellungen für GnuPG. Die Vorgaben sind sinnvoll und sollten übernommen werden.
4. Generieren der Schlüsselpaare für alle im Schritt 1 ausgewählten Konten. Die Passphrase für den Zugriff auf den privaten Key sollte man sich **vorher gut überlegen** und merken! Es heißt *Passphrase* und nicht *Passwort*. Die Passphrase darf ruhig etwas länger sein und auch Leer- bzw. Sonderzeichen enthalten.

Kryptografischen Funktionen können nicht unbegrenzt den Fortschritten der Kryptoanalyse widerstehen. Es ist sinnvoll, die Nutzungszeit des Schlüssels mit einem Haltbarkeitsdatum zu versehen. Eine Nutzung länger als **5 Jahre** sollte man nur in begründeten Ausnahmen in Erwägung ziehen. Bei der Schlüsselerstellung sollte ein Verfallsdatum angegeben

werden.

Mit jedem Schlüsselpaar kann auch ein Zertifikat für den Rückruf erstellt und sicher gespeichert werden. Mit diesem Zertifikat kann man einen Schlüssel für ungültig erklären, wenn der private Key kompromittiert wurde oder die Passphrase in Vergessenheit gerät.

Dieser 4. Schritt kann übersprungen werden, wenn man bereits gültige OpenPGP Schlüssel hat.

## 5. FERTIG

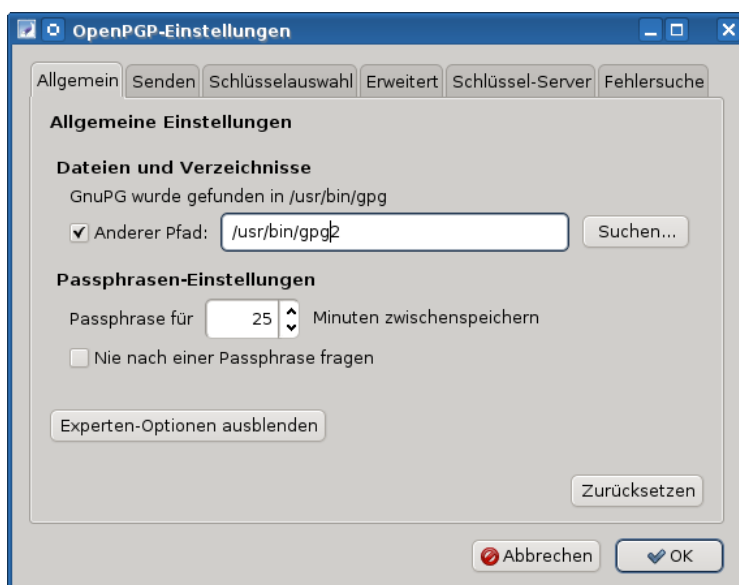


Abbildung 3: Einstellungen von EnigMail

Sollte Enigmail das Programm *gpg* nicht finden, weil man lieber die Version 2 *gpg2* von GnuPG nutzen möchte oder weil man es unter WINDOWS in einem selten verwendeten Verzeichnis liegt, wählt man den Menüpunkt *OpenPGP / Einstellungen* und gibt in der Dialogbox den Pfad zum GPG-Programm ein (Bild 3).

## 2.3 Schlüsselverwaltung

Die Schlüsselverwaltung findet man in Thunderbird unter dem Menüpunkt *OpenPGP / Schlüssel verwalten*. Ist die Liste noch leer, wählt man zuerst den Menüpunkt *Erzeugen / Neues Schlüsselpaar*. Diesen Schritt übernimmt jedoch der Assistent zur Einrichtung von EnigMail.

### Exportieren des eigenen öffentlichen Schlüssels

Um verschlüsselt zu kommunizieren, muss den Kommunikationspartnern der eigene öffentliche Schlüssel zur Verfügung gestellt werden. Der einfachste Weg

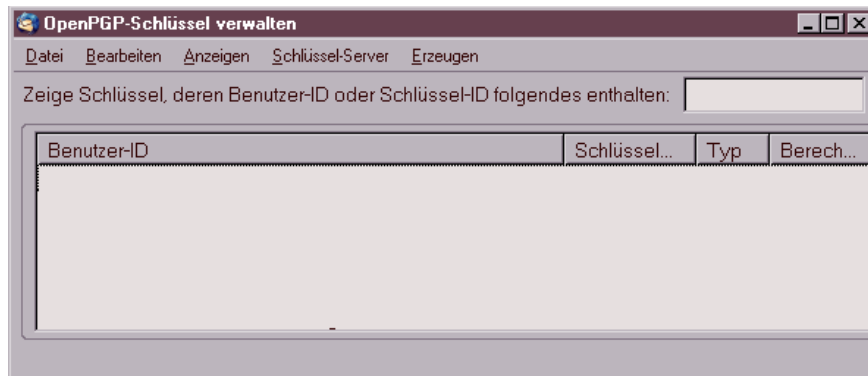


Abbildung 4: Schlüsselverwaltung von EnigMail

nutzt die Schlüsselserver im Internet. In der Schlüsselverwaltung findet man den Menüpunkt *Schlüssel-Server / Schlüssel hochladen*. Der öffentliche Schlüssel wird auf den Schlüsselserver exportiert und steht dort allen Partnern zur Verfügung. Die verschiedenen Server synchronisieren ihren Datenbestand.

Alternativ könnte man den öffentlichen Schlüssel in eine Datei exportieren und diese Datei anschließend als E-Mail-Attachment versenden oder auf einem Webserver ablegen. Den Menüpunkt für den Export in eine Datei findet man unter *Datei / Schlüssel exportieren* in der Schlüsselverwaltung.

### Import der Schlüssel der Partner

Um an einen Kommunikationspartner verschlüsselte E-Mails zu senden oder die Signatur erhaltener Nachrichten zu prüfen, benötigt man den öffentlichen Schlüssel des Partners.

- Am einfachsten lässt sich dieser importieren, wenn man eine signierte E-Mail erhalten hat. Ein Klick auf den blauen Stift rechts oben im Header der E-Mail reicht aus, um den öffentlichen Schlüssel von einem Schlüsselserver zu importieren.
- Zum Importieren des Schlüssel eines Partners aus einer Datei, die man als Attachment oder per Download erhalten hat, wählt man den Menüpunkt *Datei / Importieren*
- Auch ohne eine signierte E-Mail erhalten zu haben, kann man die Schlüsselserver nach dem zu einer E-Mail Adresse gehörenden Schlüssel durchsuchen. Die Funktion findet man unter dem Menüpunkt *Schlüssel-Server / Schlüssel suchen*. Man gibt in der sich öffnenden Dialogbox die E-Mail-Adresse des Empfängers ein und bestätigt die Suchanfrage mit OK.

Wurden zur Suchanfrage passende Schlüssel gefunden, werden diese in einer Liste angezeigt. Wählen Sie aus dieser Liste den zu importierenden Schlüssel und bestätigen Sie die Auswahl mit OK.

## 2.4 Signieren und Verschlüsseln erstellter E-Mails

Wurde in den Kontoeinstellungen in der Sektion *OpenPGP* die Option *Nachrichten standardmäßig verschlüsseln* aktiviert, sind beim Schreiben einer E-Mail keine weiteren Hinweise zu beachten. Anderenfalls ist für jede E-Mail explizit festzulegen, dass sie verschlüsselt werden soll.

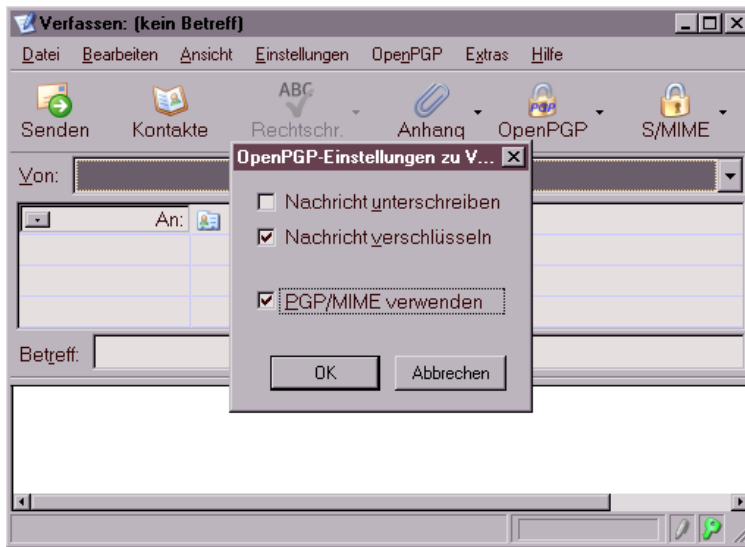


Abbildung 5: Signieren und Verschlüsseln einer E-Mail

Das Fenster für das Erstellen einer neuen E-Mail (Bild 5) zeigt nach der Installation des Enigmail-PlugIns einen neuen Button *OpenPGP*. Klickt man auf diesen Button, öffnet sich der im Bild 5 gezeigte Dialog, der es ermöglicht, die Krypto-Eigenschaften für diese E-Mail festzulegen.

Sollte die E-Mail Anhänge enthalten, ist die Option *PGP / MIME* zu aktivieren, um die Attachements standardkonform zu verschlüsseln.

**Achtung:** Die Betreffzeile wird nicht (!) mit verschlüsselt. Sicher wird man die Kontonummer nicht in der Betreffzeile schreiben, aber auch ein ausführlicher Betreff ermöglicht zusammen mit der/den Adressen der Empfänger einige Aussagen über die Kommunikation.

Wenn man als Betreff beispielsweise schreibt:

*Treffen der Aktivisten-Gruppe ... am 13.01.09*

und diese Mail per CC an alle Mitglieder der Gruppe versendet, sind 90% der relevanten Informationen bekannt und man kann sich die Verschlüsselung der Mail sparen.

Alternativ ist es auch möglich, lediglich für bestimmte Empfänger festzulegen, dass alle E-Mails signiert oder verschlüsselt werden sollen. Für die Festlegung dieser Regeln ist der entsprechende Dialog über *OpenPGP / Empfängerregeln* in Thunderbird zu öffnen.

## 2.5 Verschlüsselung in Webformularen

Auch bei der Nutzung eines Webmail Accounts oder Webforms für die Versendung anonymer E-Mails muss man auf Verschlüsselung nicht verzichten.

Einige GUIs für GnuPG (z.B. KGPG) enthalten einen Editor. Man kann den Text in diesem Editor schreiben, mit einem Klick auf den entsprechenden Button signieren oder verschlüsseln und das Ergebnis über die Zwischenablage in die Textbox der Website einfügen. Entschlüsseln funktioniert in umgekehrter Reihenfolge.

Enthält das bevorzugte Tool für die Schlüsselverwaltung keinen Texteditor, kann man folgende Alternativen nutzen, die auch für unterwegs (auf dem USB-Stick) geeignet sind:

1. Das kleine Tool **gpg4usb** (<http://gpg4usb.cpunk.de>) bietet einen Editor mit den Buttons für das Ver- und Entschlüsseln des Textes, Dateiverschlüsselung sowie eine kleine Schlüsselverwaltung (Signieren und Prüfen der Signatur steht noch auf der ToDo Liste). Das ZIP-Archiv enthält Versionen für Windows und Linux. Es kann einfach auf dem USB-Stick genutzt werden.
2. Die Applikation **Portable PGP** <http://ppgp.sourceforge.net> ist eine Java-Anwendung (plattformunabhängig), die ebenfalls Texte und Dateien ver- und entschlüsseln kann. Eine einfache Schlüsselverwaltung ist ebenfalls enthalten. Zusätzlich zu Portable PGP benötigt man eine Java Laufzeitumgebung. Eine portable Version der Sun-JRE gibt es bei [portableapps.com](http://portableapps.com).

## 2.6 GnuPG SmartCard nutzen

Die Sicherheit asymmetrischer Verschlüsselung hängt in hohem Maße von der sicheren Aufbewahrung des privaten Keys ab. Nutzt man GnuPG auf mehreren Rechnern, insbesondere wenn andere Nutzer Administrator- bzw. Root-Privilegien auf diesen Rechnern haben, könnte der private Key in falsche Hände gelangen.

Böswillige Buben könnten mit einem Trojaner versuchen, den privaten Key zu kopieren und das Passwort mit Tools wie *Elcomsoft Distributed Password Recovery* ermitteln. Die unbedachte Entsorgung einer Festplatte oder eines Computers ist ein weiteres Risiko, wenn der private Key nicht zuverlässig gelöscht wurde.

**SmartCards:** ermöglichen eine sichere Nutzung von GnuPG unter diesen Bedingungen. Der private Key ist ausschließlich auf der SmartCard gespeichert, er verläßt diese sichere Umgebung nicht. Sämtliche kryptografischen

Operationen werden auf der Card ausgeführt. CardReader (USB) und GnuPG-SmartCards gibt es bei [kernelconcepts.de](http://kernelconcepts.de).

**CryptoStick:** Da das Handling mit CardReader und SmartCard unter Umständen etwas umständlich sein kann, wird in der GPF ein USB-Stick entwickelt, der CardReader plus eine SmartCard in einem kleinen Gehäuse enthält und voll kompatibel mit der Version 2.0 der OpenPGP SmartCard ist. Projektseite: <http://wiki.privacyfoundation.de/GPFCryptoStick>



Abbildung 6: CryptoStick der GPF

### Hardware-Treiber installieren

Vor der Nutzung der SmartCard ist der Hardware-Treiber für den CardReader zu installieren.

- **WINDOWS:** Die Lieferung des CardReaders von [kernelconcepts.de](http://kernelconcepts.de) enthält eine CD mit den nötigen Treiber für WINDOWS. Das zum Gerät passende ZIP-Archiv ist zu unpacken und *setup.exe* als Administrator zu starten.

Für den CryptoStick der GPF gibt es den PC Twin USB PC/SC Treiber. Download Links: [https://www.awxcnx.de/handbuch\\_32r.htm](https://www.awxcnx.de/handbuch_32r.htm)

- **Linux:** Da Linux out-of-the-box viel mehr Hardware unterstützt als Windows, sind die nötigen Treiber in den Repositories enthalten. Unter Debian/Ubuntu installiert man alles Nötige für die Nutzung der SmartCard mit folgendem Kommando:

```
# aptitude install pcsd libpcsclite1 libccid
```

Für den CryptoStick v1.2 benötigt man keine Treiber, sondern nur eine UDEV-Regel. Für Debian/Ubuntu steht das Paket *cryptostick-1.0\_all.deb* in unserem Repository bereit. Für alle anderen Distributionen ist die Datei *40-cryptostick.rules* nach dem Download in */etc/udev/rules.d* zu speichern. Download Links: [https://www.awxcnx.de/handbuch\\_32r.htm](https://www.awxcnx.de/handbuch_32r.htm)

Die Pakete *openct* und *opensc* sollten entfernt werden, da diese zu Beeinträchtigungen führen können.

Außerdem benötigen die aktuelle OpenPGP-SmartCard und der CryptoStick GnuPG mindestens in der Version 1.4.9+ oder die 2.0.12+. Unter WINDOWS funktioniert erst die Version 1.4.10. Aktualisieren sie ihre GnuPG Version, wenn nötig.

Wer "gpg2" nutzen möchte, sollte beachten, dass der "gpg-agent" unbedingt nötig ist. In der Datei *\$HOME/.gnupg/gpg.conf* ist am Ende einfach ein *use-agent* einzufügen. Dann meldet man sich vom Desktop ab und wieder an.

Nachdem die Software installiert wurde, sollte man prüfen, ob alles funktioniert. SmartCard anschließen und auf der Konsole bzw. DOS-Box eingeben:

```
> gpg --card-status
Application ID ...: D27600xxxxxxxxxxxxxxxx
Version .....: 2.0
Manufacturer .....: unknown
....
```

### SmartCards und CryptoStick mit Enigmail nutzen

Enigmail ist ab der Version 1.0.1 voll kompatibel mit der SmartCard und dem CryptoStick. Aktuelle Linux-Versionen enthalten in der Regel eine ältere Version in den Repositories. Wer diese Pakete installiert hat, sollte sie wieder entfernen und eine aktuelle Enigmail Version von der Mozilla Addon Website installieren. Für Thunderbird 2.0.x kann auch Enigmail 0.96 genutzt werden. Evtl. muss man die SmartCard oder CryptoStick auf der Kommandozeile administrieren (siehe unten).

Das Plug-In bietet eine grafische Oberfläche, um die SmartCard zu verwalten. Diese Funktionen öffnet man über den Menüpunkt *OpenPGP - Smartcard verwalten*.

1. Als Erstes kann man die Card personalisieren und den Namen usw. editieren, eine URL für den Public Key angeben... (*Edit Card Data*).
2. Im zweiten Schritt sollte der PIN und der Admin-PIN geändert werden. Der PIN ist eine 6-stellige Zahlenkombination (Default: 123456), welche den User-Zugriff auf die Card sichert. Der Admin-PIN ist eine 8-stellige Zahlenkombination (Default: 12345678) für die Verwaltungsoperationen.

Wurde der PIN 3x falsch eingegeben, wird die Card gesperrt und kann mit dem Admin-PIN wieder entsperrt werden (*Unblock PIN*). Wird der Admin-PIN 3x falsch eingegeben, ist die SmartCard zerstört!

Die Festlegung auf 6- bzw. 8-stellige Zahlenkombinationen legt es nahe, ein Datum aus dem persönlichen Leben als PINs zu nutzen. Das reduziert die Vergesslichkeit. Es sollte jedoch kein einfach zu erratenes Datum wie der Geburtstag des Töchterchens sein.

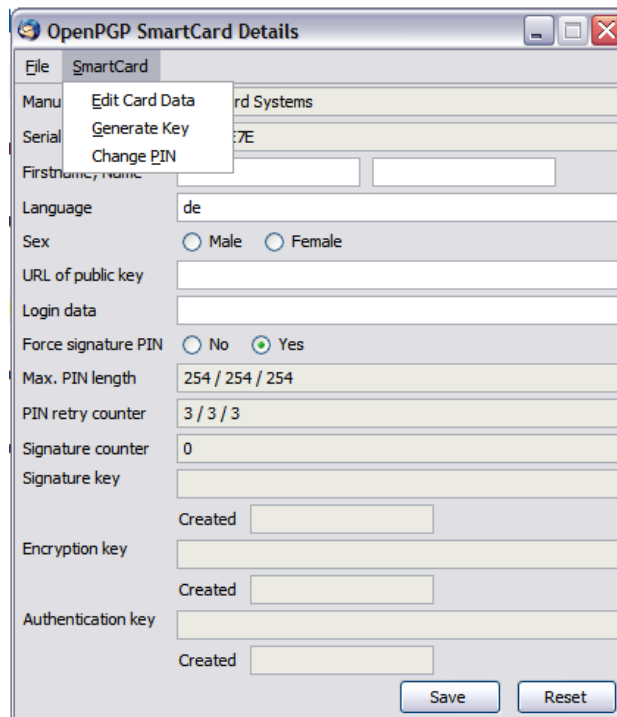


Abbildung 7: SmartCard verwalten

3. Als letzten Schritt vor der Nutzung der SmartCard im täglichen Krypto-Chaos sind die Keys auf der SmartCard zu generieren. Der entsprechende Dialog bietet die Auswahl eines Mail-Account an, für den die SmartCard genutzt werden soll. Für diesen Account darf kein(!) OpenPGP-Key vorhanden sein. Andernfalls bricht der Vorgang mit einer wenig verständlichen Fehlermeldung ab.

Es sollte unbedingt bei der Erzeugung des Schlüssels ein Backup der Card-Keys angelegt und mit einem Passwort gesichert werden. Später ist kein Zugriff auf diese Schlüssel mehr möglich. Bei Beschädigung der SmartCard kann der gesicherte Card-Key in eine neue SmartCard importiert werden. Das Backup wird im GnuPG-Verzeichnis abgelegt und ist auf einem sicheren Datenträger zu speichern!

Wurden die Schlüssel erfolgreich generiert, findet man in der *Schlüsselverwaltung* ein neues Paar. Der Public Key dieses Schlüsselpaars kann wie üblich exportiert und den Partnern zur Verfügung gestellt werden. Der Private Key dieses Paares definiert lediglich, dass die kryptografischen Operationen auf einer SmartCard auszuführen sind. Er ist ohne die passende Card unbrauchbar.



Abbildung 8: SmartCard-PINs ändern

### Funktionen für Genießer

Die Nutzung von gpg auf der Kommandozeile bietet etwas mehr Möglichkeiten, als bisher im Enigmail-GUI implementiert sind. Natürlich stehen auch die mit dem GUI durchführbaren Funktionen auf der Kommandozeile zur Verfügung.

Einen Überblick über alle SmartCard-Funktionen gibt die Hilfe. Als erstes muss man den Admin Mode aktivieren, dann hat man vollen Zugriff auf alle Funktionen:

```
> gpg --card-edit
Befehl> admin
Befehl> help
```

Neue Schlüssel generiert man auf der SmartCard mit:

```
> gpg --card-edit
Befehl> admin
Befehl> generate
```

Hat man mehrmals den PIN falsch eingegeben kann man ein neuen (alten) PIN (rück-)setzen, wenn man den Admin-PIN kennt:

```
> gpg --card-edit
Befehl> admin
Befehl> passwd
```

Möglicherweise hat man bereits eine OpenPGP Schlüssel mit vielen Signaturen. Den möchte man nicht wegwerfen und im Web of Trust noch einmal von vorn beginnen. Als Ausweg bietet es sich an, einen vorhandenen, starken Schlüssel mit der SmartCard zusätzlich zu schützen. Der Zugriff auf den geheimen Schlüssel ist dann nur mit der SmartCard möglich. Es ist dem vorhandenen Schlüssel mit der ID key-id ein Subkey der SmartCard hinzuzufügen. Das geht nur auf der Kommandozeile:

```
> gpg --edit-key key-id
command> addcardkey
```

Dabei wird ein evtl. auf der SmartCard vorhandener Key zertört!



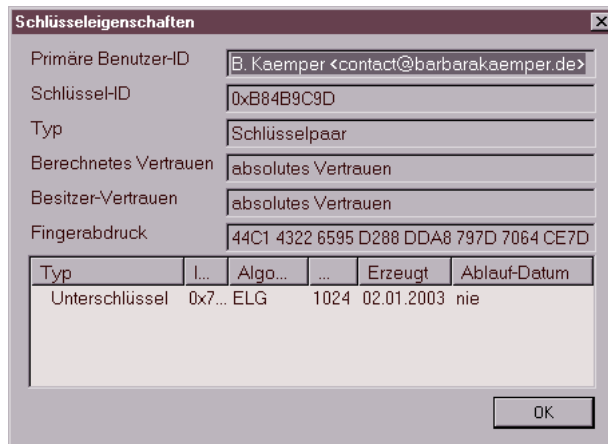


Abbildung 10: Schlüssel-Eigenschaften

ihrer Überprüfung online verfügbar machen, entsteht das Web-of-Trust und es wird schwer, gefälschte Schlüssel in Umlauf zu bringen.

### Certification Authorities

Diese Infrastruktur kann auch von vertrauenswürdigen Institutionen (Certification Authorities, CAs) genutzt werden. Die Nutzer wenden sich an die CA und lassen gegen Vorlage von Ausweisdokumenten den eigenen OpenPGP-Key signieren. Alle Partner benötigen lediglich den öffentlichen Schlüssel der CA, um die Echtheit der Schlüssel zu überprüfen.

Beispiele für Certification Authorities sind:

- Krypto-Kampagne der Zeitschrift Ct
- OpenPGP-CA der German Privacy Foundation e.V.
- PCA des Deutschen Forschungsnetzes (DFN-PCA)

### Keysigning-Party

Wenn sich mehrere OpenPGP-Nutzer treffen um sich gegenseitig die Echtheit ihrer Schlüssel zu bestätigen, nennt man es eine *Keysigning-Party*. Dabei kommt es nicht darauf an, dass die Beteiligten sich persönlich kennen. Die Echtheit des Schlüssels können auch Unbekannte gegen Vorlage von Ausweisdokumenten und Fingerprint des Key bestätigen.

Eine Keysigning-Party läuft üblicherweise folgendermaßen ab:

1. Der Organisator lädt zu einer Party ein und bittet um Anmeldungen.
2. Wer an der Party teilnehmen möchte, sendet seinen public OpenPGP-Key zusammen mit Namen und dem Fingerprint an den Organisator.

3. In Vorbereitung der Party erstellt der Organisator einen Keyring für alle Beteiligte und eine Liste mit Namen, Key-IDs und Fingerprints von allen Teilnehmern.
4. Der Keyring und die Liste werden an alle Teilnehmer verteilt. Die Teilnehmer können auf der Party die Identität gegenseitig durch Vorlage von Ausweisdokumenten prüfen.
5. Wieder zuhause können die Schlüssel im Party-Keyring signiert und an die Inhaber per E-Mail versendet werden. In der Regel erfolgt dieser Schritt nicht beim Treffen.

Wer häufiger an Keysigning-Partys teilnimmt, kann unter Linux das Tool *caff* für den letzten Schritt nutzen. Das Tool ist im Paket *signing-party* für nahezu alle Linux-Distributionen verfügbar und kann mit dem Paket-Manager der Wahl installiert werden.

Nach der Installation ist die Datei `$HOME/.caffrc` als Textdatei anzulegen und die Werte für den eigenen Namen, E-Mail Adresse, OpenPGP-ID sowie die Parameter zur Versendung von E-Mails sind zu konfigurieren:

```
$CONFIG{'owner'} = 'Michi Müller';
$CONFIG{'email'} = 'm@m.de';
$CONFIG{'keyid'} = [ qw{01234567890ABCDE} ];

$CONFIG{'mailer-send'} = [ 'smtp', Server => 'mail.server', Auth => ['user','pass'] ];
```

Ein kleines Kommando im Terminal signiert alle Schlüssel des Party-Keyring, verpackt sie in E-Mails, die mit dem Key der Empfänger verschlüsselt werden, und sendet die E-Mails an die Inhaber der OpenPGP-Keys:

```
> caff --key-file party-keyring.asc
```

## 2.8 Schlüssel zurückrufen

Soll ein Schlüsselpaar nicht mehr verwendet werden (beispielsweise weil der geheime Schlüssel kompromittiert wurde oder die Passphrase in Vergessenheit gefallen ist), kann der öffentliche Schlüssel für ungültig erklärt werden.

Öffnen Sie die Schlüsselverwaltung, wählen Sie den Schlüssel, der für ungültig erklärt werden soll. Rufen Sie den Menüpunkt *Bearbeiten / zurückrufen* auf. Nach einer Sicherheitsfrage und Eingabe der Passphrase wird der Schlüssel auf den Schlüsselservers im Internet für ungültig erklärt. Auch wenn der geheime Schlüssel nicht mehr vorliegt oder die Passphrase in Vergessenheit geraten ist, kann der öffentliche Schlüssel für ungültig erklärt werden, indem das unter Punkt 4 erstellte Rückrufzertifikat importiert wird.

## 3 Eine Bemerkung zum Abschluß

*“Mache ich mich verdächtig, wenn ich meine E-Mails verschlüsselt?”*

Eine Frage, die häufig gestellt wird, wenn es um verschlüsselte E-Mails geht. Bisher gab es darauf folgende Antwort:

*“Man sieht es einer E-Mail nicht an, ob sie verschlüsselt ist oder nicht. Wer befürchtet, dass jemand die Mail beschnüffelt und feststellen könnte, dass sie verschlüsselt ist, hat einen Grund mehr, kryptografische Verfahren zu nutzen!”*

Aktuelle Ereignisse zeigen, dass diese Frage nicht mehr so einfach beantwortet werden kann. Dem promovierten Soziologen Andrej H. wurde vorgeworfen, Mitglied einer terroristischen Vereinigung nach §129a StGB zu sein. Der Haftbefehl gegen ihn wurde unter anderem mit **konspirativem Verhalten** begründet, da er seine E-Mails verschlüsselte.

Am 21. Mai 2008 wurden in Österreich die Wohnungen von Aktivisten der Tierrechtsszene durchsucht und 10 Personen festgenommen. Der Haftbefehl wurde mit Verdunklungsgefahr begründet, da die Betroffenen z.B. über verschlüsselte E-Mails kommunizierten.

Am 18.10.07 hat der Bundesgerichtshof (BGH) in seinem Urteil [Az.: StB 34/07](#) den Haftbefehl gegen Andrej H. aufgehoben und eindeutig festgestellt, dass die Verschlüsselung von E-Mails als Tatverdacht NICHT ausreichend ist, entscheidend sei der Inhalt:

*“Ohne eine Entschlüsselung der in den Nachrichten verwendeten Tarnbegriffe und ohne Kenntnis dessen, was bei den - teilweise observierten und auch abgehörten - Treffen zwischen dem Beschuldigten und L. besprochen wurde, wird hierdurch eine mitgliedschaftliche Einbindung des Beschuldigten in die ‘militante gruppe’ jedoch nicht hinreichend belegt.”*

Außerdem geben die Richter des 3. Strafsenat des BGH zu bedenken, dass Andrej H. *“ersichtlich um seine Überwachung durch die Ermittlungsbehörden wusste”*. Schon allein deshalb konnte er *“ganz allgemein Anlass sehen”*, seine Aktivitäten zu verheimlichen. Woher Andrej H. von der Überwachung wusste, steht bei <http://annalist.noblogs.org>.

Trotz dieses Urteils des BGH bleibt für uns ein bitterer Nachgeschmack über die Arbeit unser Ermittler und einiger Richter. Zumindest die Ermittlungsrichter sind der Argumentation der Staatsanwaltschaft gefolgt und haben dem Haftbefehl erst einmal zugestimmt.