E-Mail-Signatur und -Verschlüsselung mit PGP (GnuPG)

René Knipschild
Custom Software Development
www.rkcsd.com
Postfach 1468
D-34484 Korbach

15. September 2014

Inhaltsverzeichnis

1 Einleitung

E-Mails sind so offen lesbar wie Postkarten. Klassiche Briefe sind grundsätzlich erstmal sicherer. Es ist jedoch möglich, mithilfe von Computertechnik sicher zu kommunizieren. Die klassische E-Mail kann u.a. mithilfe eines Verfahrens, was sich PGP („Pretty Good Privacy“ = „wirklich geschützte Privatsphäre“) schimpft, sicher verschlüsselt werden. Das Verfahren basiert auf mathematischen Problemen, von denen man nach aktuellem Stand der Technik ausgeht, dass sie schwierig lösbar sind.[1] Konkret wird das Verschlüsseln mittels eines Schlüssels möglich, dessen Ergebnis nur mit einem zugehörigen anderen Schlüssel wieder entschlüsselt werden kann. Online-Version dieses Artikels in der rkWiki: http://rkcsd.eu/pgpsetup

Niemand sollte dazu beitragen, dass Geheimdienste, Regierungen oder Privatfirmen Zugriff auf die persönliche oder geschäftliche Kommunikation haben. Betriebsspionage vermeiden und persönliche Freiheit wahren[2] – im Informationszeitalter ist Verschlüsselung für jeden ein Muss. Sich mit diesem Thema nicht beschäftigen zu wollen ist vergleichbar mit dem Unwillen, sich mit Politik und Wahlen zu beschäftigen – der Anfang vom Ende einer freien Gesellschaft also.

2 Wie funktioniert’s?

Es ist wichtig, zumindest im Groben im Hinterkopf zu haben, wie PGP funktioniert. **In der pratischen Benutzung wird alles hier gezeigte jedoch vollautomatisch vom PC erledigt, also keine Sorge.** Mit dem ***öffentlichen*** *Schlüssel* kann so verschlüsselt werden, dass nurnoch mit dem ***privaten*** *Schlüssel* entschlüsselt werden kann.[3] Das bedeutet, dass jeder meinen öffentlichen Schlüssel haben darf, um mir verschlüsselte E-Mails zu senden, und ich allein besitze die Datei, mit der die E-Mail entschlüsselt werden kann.

Das PGP-E-Mail-System basiert also darauf, dass ein Schlüsselaustausch vor der Kommunikation stattfindet.[4] Ich muss meinen Kommunikationspart-

[1] Bewiesen ist dies aber nicht. Man kann jedoch mit gutem Gewissen die Behauptung aufstellen, dass diese Verschlüsselungstechnik aktuell nicht zu knacken ist.

[2] Grundvorrausetzung hierfür ist allerdings auch, offene Kommunikationsinfrastruktur wie E-Mails zu verwenden. Jeder kann einen eigenen Mailserver oder einen eigenen Chatserver betreiben. Wenn ein Server ausfällt, läuft die Kommunikation über andere Server weiter. Wer stattdessen über properitäre Protokolle (einzelne Plattformen) wie WhatsApp, Skype, etc. kommuniziert kann jederzeit vom Kommunizieren ausgeschlossen werden. Fallen diese Plattformen aus, ist Schicht im Schacht.

[3] Unter den Schlüsseln sind Dateien mit, riesengroßen (100 Stellen und mehr) Zufallszahlen darin zu verstehen – vereinfacht ausgedrückt.

[4] Das ist der Unterschied zu symmetrischen Verschlüsselungsverfahren, wo der Schlüssel geheim bleiben muss.

nern meinen öffentlichen Schlüssel geben (damit diese mir verschlüsselte Nachrichten schicken können) und sie müssen mir jeweils den ihrigen geben (damit ich ihnen verschlüsselte Nachrichten schicken kann). Sinn ist, den öffentlichen Schlüssel zu veröffentlichen, damit mir jeder verschlüsselte Mails schicken kann. Es ist daher nicht schlimm, wenn beispielsweise die öffentlichen Schlüssel via einer unverschlüsselten E-Mail ausgetauscht werden.

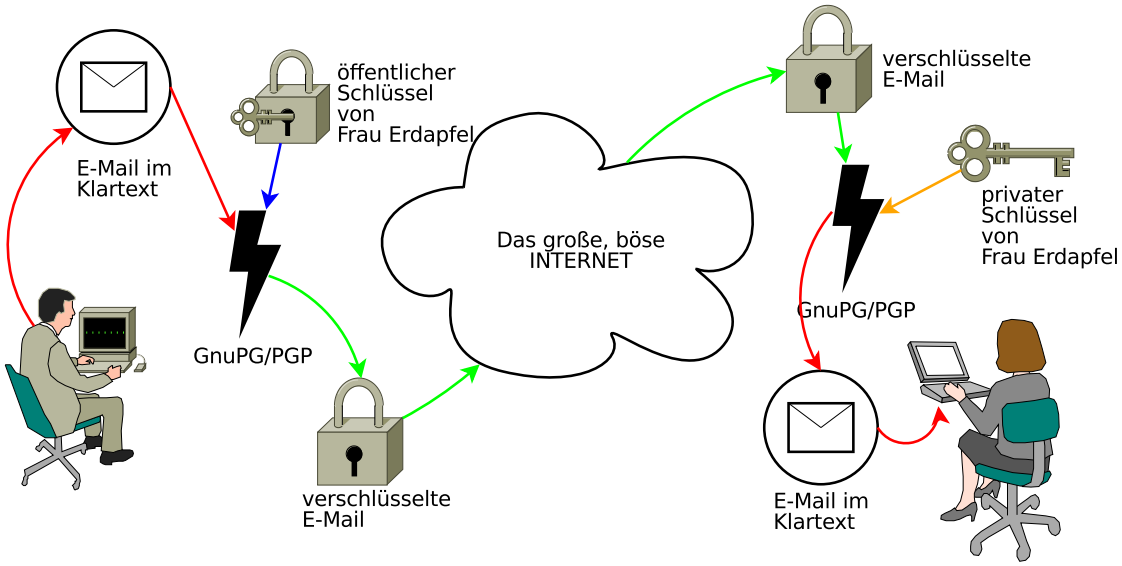

Abbildung 1: Das PGP-System

Eine sehr gelungene Illustration finden Sie auch als Video unter http://www.mailbox.org/stiftfilm-wie-funktioniert-e-mail-verschluesselung-mit-pgp/.

3 Einrichtung von PGP mittels GnuPG

3.1 Generieren der Schlüssel

Um die Nutzung von PGP zu beginnen, muss man sich ersteinmal ein eigenes Schlüsselpaar erzeugen, also einen privaten und einen öffentlichen Schlüssel. Wir arbeiten mit dem Programm GnuPG, welches unter Linux-Systemen normalerweise standardmäßig installiert ist (bzw. sich unter z.B. Debian-Systemen mit dem Kommando apt-get install gnupg nachinstallieren ließe). Unter Windows muss für die folgenden Kommandos gpg.exe im Programmverzeichnis statt gpg aufgerufen werden.[5]

[5] GnuPG-Downloads erhalten Sie unter http://www.gnupg.org/download/index.html, Thunderbird und Enigmail erhalten Sie über http://getthunderbird.com/ bzw. http://addons.mozilla.org/. Bei Nichtverfügbarkeit können wir Ihnen auf Anfrage Downloads über unsere eigenen Archiv-Server ermöglichen.

Konsole
```
$ gpg --gen-key
```

Mit diesem Befehl wird in unserem Home-Verzeichnis ein Ordner .gnupg mit der Schlüsseldatenbank erstellt. Darin werden zunächst unsere eigne-ne Schlüssel gespeichert, später kommen noch die öffentlichen Schlüssel unserer Kommunikationspartner hinzu. Nachdem wir den Befehl abgesetzt haben, werden wir nach Schlüsseltyp und -länge gefragt. Hier genügt es üb-licherweise, die Standardeinstellungen mit [ENTER] zu bestätigen. Das Ab-laufdatum des Schlüssels kann frei gewählt werden. Aus Sicherheitsgründen kann ein Ablaufen des Schlüssels in Erwägung gezogen werden. Bei Name und E-Mailadresse sollten natürlich die richtigen Daten eingegeben werden. Es sollte die E-Mailadresse eingestellt werden, die zur PGP-Kommunikation verwendet werden soll. (Später können noch weitere hinzugefügt werden.)

Nachfolgend mal komplett abgebildet, wie das in der Shell aussieht. Wie man sieht, werden die Schlüssel anhand von Zufallszahlen erzeugt, die wie-derum nur dann ausreichend im System „vorrätig“ sind, wenn ein bisschen Last „anliegt“. Ich habe hier an einer Stelle eine MD5-Summe erzeugt, dann wurde das Erzeugen der Schlüssel erfolgreich abgeschlossen.

Konsole
```
$ gpg --gen-key
gpg (GnuPG) 1.4.12; Copyright (C) 2012 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: Verzeichnis ‘/home/rk/.gnupg’ erzeugt
gpg: Neue Konfigurationsdatei ‘/home/rk/.gnupg/gpg.conf’ erstellt
gpg: WARNUNG: Optionen in ‘/home/rk/.gnupg/gpg.conf’ sind waehrend
dieses Laufes noch nicht wirksam
gpg: Schluesselbund ‘/home/rk/.gnupg/secring.gpg’ erstellt
gpg: Schluesselbund ‘/home/rk/.gnupg/pubring.gpg’ erstellt
Bitte waehlen Sie, welche Art von Schluessel Sie moechten:
   (1) RSA und RSA (voreingestellt)
   (2) DSA und Elgamal
   (3) DSA (nur unterschreiben/beglaubigen)
   (4) RSA (nur signieren/beglaubigen)
Ihre Auswahl?
```

Hier kann man bei der Standart-Schlüssel-Art bleiben. Elgamal ist theoretisch zwar besser, aber RSA ist bislang nicht gebrochen.

Konsole
```
Ihre Auswahl? 1

RSA-Schluessel koennen zwischen 1024 und 4096 Bit lang sein.
Welche Schluessellaenge wuenschen Sie? (2048) 4096
Die verlangte Schluesellaenge betraegt 4096 Bit
Bitte waehlen Sie, wie lange der Schluessel gueltig bleiben soll.
         0 = Schluessel verfaellt nie
      <n>  = Schluessel verfaellt nach n Tagen
      <n>w = Schluessel verfaellt nach n Wochen
      <n>m = Schluessel verfaellt nach n Monaten
```

```
      <n>y = Schluessel verfaellt nach n Jahren
Wie lange bleibt der Schluessel gueltig? (0)
```

Aus Sicherheitsgründen sollte der Schlüssel nach einer Zeit ablaufen, praktischer ist natürlich ein unbegrenzt gültiger Schlüssel. Im Beispiel wählen wir zwei Jahre Gültigkeit. Je länger (in Bit) ein Schlüssel ist, desto schwieriger wird das zugrundeliegende mathematische Problem, also desto sicherer ist der Schlüssel.

Konsole

```
Wie lange bleibt der Schluessel gueltig? (0) 2y
Key verfaellt am Sa 17 Okt 2015 12:04:44 CEST
Ist dies richtig? (j/N) j

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

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

Hier werden nun einfach die persönlichen Daten eingegeben. Standardmäßig wird die Mailadresse zur Zuordnung des Schlüssels benutzt, daher muss es sich hierbei um die Mailadresse handeln, mit der hinterher auch verschlüsselt gemailt werden soll.

Konsole

```
Ihr Name ("Vorname Nachname"): Rene Knipschild
Email-Adresse: knipschild@rkcsd.com
Kommentar: Custom Software Development
Sie benutzen den Zeichensatz ‘utf-8’
Sie haben diese User-ID gewaehlt:
    "Rene Knipschild (Custom Software Development) <knipschild@rkcsd.com>"

Aendern: (N)ame, (K)ommentar, (E)-Mail oder (F)ertig/(B)eenden?
```

Alles richtig? Dann bestätigen. Jetzt kommt noch das wichtige Kennwort, was den PGP-Schlüssel schützt und später immer bei Entschlüsselung oder beim Signieren eigener Mails eingegeben werden muss.

Konsole

```
Aendern: (N)ame, (K)ommentar, (E)-Mail oder (F)ertig/(B)eenden? F
Sie benoetigen eine Passphrase, um den geheimen Schluessel zu schuetzen.
```

Jetzt wird der geheime und öffentliche Schlüssel anhand von Zufallszahlen sicher erzeugt.

Konsole

```
Wir muessen eine ganze Menge Zufallswerte erzeugen.  Sie koennen dies
unterstuetzen, indem Sie z.B. in einem anderen Fenster/Konsole irgendetwas
tippen, die Maus verwenden oder irgendwelche anderen Programme benutzen.
.....................+++++
.+++++
```

```
Wir muessen eine ganze Menge Zufallswerte erzeugen.  Sie koennen dies
unterstuetzen, indem Sie z.B. in einem anderen Fenster/Konsole irgendetwas
tippen, die Maus verwenden oder irgendwelche anderen Programme benutzen.

Es sind nicht genuegend Zufallswerte vorhanden.  Bitte fuehren Sie andere
Arbeiten durch, damit das Betriebssystem weitere Entropie sammeln kann!
(Es werden noch 169 Byte benoetigt.)
+++++

Es sind nicht genuegend Zufallswerte vorhanden.  Bitte fuehren Sie andere
Arbeiten durch, damit das Betriebssystem weitere Entropie sammeln kann!
(Es werden noch 217 Byte benoetigt.)
+++++
gpg: /home/rk/.gnupg/trustdb.gpg: trust-db erzeugt
gpg: Schluessel 560EACC8 ist als uneingeschraenkt vertrauenswuerdig gekennzeichnet
Oeffentlichen und geheimen Schluessel erzeugt und signiert.

gpg: "Trust-DB" wird ueberprueft
gpg: 3 marginal-needed, 1 complete-needed, PGP Vertrauensmodell
gpg: Tiefe: 0  gueltig:   1  unterschrieben:   0  Vertrauen: 0-, 0q, 0n, 0m, 0f, 1u
gpg: naechste "Trust-DB"-Pflichtueberpruefung am 2015-10-17
pub   4096R/560EACC8 2013-10-17 [verfaellt: 2015-10-17]
  Schl.-Fingerabdruck = 8D2A 050A 046E 8193 4E07  5052 8BF7 271C 560E ACC8
uid             Rene Knipschild (Custom Software Development) <knipschild@rkcsd.com>
sub   4096R/D00FF918 2013-10-17 [verfaellt: 2015-10-17]
```

Damit ist die Einrichtung abgeschlossen und PGP betriebsbereit. Es ist übrigens auch möglich, PGP mittels eines grafischen Aufsatzes ohne Kommandozeile einzurichten. Ich halte allerdings nichts von diesen unnötigen grafischen Oberflächen. Per Befehl kommt man schneller zum Ziel. ;-)

3.2 Hinzufügen zusätzlicher Identitäten

Wenn man mehrere E-Mailadressen mit dem Schlüssel nutzen will, kann man zusätzliche Identitäten zu dem Schlüssel hinzufügen. Dazu muss man einfach den Schlüssel mit folgendem Befehl bearbeiten:

Konsole
```
$ gpg --edit-key <Schluessel-ID>
```

Die Schlüssel-ID wurde beim Erzeugen angezeigt. Im Beispiel lautet sie 560EACC8. Man kann die Schlüssel-IDs nachträglich anzeigen, indem man den Befehl

Konsole
```
$ gpg --list-keys
```

eingibt. Dann erhält man auch die IDs der öffentlichen Schlüssel der Kommunikationspartner.

Im folgenden Prompt mithilfe des befehls adduid bzw. deluid Identitäten hinzufügen oder wieder entfernen.

3.3 Sichern und Austausch der Schlüssel

Alle verschlüsselten E-Mails können von GnuPG nur mithilfe des eigenen geheimen Schlüssels entschlüsselt werden. Geht dieser verloren, sind auch die E-Mails unlesbar und somit unwiederbringlich verloren. Es ist daher dringend nötig, den geheimen Schlüssel zu sichern, falls der eigene PC z.B. kaputt geht. Datensicherheit und Datenschutz schließen sich leider zum Teil aus: Trotz Backup darf der geheime Schlüssel niemals an Dritte gelangen. Das Backup sollte also auf einem verschlüsselten Datenträger gespeichert werden. Am besten sollte sogar die Festplatte des eigenen Rechners verschlüsselt werden, damit auch hier der Schlüssel nicht durch verlust oder Diebstahl an Dritte gelangen kann. Für nicht-PC-affine Menschen empfehle ich, den Schlüssel schlicht auszudrucken und gemeinsam mit den wichtigen Unterlagen abzuheften, ggf. in einem verschlossenen Aktenschrank.

Der geheime Schlüssel kann folgendermaßen exportiert werden:

Konsole
```
$ gpg -a --output gpg-secret-key.asc --export-secret-keys <Schluessel-ID>
```

Die Datei `gpg-secret-key.asc` an einem sicheren Ort aufbewahren[6]) oder mit einem Texteditor öffnen und ausdrucken. Diese Datei darf unter keinen Umständen in die Hände dritter gelangen! Wir exportieren im nächsten Schritt den öffentlichen Schlüssel, der zur Weitergabe bestimmt ist. Diese Dateien auf jeden Fall getrennt speichern, damit man nicht aus Versehen den geheimen Schlüssel veröffentlicht.

Um nun den zur Weitergabe bestimmten öffentlichen Schlüssel zu exportieren benutzen wir folgenden Befehl:

Konsole
```
$ gpg -a --output gpg-public-key.asc --export <Schl ssel-ID>
```

Damit der Schlüssel als vertrauenswürdig eingestuft wird, den Schlüssel wie oben beschrieben bearbeiten und mithilfe von sign signieren. Optional kann noch mithilfe von `trust` die Vertrauensstufe festgelegt werden.

4 Praktische Nutzung

Wenn die Einrichtung und der Import der öffentlichen Schlüssel der Kommunikationspartner erfolgreich war, benutzt sich GnuPG wie von selbst. Im Mailprogramm muss nurnoch mit „GnuPG verschlüsseln“ vor dem Schreiben der entsprechenden Mails ausgewählt werden (Bild: Thunderbird mit Enigmail unter Windows 8) und schon verschlüsselt das Mailprogramm die E-Mails. Es muss nur sichergestellt sein, dass das Mailprogramm (z.B. Evolution oder

[6] AUF KEINEN FALL AUF EINEN ONLINE-SERVER HOCHLADEN

Thunderbird mit Enigmail) das PGP-Programm findet. Notfalls den Pfad korrekt einstellen, aber normalerweise stimmt diese Einstellung standardmäßig bzw. PGP wird automatisch gefunden.

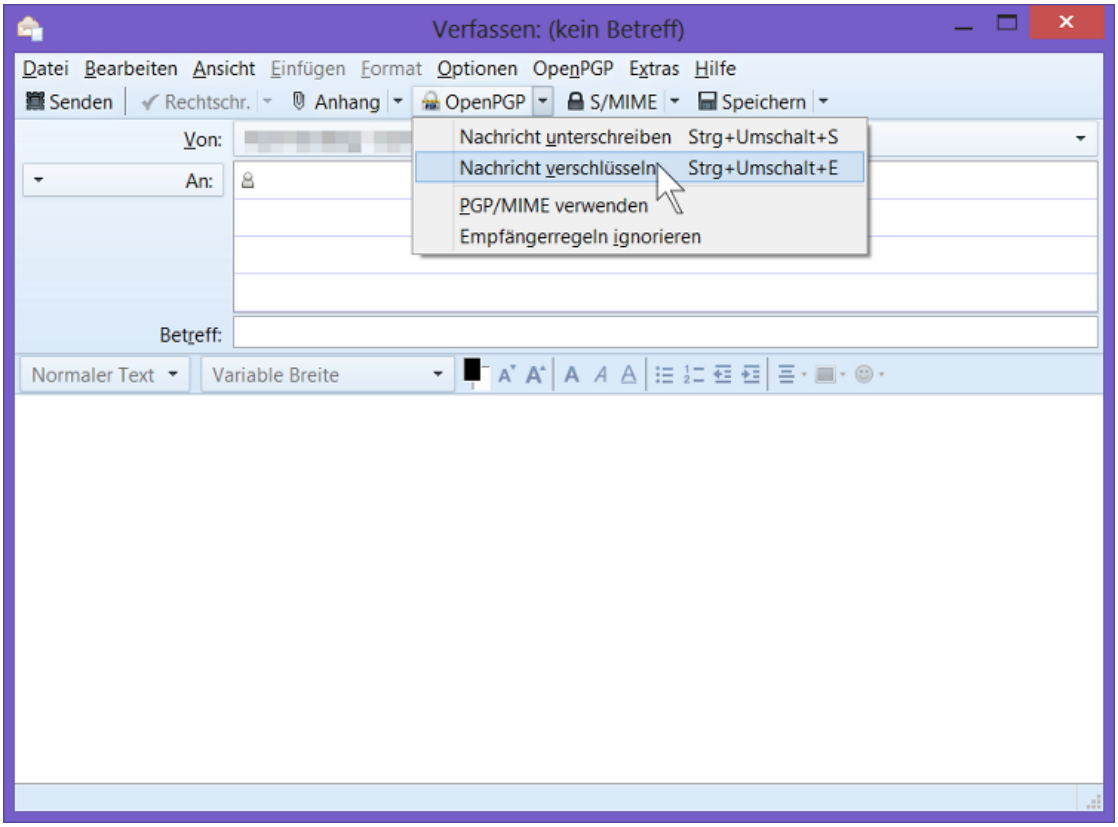

Abbildung 2: Verschlüsselte Mail unter Windows mit Thunderbird verfassen

Es muss nurnoch sichergestellt werden, dass beim Verschicken von E-Mails die Nachrichten im „Gesendet“-Verzeichnis für einen selbst verschlüsselt gespeichert werden. Hierzu entsprechenden Haken im Mailprogramm setzen. (Bild: Evolution unter Ubuntu Linux) Dies ist deshalb nötig, weil die ausgehende Nachricht ja nur für den Empfänger entschlüsselbar ist.

5 Weitere Informationen

Dieser Artikel soll kurz und knapp die Einrichtung von Mailverschlüsselung darlegen. Detailiertere Infos gibt es zum Beispiel in der Ubuntuusers Wiki.

Eine Alternative zu GnuPG stellt S/MIME dar, welches ganz ähnlich wie PGP funktioniert und ebenso Sicherheit bietet. Das Problem bei S/MIME ist jedoch, dass es ähnlich zu SSL nötig ist, von CAs ausgestellte Zertifikate zur Verschlüsselung zu verwenden. Die Validierung der Vertrautheit bei SSL-CAs ist

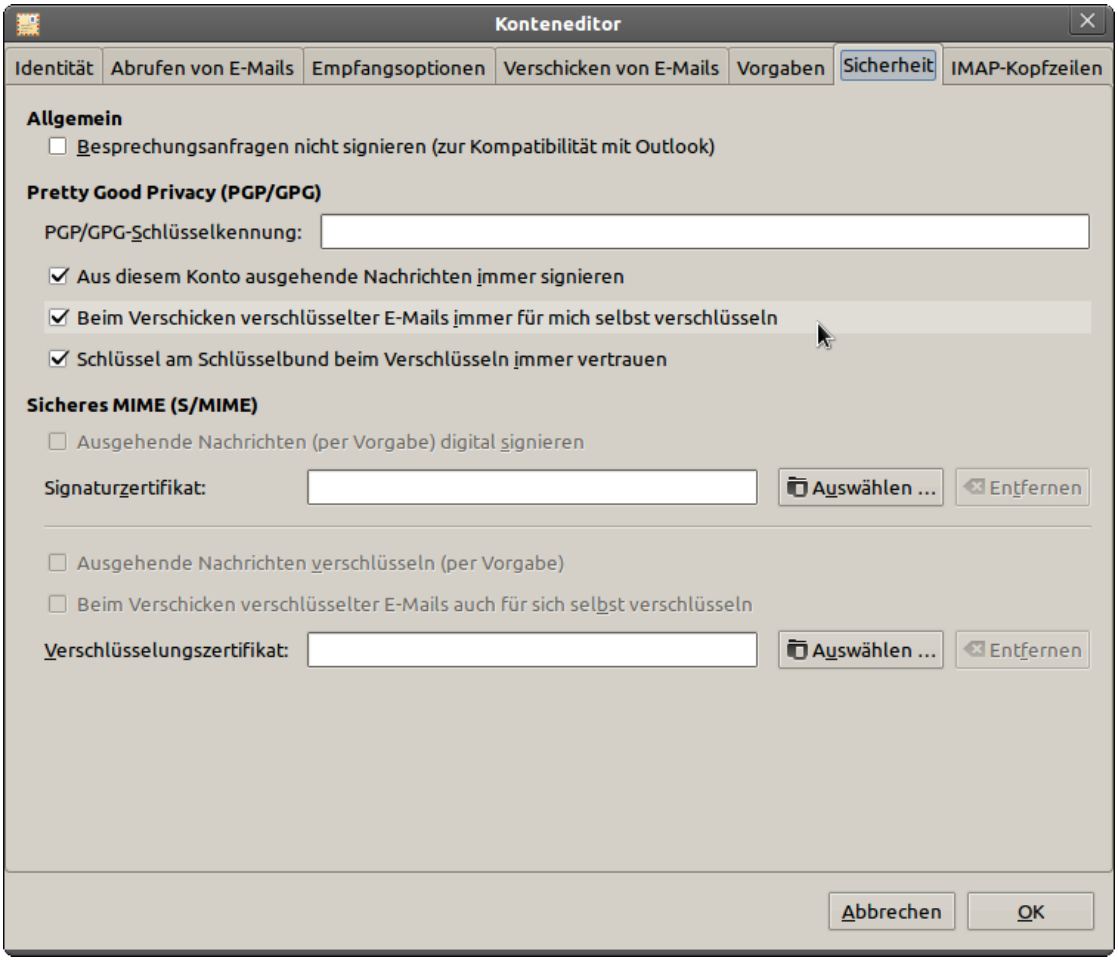

Abbildung 3: Verschickte Mails für mich selbst verschlüsseln mit Evolution unter Ubuntu Linux

jedoch unnötig kompliziert und bietet auch nicht unbedingt höhere Sicherheit, da man dazu ja den CAs blind vertrauen muss. Key-Austausche können bei beiden Verfahren manipuliert werden, vorinstallierte CAs bei S/MIME ebenso. Wir empfehlen PGP, da der CA-Rattenschwanz und somit ggf. hohe Kosten entfallen. Wer best mögliche Sicherheit will muss die Keys persönlich austauschen und nicht übers Internet schicken, da die Key-Austausche – wie gesagt – manipulierbar sind.

Was nicht sicher ist und bleibt sind die Mail-Header, die unverschlüsselt übertragen werden müssen. Wenn man die Adresse auf einen Brief nicht im Klartext schreibt, kommt der ja auch nicht an. Natürlich übermitteln Mailserver die E-Mails komplett verschlüsselt, aber beim Provider ist „wer mailt mit wem“ offen einsehbar.