



Das Gpg4win Kompendium

Eine Veröffentlichung des Gpg4win Projekts

Basierend auf einem Original von

Manfred J. Heinze, Karl Bihlmeier, Isabel Kramer
Dr. Francis Wray und Ute Bahn.

Überarbeitet von

Werner Koch, Emanuel Schütze und Jan-Oliver Wagner.

Version 3.0.0-beta1 vom 22. Oktober 2008

Impressum

Copyright © 2002 Bundesministerium für Wirtschaft und Technologie¹

Copyright © 2005 g10 Code GmbH

Copyright © 2008 Intevation GmbH

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled „GNU Free Documentation License“.

[Dieser Absatz ist eine unverbindliche Übersetzung des oben stehenden Hinweises.]

Es wird die Erlaubnis gegeben, dieses Dokument zu kopieren, zu verteilen und/oder zu verändern unter den Bedingungen der GNU Free Documentation License, Version 1.2 oder einer späteren, von der Free Software Foundation veröffentlichten Version. Es gibt keine unveränderlichen Abschnitte, keinen vorderen Umschlagtext und keinen hinteren Umschlagtext. Eine Kopie der „GNU Free Documentation License“ findet sich im Anhang mit dem gleichnamigen Titel. Inoffizielle Übersetzungen dieser Lizenz finden Sie unter <http://www.gnu.org/licenses/translations.html>.

Wie das Kryptographieprogramm Gpg4win selbst, wurde dieses Dokument nicht für Mathematiker, Geheimdienstler und Kryptographen geschrieben, sondern für jedermann.

¹Wenn dieses Dokument kopiert, verteilt und/oder verändert wird, soll in keiner Form der Eindruck eines Zusammenhangs mit dem Bundesministerium für Wirtschaft und Technologie erweckt werden.

Inhaltsverzeichnis

Über dieses Handbuch	7
I. Einsteiger	8
1. Was ist Gpg4win?	9
2. Warum überhaupt verschlüsseln?	11
3. Zwei Wege, ein Ziel: OpenPGP & S/MIME	14
4. Sie installieren Gpg4win	17
5. Sie erzeugen Ihr Schlüsselpaar	28
5.1. OpenPGP-Schlüsselpaar erstellen	31
5.2. X.509-Schlüsselpaar erstellen	36
5.3. Schlüsselpaar-Erstellung abgeschlossen	41
6. Sie veröffentlichen Ihr Zertifikat	42
6.1. Veröffentlichen per E-Mail	43
6.2. Veröffentlichen per Schlüsselserver	48
7. Sie entschlüsseln eine E-Mail	50
8. Sie importieren ein Zertifikat	54
9. Sie verschlüsseln eine E-Mail	57
10. Sie signieren eine E-Mail	63
10.1. Signieren mit GpgOL	64
10.2. Signatur mit GpgOL überprüfen	70
10.3. Gründe für eine gebrochene Signatur	72
10.4. Verschlüsseln und signieren	73
11. Wie Sie Ihre E-Mails verschlüsselt archivieren	74

II. Fortgeschrittene	77
12. Wie funktioniert Gpg4win?	78
13. Die Passphrase	89
14. Zertifikat im Detail	92
15. Die OpenPGP-Schlüsselserver	94
16. Die Zertifikatsprüfung	98
17. Dateianhänge verschlüsseln	105
18. Dateien signieren und verschlüsseln	106
18.1. Dateien signieren und überprüfen	107
18.2. Dateien verschlüsseln und entschlüsseln	114
19. Im- und Export eines geheimen Schlüssels	123
19.1. Export	124
19.2. Import	125
20. Systemweite Konfigurationen und Vorbelegungen für S/MIME	127
21. Bekannte Probleme und was man tun kann	128
21.1. GpgOL Menüs und Dialog nicht mehr in Outlook zu finden	128
21.2. GpgOL Icons zum Signieren/Verschlüsseln fehlen in der Symbolleiste	128
21.3. Keine S/MIME Operationen mehr möglich (Systemdienst „DirMngr“ läuft nicht) . .	128
22. Wo finde ich die Dateien und Einstellungen von Gpg4win?	129
22.1. Persönliche Einstellungen der Anwender	129
22.2. Zwischengespeicherte Sperrlisten	129
22.3. Vertrauenswürdige Wurzeln von DirMngr	129
22.4. Weitere Zertifikate von DirMngr	130
22.5. Konfiguration zur Verwendung externer LDAP Verzeichnisdienste	130
22.6. Systemweite vertrauenswürdige Wurzel-Zertifikate	131
23. Fehler in den Gpg4win-Programmen aufspüren	132
23.1. Logbuch von Kleopatra einschalten	133
23.2. Logbuch von GpgOL einschalten	134
23.3. Logbuch von DirMngr einschalten	134
23.4. Logbuch von GnuPG einschalten	135
24. Warum Gpg4win nicht zu knacken ist . . .	136

25. GnuPG und das Geheimnis der großen Zahlen	137
25.1. Das Rechnen mit Restklassen	139
25.2. RSA-Algorithmus und Rechnen mit Restklassen	142
25.3. RSA Verschlüsselung mit kleinen Zahlen	143
25.4. Die Darstellung mit verschiedenen Basiszahlen	148

III. Anhang	156
A. Hinweise zum Outlook Plugin <i>GpgOL</i>	157
B. GnuPG mit anderen E-Mail-Programme nutzen	160
C. Automatische Installation von Gpg4win	161
D. Umstieg von anderen GnuPG Programmen	163
E. Deinstallation von Gpg4win	164
F. History	165
G. GNU Free Documentation License	166

Über dieses Handbuch

Das Gpg4win-Anleitungs- und Übungsmaterial besteht aus drei Teilen:

- **Teil I für Einsteiger:** Der Schnelleinstieg in Gpg4win.
- **Teil II für Fortgeschrittene:** Das Hintergrundwissen für Gpg4win.
- **Der Übungsroboter Adele**, mit dem Sie die E-Mail-Ver- und Entschlüsselung (mit OpenPGP) so oft üben können, wie Sie wollen.

Teil I für „Einsteiger“ führt Sie kurz und knapp durch die Installation und die alltägliche Benutzung der Gpg4win-Software. Der Zeitbedarf für das Durcharbeiten des Schnelleinstiegs hängt unter anderem davon ab, wie gut Sie sich mit Ihrem PC und Windows auskennen. Sie sollten sich in etwa eine Stunde Zeit nehmen.

Teil II für „Fortgeschrittene“ liefert Hintergrundwissen, das Ihnen die grundlegenden Mechanismen von Gpg4win verdeutlicht und die etwas seltener benutzten Fähigkeiten erläutert.

Beide Handbuchteile können unabhängig voneinander benutzt werden. Zu Ihrem besseren Verständnis sollten Sie aber möglichst beide Teile in der angegebenen Reihenfolge lesen.

Der Übungsroboter Adele steht Ihnen im Internet zur Verfügung. Adele empfängt und sendet verschlüsselte OpenPGP E-Mails und entschlüsselt sie auch. Sie können also mit Adele einen kompletten Verschlüsselungsdialog so lange üben, bis Sie sich völlig mit dem Gebrauch der Software vertraut gemacht haben.

Adele ist im Rahmen des alten GnuPP Projektes entstanden und läuft dort noch immer. „Das Gpg4win Kompendium“ verwendet diesen zuverlässigen Übungsroboter und dankt den Inhabern von gnupp.de für den Betrieb von Adele.

Teil I.
Einsteiger

1. Was ist Gpg4win?

Das Projekt Gpg4win (GNU Privacy Guard for Windows) ist eine Verschlüsselungssoftware für E-Mails und Dateien. Gpg4win bezeichnet ein Gesamtpaket, welches in Version 2 die folgenden Programme umfasst:

GnuPG: GnuPG ist das Kernstück von Gpg4win: Die Verschlüsselungs-Software.

Kleopatra: Die zentrale Zertifikatsverwaltung von Gpg4win. Unterstützt OpenPGP und X.509 (S/MIME) und sorgt für eine einheitliche Benutzerführung für alle kryptographischen Operationen.

GpgOL: GnuPG für Outlook (GpgOL) ist eine Erweiterung für Microsoft Outlook 2003 und 2007, die verwendet wird um Nachrichten mit OpenPGP oder S/MIME zu verschlüsseln.

GpgEX: GPG Explorer eXtension (GpgEX) ist eine Erweiterung für den Windows Explorer, die es ermöglicht, Dateien über das Kontextmenü zu verschlüsseln.

GPA: Der GNU Privacy Assistent (GPA) ist neben Kleopatra ein alternatives Programm zum Verwalten von OpenPGP Schlüsseln.

Claws Mail: Claws Mail ist ein vollständiges E-Mail-Programm mit sehr guter Unterstützung für GnuPG.

Mit dem Verschlüsselungsprogramm GnuPG (GNU Privacy Guard) kann jedermann E-Mails sicher, einfach und kostenlos verschlüsseln. GnuPG kann ohne jede Restriktion privat oder kommerziell benutzt werden. Die von GnuPG eingesetzte Verschlüsselungstechnologie ist sehr sicher und kann nach dem heutigen Stand von Forschung und Technik nicht gebrochen werden.

GnuPG ist **Freie Software**¹. Das bedeutet, dass jedermann das Recht hat, sie nach Belieben kommerziell oder privat zu nutzen. Jedermann darf den Quellcode, also die eigentliche Programmierung des Programms, genau untersuchen und auch selbst Änderungen durchführen und diese weitergeben.²

Für eine Sicherheits-Software ist diese garantierte Transparenz des Quellcodes eine unverzichtbare Grundlage. Nur so läßt sich die Vertrauenswürdigkeit eines Programmes prüfen.

GnuPG basiert auf dem internationalen Standard **OpenPGP** (RFC 2440), ist vollständig kompatibel zu PGP und benutzt die gleiche Infrastruktur (Schlüsselserver etc.). Seit Version 2 von GnuPG wird auch der kryptographische Standard **S/MIME** (CMS/RFC 3852 bzw. X.509) unterstützt.

¹ oft ungenau auch als Open Source Software bezeichnet

² Obwohl dies ausdrücklich erlaubt ist, sollte man ohne ausreichendes Fachwissen nicht leichtfertig Änderungen durchführen, da hierdurch die Sicherheit der Software beeinträchtigt werden kann.

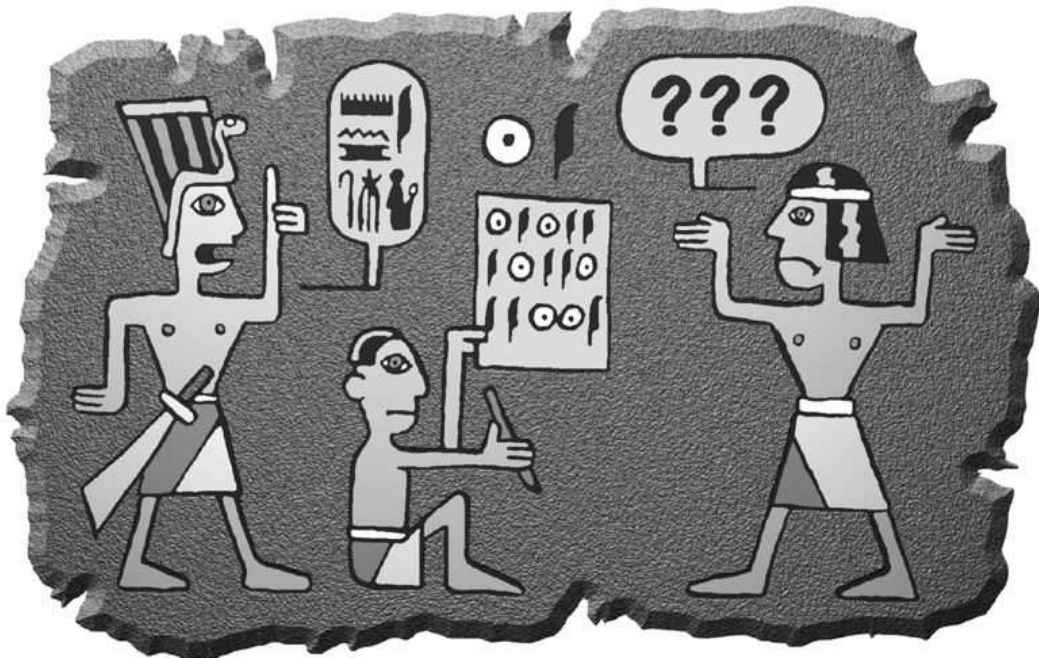
PGP („Pretty Good Privacy“) ist keine Freie Software, sie war lediglich vor vielen Jahren kurzzeitig zu ähnlichen Bedingungen wie GnuPG erhältlich. Diese Version entspricht aber schon lange nicht mehr dem Stand der Technik.

Die Vorläufer von Gpg4win wurden durch das Bundesministerium für Wirtschaft und Technologie, Gpg4win und Gpg4win-2 durch das Bundesamt für Sicherheit in der Informationstechnik unterstützt.

Weitere Informationen zu GnuPG und den Projekten der Bundesregierung zum Schutz des Internets finden Sie auf der Website www.bsi-fuer-buerger.de des Bundesamtes für Sicherheit in der Informationstechnik.

2. Warum überhaupt verschlüsseln?

Die Verschlüsselung von Nachrichten wird manchmal als das zweitälteste Gewerbe der Welt bezeichnet. Verschlüsselungstechniken benutzten schon der Pharao Khnumhotep II, Herodot und Cäsar. Dank Gpg4win ist Verschlüsselung nunmehr für jedermann frei und kostenlos zugänglich. . .



Die Computertechnik hat uns phantastische Mittel in die Hand gegeben, um rund um den Globus miteinander zu kommunizieren und uns zu informieren. Aber Rechte und Freiheiten, die in anderen Kommunikationsformen längst selbstverständlich sind, müssen wir uns in den neuen Technologien erst sichern. Das Internet ist so schnell und massiv über uns hereingebrochen, dass wir mit der Wahrung unserer Rechte noch nicht so recht nachgekommen sind.

Beim altmodischen Briefschreiben haben wir die Inhalte unserer Mitteilungen ganz selbstverständlich mit einem Briefumschlag geschützt. Der Umschlag schützt die Nachrichten vor fremden Blicken, eine Manipulation am Umschlag kann man leicht bemerken. Nur wenn etwas nicht ganz so wichtig ist, schreibt man es auf eine ungeschützte Postkarte, die auch der Briefträger oder andere lesen können.

Ob die Nachricht wichtig, vertraulich oder geheim ist, das bestimmt man selbst und niemand sonst.

Diese Entscheidungsfreiheit haben wir bei E-Mail nicht. Eine normale E-Mail ist immer offen wie eine Postkarte, und der elektronische „Briefträger“ – und andere – können sie immer lesen. Die Sache ist sogar noch schlimmer: die Computertechnik bietet nicht nur die Möglichkeiten, die vielen Millionen E-Mails täglich zu befördern und zu verteilen, sondern auch, sie zu kontrollieren.

Niemand hätte je ernsthaft daran gedacht, alle Briefe und Postkarten zu sammeln, ihren Inhalt auszuwerten oder Absender und Empfänger zu protokollieren. Das wäre einfach nicht machbar gewesen, oder es hätte zu lange gedauert. Mit der modernen Computertechnik ist das technisch möglich. Es gibt mehr als einen Hinweis darauf, dass dies genau heute schon im großen Stil mit Ihrer und meiner E-Mail geschieht¹.

Denn: Der Umschlag fehlt.



¹Hier sei nur an das Echelon System erinnert (siehe <http://www.heise.de/tp/r4/artikel/6/6928/1.html>).

Was wir Ihnen hier vorschlagen, ist ein Umschlag für Ihre elektronischen Briefe. Ob Sie ihn benutzen, wann, für wen und wie oft, ist ganz allein Ihre Sache. Software wie Gpg4win gibt Ihnen lediglich die Wahlfreiheit zurück. Die Wahl, ob Sie persönlich eine Nachricht für wichtig und schützenswert halten oder nicht.

Das ist der Kern des Rechts auf Brief-, Post- und Fernmeldegeheimnis im Grundgesetz, und dieses Recht können Sie mit Hilfe der Software Gpg4win wahrnehmen. Sie müssen sie nicht benutzen – Sie müssen ja auch keinen Briefumschlag benutzen. Aber es ist Ihr gutes Recht.

Um dieses Recht zu sichern, bietet Gpg4win Ihnen sogenannte „starke Verschlüsselungstechnik“. „Stark“ bedeutet hier: mit keinem gegenwärtigen Mittel zu knacken. In vielen Ländern waren starke Verschlüsselungsmethoden bis vor ein paar Jahren den Militärs und Regierungsbehörden vorbehalten. Das Recht, sie für jeden Bürger nutzbar zu machen, haben sich die Internetnutzer mühsam erobert; manchmal auch mit der Hilfe von klugen und weitsichtigen Menschen in Regierungsinstitutionen, wie im Falle der Portierung von GnuPG auf Windows. GnuPG wird von Sicherheitsexperten in aller Welt als eine praktikable und sichere Software angesehen.

Wie wertvoll diese Sicherheit für Sie ist, liegt ganz in Ihrer Hand.

Sie allein bestimmen das Verhältnis zwischen Bequemlichkeit bei der Verschlüsselung und größtmöglicher Sicherheit. Dazu gehören die wenigen, aber umso wichtigeren Vorkehrungen, die Sie treffen müssen, um Gpg4win richtig zu nutzen. In diesem Handbuch werden wir Ihnen dieses Vorgehen Schritt für Schritt erläutern...

3. Zwei Wege, ein Ziel: OpenPGP & S/MIME

Wie so oft gibt es für das gleiche Ziel verschiedene Wege, ähnlich ist es auch in der Verschlüsselung Ihrer E-Mails mit den Standards OpenPGP und S/MIME. Beide Standards und ihre Umsetzungen in Software ermöglichen die E-Mail-Verschlüsselung mit Freier Software, wie zum Beispiel Gpg4win.

Beim Verschlüsseln bzw. bei der Sicherheit der geheimen Datenübertragung sind zwei Perspektiven wichtig, einmal die Gewährleistung der **Geheimhaltung** und zum anderen die **Authentizität** des Absenders. Authentizität bedeutet hier, dass der Inhalt auch tatsächlich vom besagten Absender ist.

Die Gemeinsamkeit: Das Public-Key-Verfahren

Konzeptionell steckt hinter OpenPGP und S/MIME das gleiche System zur Geheimhaltung, und zwar das Public-Key-Verfahren. Was heisst das?

Nehmen wir an, die E-Mail oder die Datei sei in einer Truhe verschlossen. Im Gegensatz zu einem „normalen“ Schloss mit einem Schlüssel gibt es beim Public-Key-Verfahren zum Verschlüsseln/Entschlüsseln **ein Schlüsselpaar**. So gibt es einen beglaubigten Schlüssel zum Verschlüsseln (das „**Zertifikat**“) und einen Schlüssel zum Entschlüsseln (der „**Geheimer Schlüssel**“).

Klingt zwar komisch, wenn man an echte Schlösser denkt, aber bei Software löst diese Idee das Problem, dass ich meinen Schlüssel für jeden Empfänger aus der Hand geben müsste. Denn normalerweise muss ein Schlüssel zum Verschlüsseln/Abschließen auch zum Entschlüsseln bzw. Aufschließen benutzt werden. Also müsste ich Ihnen, wenn ich etwas für Sie in der Truhe verschließe, die Truhe **und** den Schlüssel geben. Wenn der Schlüssel bei der Übertragung abhanden kommt oder jemand davon eine Kopie erstellt, ist das ein großes Problem.

Beim Public-Key-Verfahren verschließe ich mit Ihrem „**Zertifikat**“ die Truhe und Sie schließen die Truhe mit Ihrem „**Geheimen Schlüssel**“ auf. Ich muss also nur die Truhe zu Ihnen transportieren lassen. Das ist auf jeden Fall sicherer als den geheimen Schlüssel mit zu transportieren, selbst wenn er einen anderen Weg als die Truhe zu Ihnen nehmen würde.

Trotz dieses gleichen Ansatzes zur Geheimhaltung unterscheiden sich OpenPGP und S/MIME aber zum Beispiel bei der Schlüsselerzeugung (Näheres erfahren wir später in Kapitel 5).

Falls Sie sich jetzt fragen, wie das Public-Key-Verfahren so funktionieren kann, lesen Sie einmal Kapitel 12. Wenn Sie sich dann noch fragen, warum Gpg4win so sicher ist, sind vermutlich die Kapitel 24 und 25 genau das Richtige für Sie! Mit ein wenig Interesse und Zeit kann man dort auch die kleinen mathematischen Geheimnisse verstehen. Viel Spaß beim Entdecken.

Der Unterschied: Die Authentifizierung

Der wesentlichste Unterschied zwischen OpenPGP und S/MIME liegt im Bereich der Authentifizierung.

Um die Authentizität des Absenders festzustellen, ist bei **S/MIME** ein Zertifikat notwendig, welches die Authentizität des Schlüssel-Besitzers unzweifelhaft beglaubigt. Das heisst, dass ich meinen öffentlichen Schlüssel von einer dazu berechtigten Organisation zertifizieren lassen muss, bevor er dadurch wirklich nutzbar wird. Diese Organisation wurde wiederum von einer höher stehenden Organisation zertifiziert usw. bis man zu einem Wurzel-Zertifikat kommt. Vertraut man nun diesem Wurzel-Zertifikat, so vertraut man automatisch allen darunter liegenden Zertifizierungen. Das nennt man **hierarchisches Vertrauenskonzept**. Zumeist ist die Kette nur 3 Elemente lang: Wurzel, Zertifizierungsstelle (auch CA für Certificate Authority genannt), Anwender. Wurzel zertifiziert CA, CA zertifiziert Anwender.

Im Gegensatz dazu erlaubt **OpenPGP** neben dieser baumartigen Zertifizierung zusätzlich auch eine direkte „peer-to-peer“ Zertifizierung (Anwender A zertifiziert Anwender B, B zertifiziert A und C usw.) und macht damit aus einem Zertifizierungs-Baum ein Zertifizierungs-Netz, das sogenannte **Web-of-Trust**. Im Fall der direkten Authentifizierung bei OpenPGP haben Sie also die Möglichkeit, *ohne* die Zertifizierung von einer höheren Stelle verschlüsselte Daten und E-Mails auszutauschen. Dafür reicht es aus, wenn Sie der E-Mail-Adresse und dem dazugehörigen Zertifikat ihres Kommunikationspartners vertrauen.

Nähere Informationen zu Authentifizierungswegen, wie zum Beispiel dem Web-of-Trust oder den Zertifizierungsstellen, erhalten Sie später in Kapitel 16.

Fassen wir kurz zusammen...

Was bedeutet das für Sie?

- Sowohl **OpenPGP** als auch **S/MIME** kann Ihnen die notwendige Sicherheit bieten.
- Beide Verfahren sind **nicht kompatibel** miteinander. Sie bieten zwei separate Wege bei der Authentifizierung Ihrer geheimen Kommunikation. Man sagt, sie sind nicht interoperabel.
- **Gpg4win** als Freie Software ermöglicht Ihnen die bequeme **parallele** Nutzung beider Systeme.

Falls Ihnen all das etwas zuviel Informationen waren, machen Sie sich keine Sorgen: In den folgenden Kapiteln wird jeder Schritt von der Installation bis hin zur Verschlüsselung sowohl mit OpenPGP als auch mit S/MIME detailliert erklärt.

Die beiden nachfolgenden Symbole weisen Sie in diesem Kompendium auf spezifische Erklärungen zu OpenPGP bzw. S/MIME hin, so dass Sie immer schnell überblicken, welche Besonderheiten bei welchem Konzept zu beachten sind.



4. Sie installieren Gpg4win

Beginnen wir nun mit der Installation von Gpg4win. Beachten Sie, dass Sie dafür Administrator-Rechte auf Ihrem Windows-System benötigen.

Sollte bereits eine GnuPG basierte Anwendung, wie z.B. GnuPP, GnuPT, WinPT oder GnuPG Basics, auf Ihrem System installiert sein, so lesen sie jetzt bitte zuerst den Anhang D, um zu erfahren wie Sie Ihre vorhandenen Schlüssel und Zertifikate übernehmen können.

Falls Sie Gpg4win aus dem Internet heruntergeladen haben:

Klicken Sie bitte auf diese neu abgespeicherte Datei, die den Namen `gpg4win-1.9.9-svn972.exe` (oder höhere Versionsnummer) haben sollte. Achten Sie unbedingt darauf, dass Sie die Datei von einer vertrauenswürdigen Seite erhalten haben.

Falls Sie Gpg4win auf einer CD-ROM erhalten haben:

Legen Sie diese CD-ROM in das CD-ROM-Laufwerk Ihres PCs. Öffnen Sie Ihren „Arbeitsplatz“ und klicken Sie dort auf das CD-ROM-Icon mit dem Titel „Gpg4win“. Anschließend klicken Sie auf das Installations-Icon mit dem Titel „Gpg4win“.

Die weitere Installation ist dann identisch:

Die Frage, ob Sie das Programm installieren wollen, beantworten Sie mit [Ja].

Es begrüßt Sie dieser Dialog:

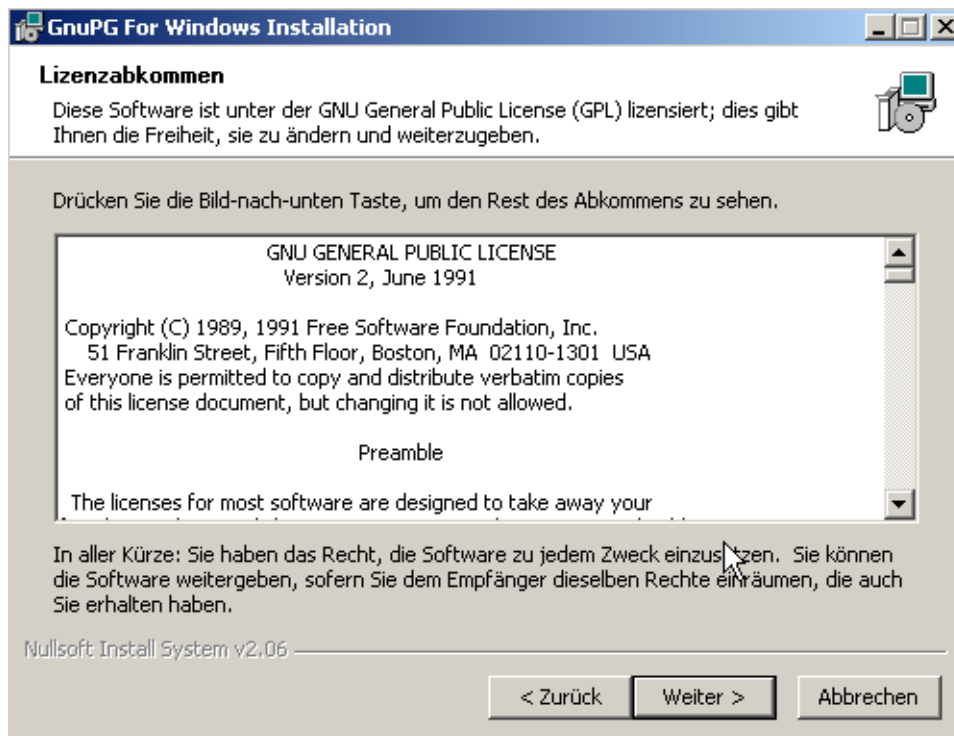


Beenden Sie alle auf Ihrem Rechner laufenden Programme, und klicken Sie dann auf [Weiter].

Auf der Seite mit dem **Lizenzabkommen**, können Sie Informationen zu den Lizenzen dieser Software lesen.

Wenn Sie die Software lediglich installieren und einsetzen wollen, so haben Sie immer das Recht dazu und sind nicht angehalten diese Texte zu lesen.

Geben Sie allerdings diese Software weiter oder wollen Sie sie verändern, so müssen Sie sich mit den Bedingungen der Lizenzen vertraut machen.

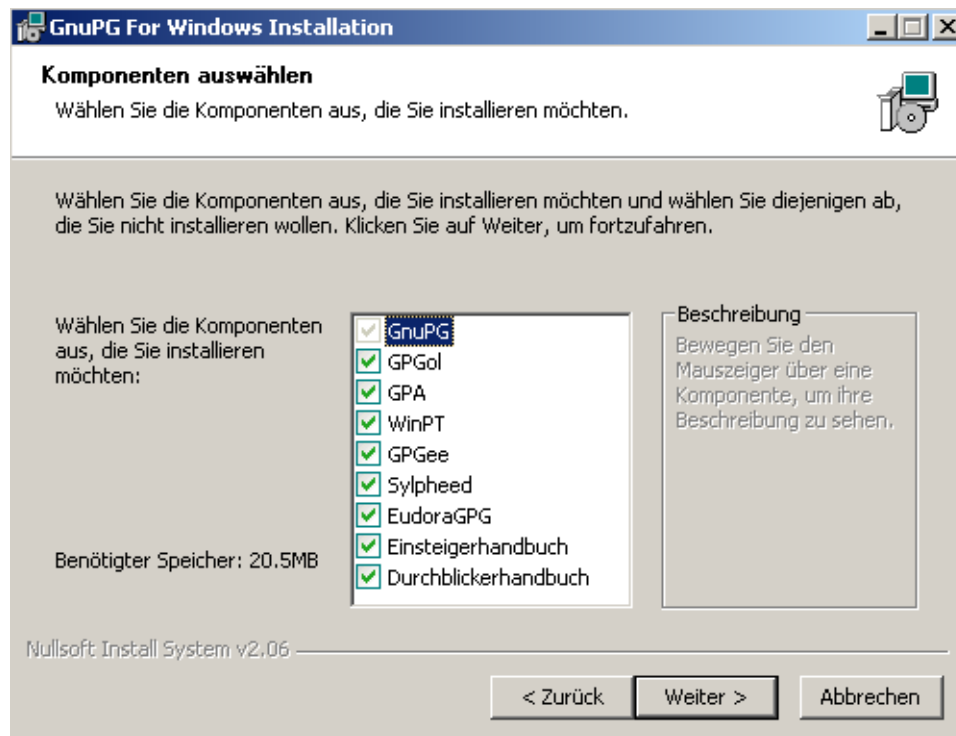


Klicken Sie auf [Weiter].

Auf der Seite mit der **Komponentenauswahl** können Sie entscheiden, welche Programme Sie installieren möchten.

Wenn Sie mit der Maus über die Auswahl laufen, dann erscheint jeweils rechts eine Kurzbeschreibung die Ihnen bei der Entscheidung hilft.

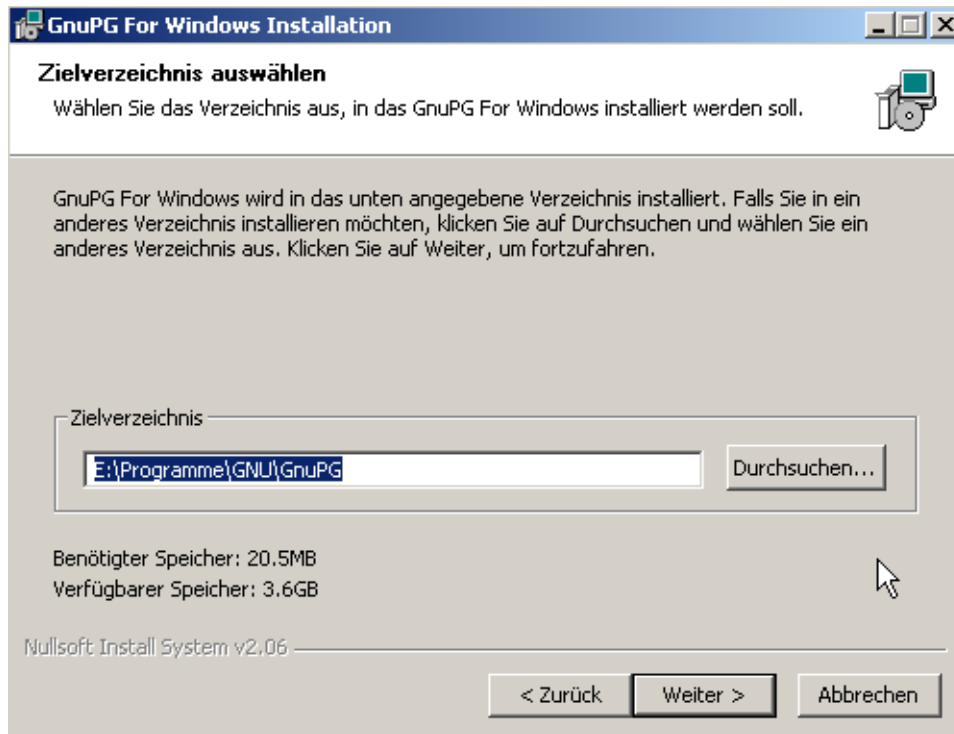
Die Anzeige des benötigten Speichers auf der Festplatte hilft Ihnen vielleicht ebenfalls weiter.



Alle sinnvollen Komponenten sind bereits vorausgewählt. Den Rest können Sie bei Bedarf auch später installieren.

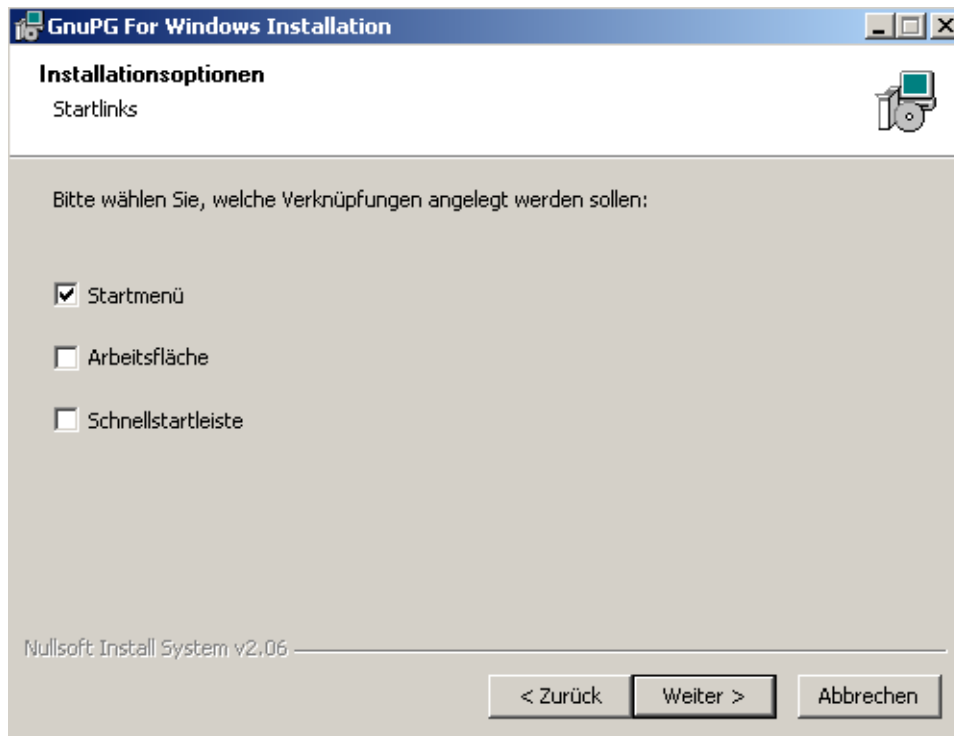
Klicken Sie auf [Weiter].

In der nun folgenden **Verzeichnisauswahl** können Sie einen Ordner auf Ihrem PC aussuchen, in dem Gpg4win installiert wird. Sie können hier in der Regel den vorgeschlagenen Programm-Ordner übernehmen; z.B.: C:\Programme\GNU\GnuPG.



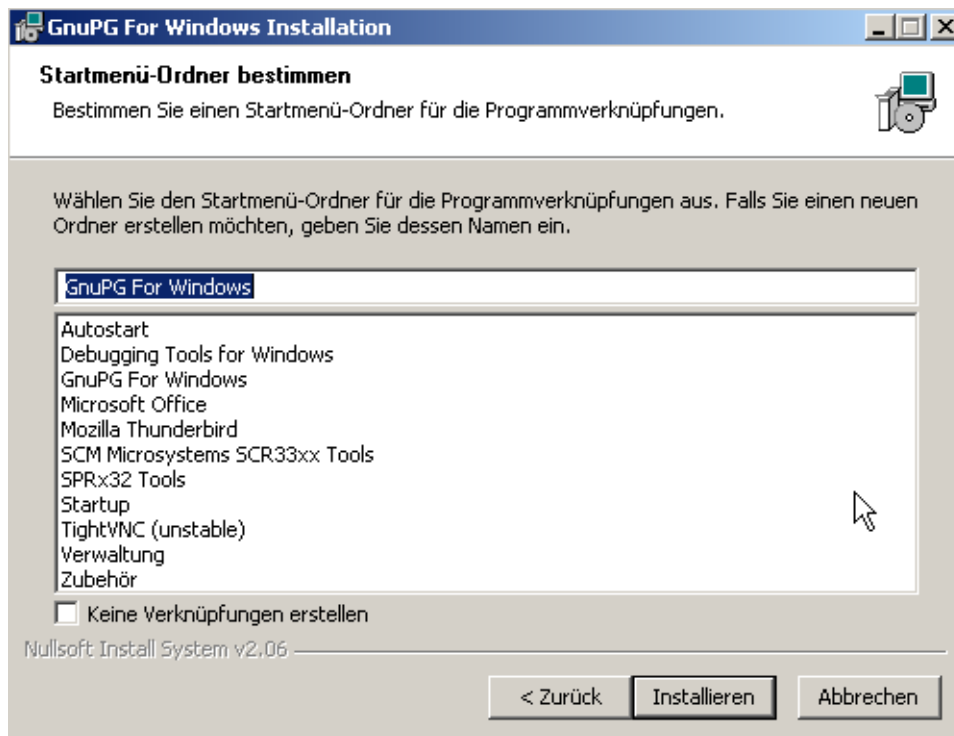
Klicken Sie anschließend auf [Weiter].

Auf der folgenden Seite können Sie festlegen, welche **Verknüpfungen** installiert werden. Voreingestellt ist lediglich eine Verknüpfung mit dem Startmenü. Bitte beachten Sie, daß sie diese Verknüpfungen auch jederzeit später mit den Bordmitteln von Windows verändern können.



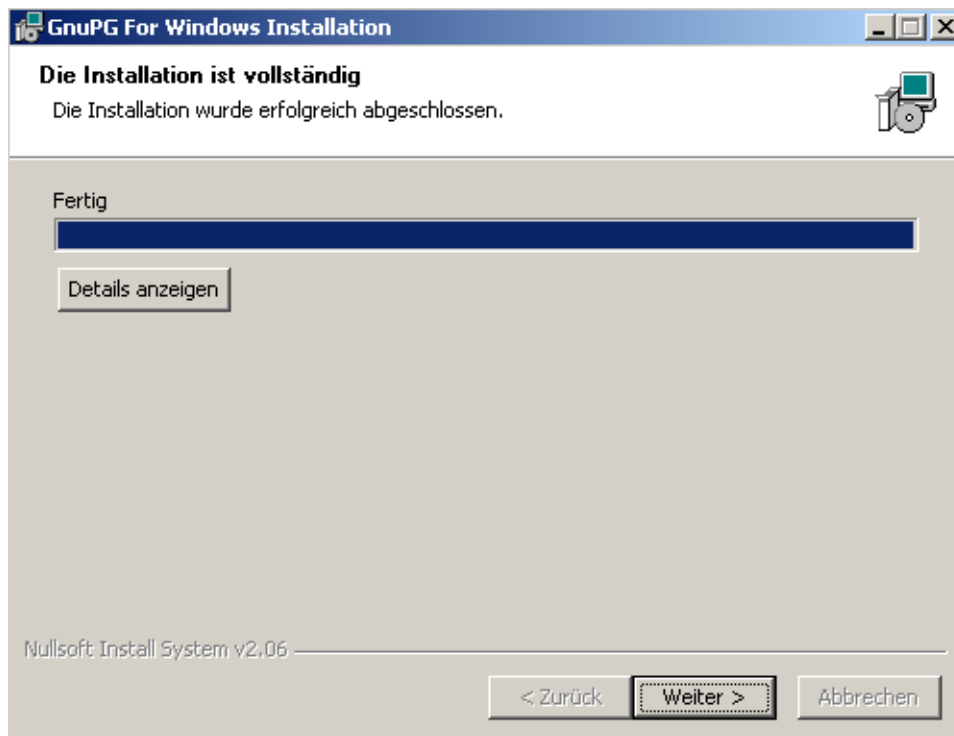
Klicken Sie anschließend auf [Weiter].

Falls Sie auf der vorhergehenden Seite eine **Verknüpfung mit dem Startmenü** ausgewählt haben (dies ist die Voreinstellung), so wird Ihnen nun eine Seite angezeigt, mit der Sie den Namen dieses Startmenüs auswählen können.



Am einfachsten übernehmen Sie die vorgeschlagene Einstellung und klicken dann auf [Installieren].

Während der nun folgenden **Installation** sehen Sie einen Fortschrittsbalken und Informationen, welche Datei momentan installiert wird. Sie können jederzeit auf [Details anzeigen] drücken, um ein Protokoll der Installation sichtbar zu machen.



Nachdem die Installation abgeschlossen ist, drücken Sie bitte auf [Weiter].

Nach erfolgreicher Installation wird Ihnen diese letzte Seite des Installationsvorgangs angezeigt:



Sofern Sie die README-Datei nicht ansehen wollen, deaktivieren Sie die Option auf dieser Seite. Klicken Sie schließlich auf [Fertig stellen].

In einigen Fällen kann es vorkommen, dass Windows neu gestartet werden muss. In diesem Fall sehen Sie statt der vorherigen die folgende Seite:



Sie können hier auswählen, ob Windows sofort neu gestartet werden soll oder später manuell. Klicken Sie hier auch auf [Fertig stellen].

Das war's schon!

Sie haben Gpg4win erfolgreich installiert und können es gleich zum ersten Mal starten.

Vorher sollten Sie aber Kapitel 12 lesen. Wir erklären dort den genialen Trick, mit dem Gpg4win Ihre E-Mails sicher und bequem verschlüsselt. Gpg4win funktioniert zwar auch, ohne dass Sie verstehen warum, aber im Gegensatz zu anderen Programmen wollen Sie Gpg4win schließlich Ihre geheime Korrespondenz anvertrauen. Da sollten Sie schon wissen, was vor sich geht.

Außerdem ist die ganze Angelegenheit ziemlich spannend. . .

Weiter geben wir Ihnen in Kapitel 13 einige Tipps, mit denen Sie sich einen sicheren und trotzdem leicht zu merkenden Passphrase ausdenken können.

Für Informationen zur **automatischen Installation** von Gpg4win (wie sie zum Beispiel für Softwareverteilungs-Systeme interessant ist), lesen Sie bitte im Anhang C „Automatische Installation von Gpg4win“ weiter.

5. Sie erzeugen Ihr Schlüsselpaar

Spätestens nachdem Sie gelesen haben, warum GnuPG eigentlich so sicher ist (Kapitel 24) und wie eine gute Passphrase als Schutz für Ihren geheimen Schlüssel entsteht (Kapitel 13), möchten Sie nun Ihr persönliches Schlüsselpaar erzeugen.

Ein Schlüsselpaar besteht, wie wir im Kapitel 3 gelernt haben, aus einem **Zertifikat** und einem **geheimen Schlüssel**. Das gilt sowohl für OpenPGP wie auch für S/MIME (die Schlüssel und Zertifikate entsprechen einem Standard mit der Bezeichnung X.509).

Eigentlich müsste man diesen wichtigen Schritt der Schlüsselpaarerstellung ein paar Mal üben können. . .

Genau das können Sie tun – und zwar für OpenPGP:

Sie können den gesamten Ablauf der Schlüsselerzeugung, Verschlüsselung und Entschlüsselung durchspielen, so oft Sie wollen, bis Sie ganz sicher sind.

Ihr Vertrauen in Gpg4win wird sich durch diese „Trockenübung“ festigen, und die „heisse Phase“ der OpenPGP-Schlüsselerzeugung wird danach kein Problem mehr sein.

Ihr Partner bei diesen Übungen wird **Adele** sein.

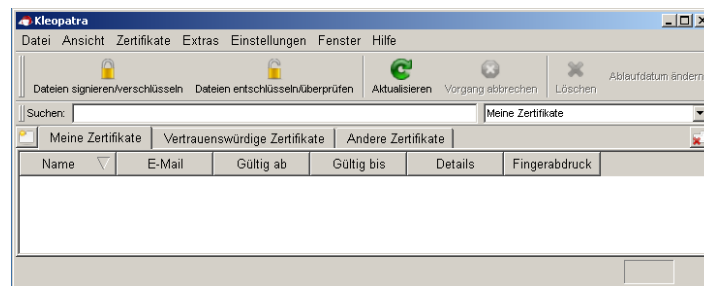
Adele ist ein Testservice, der noch aus dem alten GnuPP Projekt stammt, bis auf weiteres noch in Betrieb und natürlich auch für Gpg4win verwendet werden kann. Mit Hilfe von Adele können Sie Ihr OpenPGP-Schlüsselpaar, das wir gleich erzeugen werden, ausprobieren und testen, bevor Sie damit Ernst machen. Doch dazu später mehr.



Los geht's! Rufen Sie das Programm Kleopatra über das Windows-Startmenü auf:



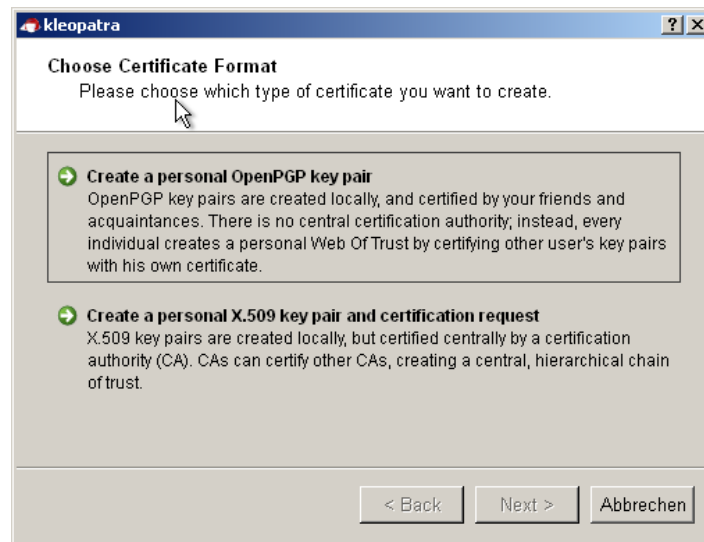
Daraufhin sehen Sie das Hauptfenster von Kleopatra – die Zertifikatsverwaltung:



Zu Beginn ist diese Übersicht leer, da wir noch keine Zertifikate oder Schlüssel erstellt haben. Dies wollen wir jetzt nachholen...

Klicken Sie auf *Datei*→*Neues Zertifikat*.

Im folgenden Dialog entscheiden Sie sich für ein Format, für das anschließend ein Zertifikat erstellt werden soll. Sie haben die Wahl: **OpenPGP** (PGP/MIME) oder **X.509** (S/MIME). Zu den Unterschieden lesen Sie bitte Kapitel 3 auf Seite 14.



Die weitere Vorgehensweise zum Erzeugen eines Schlüsselpaars gliedert sich an dieser Stelle in zwei Abschnitte:

- Abschnitt 5.1: **OpenPGP-Schlüsselpaar erstellen** (siehe nächste Seite) und
- Abschnitt 5.2: **X.509-Schlüsselpaar erstellen** (siehe Seite 36) .

Lesen Sie den entsprechenden Abschnitt weiter, für deren Zertifikatsformat Sie sich oben entschieden haben.

5.1. OpenPGP-Schlüsselpaar erstellen



Klicken Sie im obigen Auswahldialog auf [Create a personal OpenPGP key pair].
Geben Sie im nun folgenden Fenster Ihren Namen und Ihre E-Mail-Adresse an.

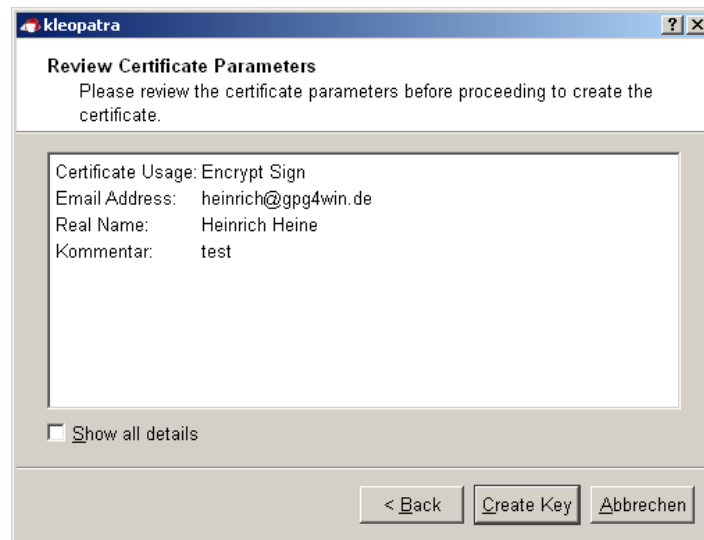
The screenshot shows a dialog box titled "kleopatra" with the subtitle "Enter Details". The main text reads: "Please enter your personal details below. If you want more control over the certificate parameters, click on the Advanced Settings button." Below this are three input fields: "Name:" with the value "Heinrich Heine" (required), "EMail:" with the value "heinrich@gpg4win.de" (required), and "Kommentar:" with the value "test" (optional). A summary line displays "Heinrich Heine (test) <heinrich@gpg4win.de>". At the bottom right is a button labeled "Advanced Settings...". At the very bottom are three buttons: "< Back", "Next >", and "Abbrechen".

Wenn Sie die OpenPGP-Schlüsselpaarerzeugung zunächst einmal **testen** wollen, dann können Sie einfach einen beliebigen Namen und irgendeine ausgedachte E-Mail-Adresse eingeben, z.B.:
Heinrich Heine und heinrichh@gpg4win.de.

Optional können Sie einen Kommentar zum Schlüssel eingeben. Normalerweise bleibt dieses Feld leer; wenn sie aber einen Testschlüssel erzeugen, sollten Sie dort als Erinnerung „test“ eingeben. Dieser Kommentar ist Teil Ihrer User-ID und genau wie der Name und die E-Mail-Adresse später öffentlich sichtbar.

Die **erweiterten Einstellungen** benötigen Sie nur in Ausnahmefällen. Sie können sich im Kleopatra Handbuch (über *Hilfe*→*Kleopatra Handbuch*) über die Details informieren.

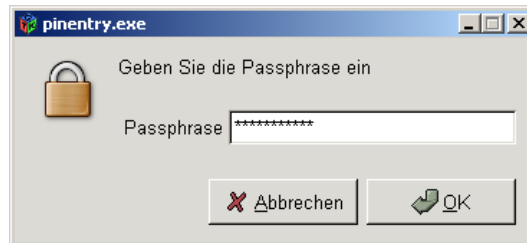
Es werden nun noch einmal alle wesentlichen Eingaben und Einstellungen zur **Kontrolle** aufgelistet. Falls Sie sich für die (voreingestellten) Experten-Einstellungen interessieren, können Sie diese über die Option *Alle Details* einsehen.



Sofern alles korrekt ist, klicken Sie anschließend auf [Create Key].

Jetzt folgt der wichtigste Teil: Die Eingabe Ihrer **Passphrase**!

Während der Schlüsselgenerierung werden Sie aufgefordert Ihre persönliche Passphrase einzugeben:



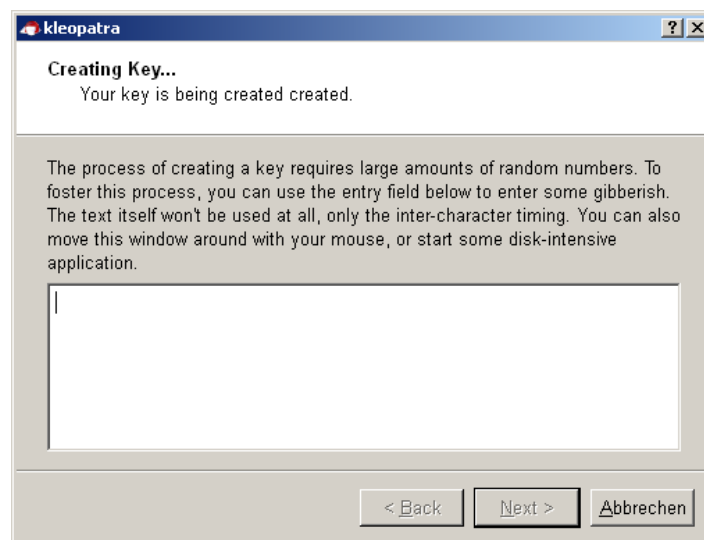
Im Kapitel 13 (Seite 89) geben wir Ihnen einige wertvolle Tipps, was Sie bei der Erzeugung einer **sicheren Passphrase** beachten sollten. Nehmen Sie die Sicherheit Ihrer Passphrase ernst!

Sie sollten nun eine geheime, einfach zu merkende und schwer zu knackende Passphrase parat haben und im obigen Dialog eintragen.

Auch an dieser Stelle können Sie – wenn Sie wollen – zunächst eine **Test-Passphrase** eingeben oder auch gleich „Ernst machen“.

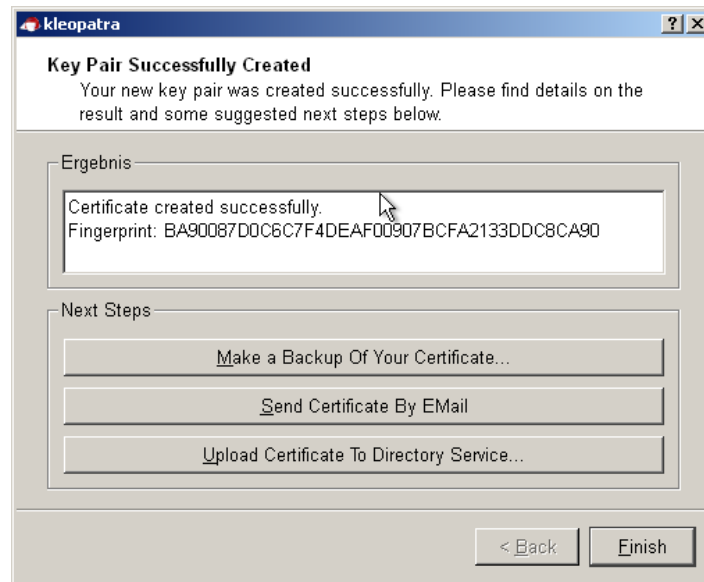
Sie müssen Ihre geheime Passphrase zweimal eingeben. Bestätigen Sie Ihre Eingabe jeweils mit [OK].

Nun wird Ihr OpenPGP-Schlüsselpaar angelegt:



Dies kann u.U. einige Minuten dauern. Sie können in der Zwischenzeit mit einer anderen Anwendung Ihres Rechner weiterarbeiten und erhöhen hierdurch sogar leicht die Qualität des erzeugten Schlüsselpaars.

Sobald die **Schlüsselpaargenerierung erfolgreich** abgeschlossen ist, erhalten Sie folgenden Dialog:



Im Ergebnis-Textfeld wird der 40-stelligen Fingerabdruck Ihres neu generierten OpenPGP-Schlüsselpaars angezeigt. Dieser sogenannte Fingerprint ist weltweit eindeutig, d.h. keine andere Person besitzt einen Schlüssel mit identischem Fingerabdruck. Es ist sogar vielmehr so, dass es schon mit 8 Zeichen ein ausserordentlicher Zufall wäre wenn diese weltweit ein zweites mal vorkämen. Daher werden oft nur die letzten 8 Zeichen des Fingerabdrucks verwendet bzw. angezeigt.

Sie brauchen sich die Zeichenkette nicht zu merken oder abzuschreiben. In den Zertifikatsdetails von Kleopatra können Sie sich diese jederzeit später anzeigen lassen.

Als nächstes können Sie eine (oder mehrere) der folgenden drei Möglichkeiten durchführen:

Erstellen Sie eine Sicherungskopie Ihres geheimen(!) OpenPGP-Schlüsselpaares.

Klicken Sie dazu auf die Schaltfläche [Sicherungskopie Ihres Zertifikats erstellen...]

Geben Sie hier den Pfad an, wohin Ihr geheimer Schlüssel exportiert werden soll:



Kleopatra wählt automatisch den Dateityp und speichert Ihr Schlüsselpaar entweder als *.asc bzw. *.gpg Datei ab – abhängig davon, ob Sie die die Option *ASCII-Mantel* ein- bzw. ausschalten.

Klicken Sie anschließend zum Exportieren auf [OK].

Wichtig: Falls Sie die Datei auf der Festplatte abgespeichert haben, so sollten Sie baldmöglichst diese Datei auf einen anderen Datenträger (USB Stick, Diskette oder CDROM) kopieren und diese Originaldatei löschen. Bewahren Sie diesen Datenträger sicher auf.

Sie können eine Sicherungskopie auch jederzeit später anlegen; wählen Sie hierzu aus dem Kleopatra-Hauptmenü: *Datei*→*Geheimen Schlüssel exportieren...* (vgl. Kapitel 19).

Versenden Sie Ihr erstelltes öffentliches(!) Zertifikat per E-Mail.

Klicken Sie auf die Schaltfläche [Zertifikat per E-Mail versenden].

Es sollte dabei eine neue E-Mail erstellt werden – mit Ihrem neuen Zertifikat im Anhang. Ihr geheimer OpenPGP-Schlüssel wird selbstverständlich *nicht* versendet. Geben Sie eine Empfänger-E-Mail-Adresse an und ergänzen Sie ggf. den vorbereiteten Text dieser E-Mail.

Beachten Sie: Nicht alle E-Mail-Programme unterstützen diese Funktion. Sollte kein neues E-Mail-Fenster aufgehen, so beenden Sie den Zertifikatserstellungsdialog, speichern Ihr *öffentliches* Zertifikat durch *Datei*→*Zertifikat exportieren* und versenden diese Datei per E-Mail an Ihre Korrespondenzpartner (Details im Abschnitt 6.1).

Speichern Sie Ihren neuen Schlüssel im Verzeichnisdienst.

Klicken Sie auf [Zertifikate zu Verzeichnisdienste senden...] und folgen Sie den Anweisungen. Sie müssen dafür vorher ein Verzeichnisdienst in Kleopatra konfiguriert haben.

Wie Sie Ihr OpenPGP-Zertifikat auf einen weltweit verfügbaren Keyserver veröffentlichen, erfahren Sie in Kaptel 15.

Beenden Sie anschließend den Kleopatra-Assistenten mit [Finish], um die Erstellung Ihres OpenPGP-Schlüsselpaares abzuschließen.

Weiter geht's mit dem Abschnitt *Schlüsselpaar-Erstellung abgeschlossen* auf Seite 41. Von da an sind die Erklärungen für OpenPGP und X.509 wieder identisch.

5.2. X.509-Schlüsselpaar erstellen

Klicken Sie im Zertifikatsformat-Auswahldialog von Seite 30 auf die Schaltfläche [Persönliches X.509-Schlüsselpaar und Beglaubigungs-Anfrage erstellen].

S/MIME

Geben Sie im nun folgenden Fenster Ihren Namen (CN), Ihre E-Mail-Adresse (EMAIL), Ihre Organisation (O) und Ihren Ländercode (C) an. Optional können Sie noch Ort (L) und Abteilung (OU) ergänzen.

kleopatra

Enter Details
Please enter your personal details below. If you want more control over the certificate parameters, click on the Advanced Settings button.

Allgemeiner Name (CN): Heinrich Heine (required)
E-Mail-Adresse (EMAIL): heinrich@gpg4win.de (required)
Ort (L): (optional)
Abteilung (OU): (optional)
Organisation (O): Test (required)
Ländercode (C): DE (required)

CN=Heinrich Heine,O=Test,C=DE

Add email address to DN (only needed for broken CAs)

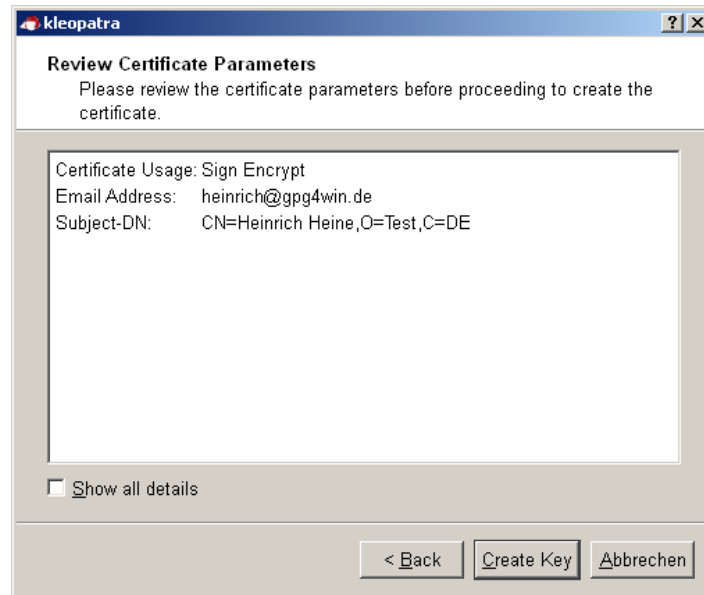
Advanced Settings...

< Back Next > Abbrechen

Wenn Sie die X.509-Schlüsselpaarerstellung zunächst einmal **testen** wollen, dann machen Sie beliebige Angaben für Name, Organisation und Ländercode sowie geben irgendeine ausgedachte E-Mail-Adresse ein, z.B. CN=Heinrich Heine, O=Test, C=DE und heinrichh@gpg4win.de.

Die **erweiterten Einstellungen** benötigen Sie nur in Ausnahmefällen. Sie können Sie im Kleopatra Handbuch (über *Hilfe*→*Kleopatra Handbuch*) über die Details informieren.

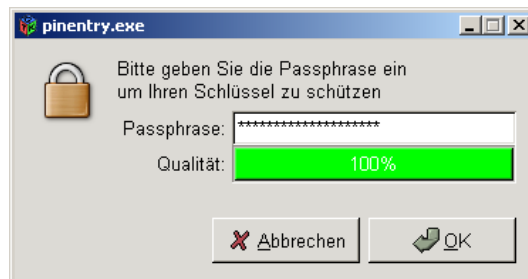
Es werden nun noch einmal alle Eingaben und Einstellungen zur **Kontrolle** aufgelistet. Falls Sie sich für die (voreingestellten) Experten-Einstellungen interessieren, können Sie diese über die Option *Alle Details* einsehen.



Sofern alles korrekt ist, klicken Sie anschließend auf [Create Key].

Jetzt folgt der wichtigste Teil: Die Eingabe Ihrer **Passphrase**!

Während der Schlüsselgenerierung werden Sie aufgefordert Ihre Passphrase einzugeben:



Im Kapitel 13, Seite 89, geben wir Ihnen einige wertvolle Tipps, was Sie bei der Erzeugung einer **sicheren Passphrase** beachten sollten. Nehmen Sie die Sicherheit Ihrer Passphrase ernst!

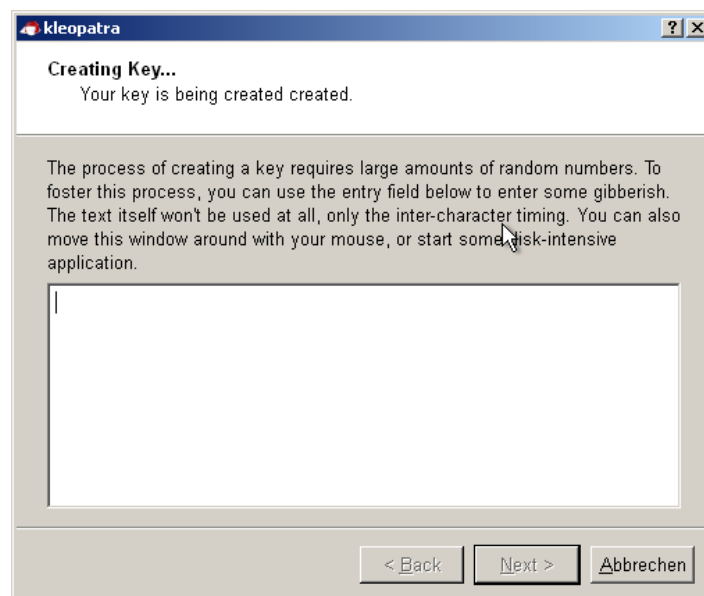
Sie sollten nun eine geheime, einfach zu merkende und schwer zu knackende Passphrase parat haben und im obigen Dialog eintragen.

Falls die Passphrase nicht sicher genug sein sollte (z.B. weil sie zu kurz ist oder keine Zahlen/Sonderzeichen enthält), werden Sie darauf hingewiesen.

Auch an dieser Stelle können Sie – wenn Sie wollen – zunächst eine **Test-Passphrase** eingeben oder auch gleich „Ernst machen“.

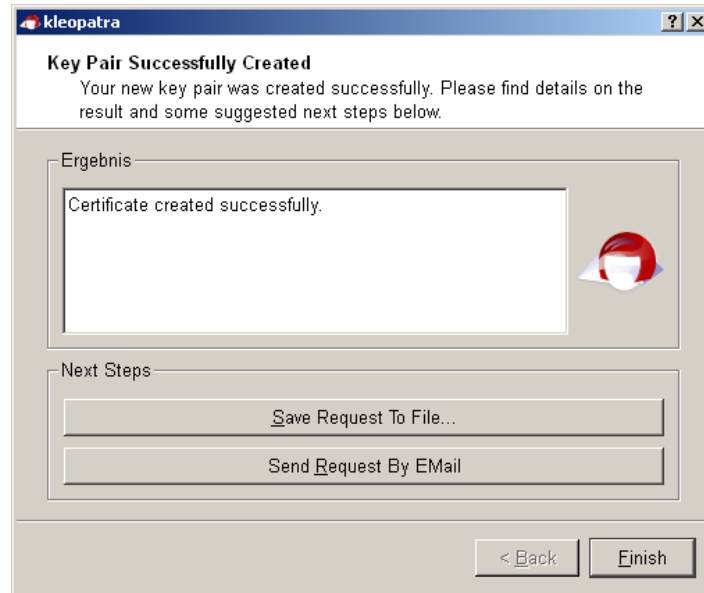
Sie müssen Ihre geheime Passphrase dreimal eingegeben. Bestätigen Sie Ihre Eingabe jeweils mit [OK].

Nun wird Ihr X.509-Schlüsselpaar angelegt:



Dies kann u.U. einige Minuten dauern. Sie können in der Zwischenzeit mit einer anderen Anwendung Ihres Rechner weiterarbeiten und erhöhen hierdurch sogar leicht die Qualität des erzeugten Schlüsselpaars.

Sobald die **Schlüsselpaargenerierung erfolgreich** abgeschlossen ist, erhalten Sie folgenden Dialog:



Als nächstes *können* Sie eine (oder mehrere) der folgenden drei Möglichkeiten durchführen:

Speichern Sie Ihre Zertifizierungs-Anfrage als Datei.

Klicken Sie dazu auf die Schaltfläche [Anfrage in Datei speichern...]

Geben Sie den genauen Pfad an, wohin Ihre X.509 Zertifizierungs-Anfrage gespeichert werden soll und bestätigen Sie Ihre Eingabe. Kleopatra fügt beim Speichern automatisch die Dateierendung *.p10 hinzu. Sie können diese Datei dann später auf verschiedene Weise an eine Zertifizierungsstelle geben.

Versenden Sie die Zertifizierungs-Anfrage per E-Mail.

Klicken Sie auf die Schaltfläche [Anfrage per E-Mail versenden].

Es wird eine neue E-Mail erstellt – mit der soeben erstellen Zertifizierungs-Anfrage im Anhang. Geben Sie eine Empfänger-E-Mail-Adresse an (in der Regel die Ihrer zuständigen Zertifizierungsstelle (CA)) und ergänzen Sie ggf. den vorbereiteten Text dieser E-Mail.

Beachten Sie: Nicht alle E-Mail-Programme unterstützen diese Funktion. Sollte kein neues E-Mail-Fenster aufgehen, so speichern Sie Ihre Anfrage zunächst in eine Datei (siehe oben) und versenden diese Datei per E-Mail an Ihre Zertifizierungsstelle.

Sobald der Request von der CA bestätigt wurde, erhalten Sie von Ihrem zuständigen CA-Systemadministrator das fertige und unterzeichnete X.509-Zertifikat. Dieses müssen Sie dann nur noch in Kleopatra importieren (vgl. Kapitel 19).

Beenden Sie anschließend den Kleopatra-Assistenten mit [Finish].

Erstellung eines X.509-Schlüsselpaars mit www.cacert.org

CAcert ist eine gemeinschaftsbetriebene, nicht-kommerzielle Zertifizierungsstelle (CA), die kostenlos X.509-Zertifikate ausstellt.

Damit Sie sich ein (Client-)Zertifikat bei CAcert erstellen können, müssen Sie sich zunächst unter www.cacert.org registrieren.

Anschließend können Sie sich mit Ihrem CAcert-Account ein (oder mehrere) Client-Zertifikat(e) erstellen: Sie sollten dabei auf eine hohe Schlüsselgröße achten. In dem startenden Assistenten legen Sie Ihre sichere Passphrase für Ihr Zertifikat fest.

Ihre X.509-Zertifikatsanfrage wird nun erstellt.

Im Anschluß daran erhalten Sie eine E-Mail mit zwei Links zu Ihrem neu erstellten X.509-Zertifikat und dem dazugehörigen CAcert-Root-Zertifikat. Laden Sie sich beide Zertifikate herunter.

Folgen Sie den Anweisungen und installieren Ihr Zertifikat mit Ihrem Browser. Mit Firefox können Sie danach z.B. über *Bearbeiten*→*Einstellungen*→*Erweitert*→*Zertifikate* Ihr installiertes Zertifikat unter dem ersten Reiter „Ihre Zertifikate“ mit dem Namen *CAcert WoT User* finden.

Wenn Sie sich von anderen Mitglieder des CAcert-Web-of-Trust bestätigen lassen, können Sie auch personalisierte Zertifikate mit Ihrem Namen (CN) ausstellen.

Speichern Sie abschließend eine Sicherungskopie Ihres X.509-Zertifikatspaars in einer *.p12 Datei. **Achtung:** Diese *.p12 Datei enthält Ihr (öffentliches) Zertifikat und Ihren zugehörigen *geheimen* Schlüssel. Achten Sie darauf, dass diese Datei nicht in unbefugte Hände gelangt.

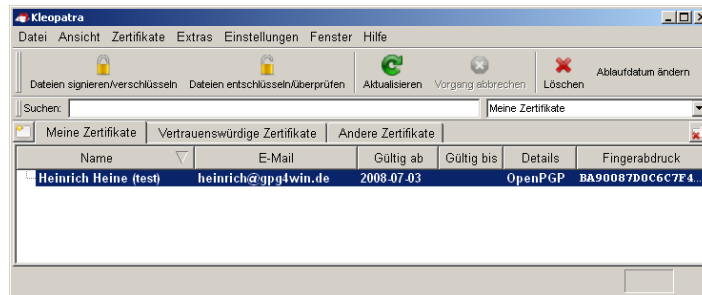
Wie Sie Ihr X.509-Schlüsselpaar in Kleopatra importieren erfahren Sie in Kapitel 19.

Weiter geht's mit dem Abschnitt *Schlüsselpaar-Erstellung abgeschlossen* auf der nächsten Seite. Von nun an sind die Erklärungen für OpenPGP und X.509 wieder identisch.

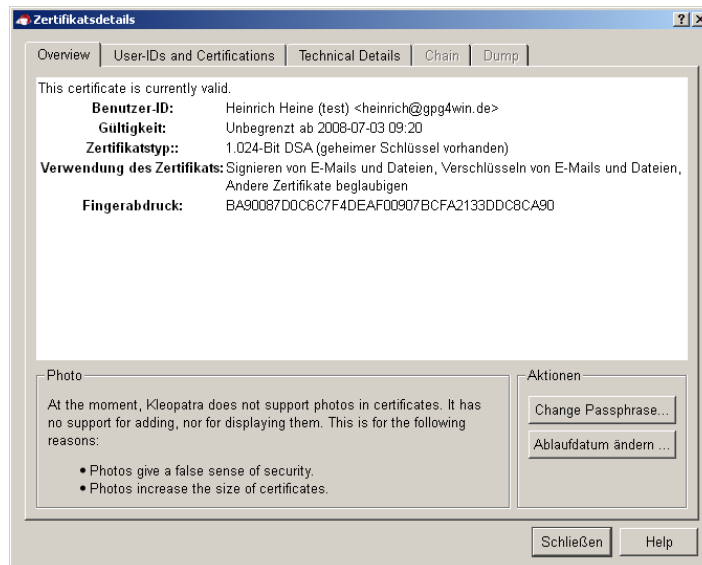
5.3. Schlüsselpaar-Erstellung abgeschlossen

Damit ist die Erzeugung Ihres OpenPGP- bzw. X.509-Schlüsselpaares abgeschlossen. Sie besitzen nun einen einmaligen und sicheren digitalen Schlüssel.

Sie sehen jetzt wieder das Hauptfenster von Kleopatra. Das soeben erzeugte OpenPGP-/X.509-Schlüsselpaar finden Sie in der Zertifikatsverwaltung unter dem Reiter *Meine Zertifikate* (hier und im weiteren wird exemplarisch ein OpenPGP-Zertifikat verwendet):



Doppelklicken Sie auf Ihr neues Zertifikat, um alle Zertifikatsdetails nachlesen zu können:



Was bedeuten die einzelnen Zertifikatsdetails?

Ihr Schlüssel ist unbegrenzt gültig d.h., er hat kein „eingebautes Verfallsdatum“. Um die Gültigkeit nachträglich zu verändern, klicken Sie auf [Ablaufdatum ändern].

Ein Schlüssel mit einer Länge von 1024 Bit ist ein sicherer Schlüssel, der trotzdem nicht zuviel Rechenkraft auf Ihrem Computer beansprucht.

Weitere Informationen zu den Zertifikatsdetails finden Sie im Kapitel 14. Sie können dieses Kapitel jetzt lesen oder später, wenn Sie diese Informationen benötigen.

6. Sie veröffentlichen Ihr Zertifikat

Beim täglichen Gebrauch von Gpg4win ist es sehr praktisch, dass Sie es beim Ver- und Entschlüsseln stets nur mit dem „ungeheimen“ Zertifikat (Ihren öffentlichen Schlüssel) zu tun haben. Solange Ihr eigener geheimer Schlüssel und die ihn schützende Passphrase sicher sind, haben Sie das Wichtigste zur Geheimhaltung bereits erledigt.

Jedermann darf und soll Ihr Zertifikat haben, und Sie können und sollen Zertifikate von Ihren Korrespondenzpartnern haben – je mehr, desto besser.

Denn:

Um sichere E-Mails austauschen zu können, müssen beide Partner jeweils das Zertifikat des anderen besitzen und benutzen. Natürlich braucht der Empfänger auch ein Programm, das mit den Zertifikaten umgehen kann, wie zum Beispiel Gpg4win.

Wenn Sie also an jemanden verschlüsselte E-Mails schicken wollen, müssen Sie dessen Zertifikat haben und zum Verschlüsseln benutzen.

Wenn – andersherum – jemand Ihnen verschlüsselte E-Mails schicken will, muss er Ihr Zertifikat haben und zum Verschlüsseln benutzen.

Deshalb werden Sie nun Ihr Zertifikat öffentlich zugänglich machen. Je nachdem, wie groß der Kreis Ihrer Korrespondenzpartner ist und welches Zertifikatsformat Sie einsetzen, gibt es verschiedene Möglichkeiten. Verbreiten Sie Ihr Zertifikat beispielsweise ...

- ... direkt per **E-Mail** an bestimmte Korrespondenzpartner (vgl. Abschnitt 6.1).
- ... auf einem **OpenPGP-Schlüsselserver**; gilt *nur* für OpenPGP (vgl. Abschnitt 6.2).
- ... über die eigene Homepage.
- ... persönlich, z.B. per USB-Stick.

Die ersten beiden Varianten werden wir uns auf den folgenden Seiten näher anschauen.

6.1. Veröffentlichen per E-Mail

Sie wollen Ihr Zertifikat Ihrem Korrespondenzpartner bekannt machen? Schicken Sie ihm doch einfach ihr exportiertes Zertifikat per E-Mail. Wie das genau funktioniert, erfahren Sie in diesem Abschnitt.

Üben Sie jetzt diesen Vorgang einmal mit Ihrem OpenPGP-Zertifikat! Adele soll uns dabei behilflich sein. *Achtung: Die folgenden Übungen gelten nur für OpenPGP! Anmerkungen zur Veröffentlichung von X.509-Zertifikaten finden Sie auf Seite 47.*



Adele ist ein sehr netter E-Mail-Roboter, mit dem Sie zwanglos korrespondieren können. Weil man gewöhnlich mit einer klugen und netten jungen Dame lieber korrespondiert als mit einem Stück Software (was Adele in Wirklichkeit natürlich ist), haben wir sie uns so vorgestellt:



Adele schicken Sie zunächst Ihr OpenPGP-Zertifikat. Wenn Adele diesen öffentlichen Schlüssel empfangen hat, verschlüsselt sie damit eine E-Mail an Sie und sendet sie zurück.

Diese Antwort von Adele entschlüsseln Sie mit Ihrem eigenen geheimen Schlüssel. Damit Sie wiederum Adele verschlüsselt antworten können, legt Adele ihren eigenen öffentlichen Schlüssel bei.

Adele verhält sich also genau wie ein richtiger Korrespondenzpartner. Allerdings sind Adeles E-Mails leider bei weitem nicht so interessant wie die Ihrer echten Korrespondenzpartner. Andererseits können Sie mit Adele so oft üben, wie Sie wollen – was Ihnen ein menschlicher Adressat wahrscheinlich ziemlich übel nehmen würde.

Wir exportieren also nun Ihr OpenPGP-Zertifikat und senden dieses per E-Mail an Adele. Wie das geht, erfahren Sie auf den nächsten Seiten.

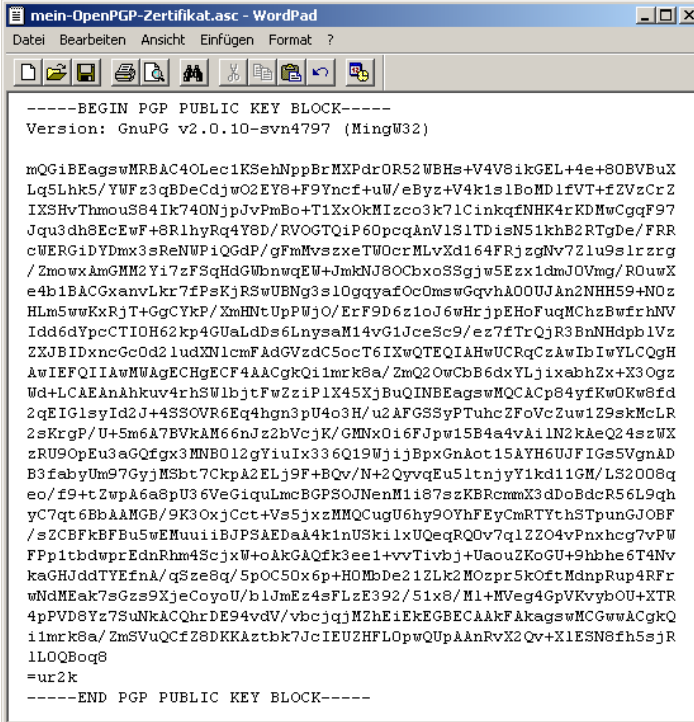
Exportieren Ihres OpenPGP-Zertifikats

Selektieren Sie in Kleopatra das zu exportierende Zertifikat (durch Klicken auf die entsprechende Zeile in der Liste der Zertifikate) und klicken Sie dann auf *Datei*→*Zertifikate exportieren ...* im Menü. Wählen Sie einen geeigneten Ordner auf Ihrem PC aus und speichern Sie das Zertifikat im Dateityp *.asc ab – z.B.: mein-OpenPGP-Zertifikat.asc.

Wichtig: Achten Sie beim Auswählen des Menüpunktes darauf, dass Sie auch wirklich nur Ihr (öffentliches) Zertifikat exportieren – und *nicht* aus Versehen Ihren zugehörigen geheimen Schlüssel exportieren.

Sehen Sie sich zur Kontrolle diese Datei an. Nutzen Sie dazu Ihren Windows Explorer und wählen denselben Ordner aus, den Sie beim Exportieren angegeben haben.

Öffnen Sie die exportierte Zertifikats-Datei mit einem Texteditor, z.B. mit WordPad. Sie sehen Ihr OpenPGP-Zertifikat im Texteditor so, wie es wirklich aussieht – ein ziemlich wirrer Text- und Zahlenblock:



```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.10-svn4797 (MingW32)

mQGIBeagswMRBAC4OLec1KSehNppBrMXPdrOR52WBHs+V4V8ikGEL+4e+80EBVuX
Lq5Lhk5/YWFz3qBDeCdJwO2EY8+F9Yncf+uW/eByz+V4k1s1BoMD1fVT+fZVzCrZ
IXSHvThmouS84Ik74ONjppJvPmBo+T1XxOkMIzco3k71CinkqfNHR4rKDMwCgqf97
Jqu3dh8EeEwf+8R1hyRq4Y8D/RVOGTQiP6OpcqAnV1S1TDIsN51khB2RTgDe/FRR
cWERGIDYDmx3sReNWPiQGdP/gFmMvszxeTWOcrMLvXd164FRjzgnV7Z1u9s1rzg
/ZmowxAmGMM2Yi7zFSqHdGwbnwqEW+JmkNJ80CbxxoSSgJwSEzx1dmJOVmg/ROUwX
e4b1BACGxanvLkr7fPsKjRSwUBNg3s1OgqyafOcOmswGqvhA00UJAn2NHH59+NOz
Hlm5wwKXrjT+GgCYkP/XmHntUpPwjo/ErF9D6z1oJ6wHrjpeHoFuqMChzBwfrhNV
Idd6dYpcCTIOH62kp4GUaLdDs6LnysaM14vG1JceSc9/ez7fTrQjR3EnNHdpp1Vz
ZZJBIDxncGcOd21udXXN1cmFAdGVzdc5ocT6IXwQTEQIAHwUCRqCzAwIbIwYLCQgH
AwIEFQIIAwMwAgECHgECP4AAcGkQi1mrk8a/ZmQ2OwCbB6dxYLjixabhZx+X3Ogz
Wd+LCAEAnAhkuv4rhSW1bjtFwZziP1X45XjBuQINBeagswMQCACp84yfrw0Kw8fd
2qEIG1syId2J+4SSOVR6Eq4hgn3pU4o3H/u2AFGSSyPTuhcZFOvcZuw1Z9skMcLR
2sKrgP/U+Sm6A7BVkAM66njz2bVcjK/GMNx0i6FJpw15B4a4vAi1N2kAeQ24szWX
zRU9OpEu3aGQfgx3MNB012gyIuX336Q19WjijBpxGnAot15AYH6UJFIGS5VgnAD
B3fabYUm97GyjMSbt7CkpA2ELj9f+BQv/N+2QyvqEu51tnjyY1kd11GM/LS2008q
eo/f9+tzwpA6a8pU36VeG1quLmcBGPSOJNenM1i87szKBRcmmX3dDoBdcR56L9qh
yC7qt6BbAAMGB/9K3OxjCct+Vs5jzMMQCugU6hy90YhFEyCmRTYthSTpunGJOBf
/sZCBfkBFBU5wEHuuiiBJPSAEDA4k1nUSk1lxUQeqRQ0v7q1ZZO4vPnxhcg7vPW
FP1tbdwprEdnRhm4ScjxW+oAkGAQfk3ee1+vvTivbj+UaouZKoGU+9hbhe6T4Nv
kaGHJddTYEfnA/qSze8q/5pOC50x6p+HOMbDe21ZLk2MOzpr5kOfTmdnpRup4RFR
wNdMEak7sGzs9XjeCoyoU/blJmEz4sFLzE392/51x8/M1+MVEg4GpVKvybOU+XTR
4pFVd8Yz7SUnkACQhrDE94vdV/vbcjqjMzhEiEkEGBECAAkFAkagswMCGwWAkCgkQ
i1mrk8a/ZmSVuQCfZ8DKKAZtbk7JcIEUZHFL0pwQUpAAAnRvX2Qv+X1ESN8fh5sJR
lLOQBoq8
=ur2k
-----END PGP PUBLIC KEY BLOCK-----
```

Variante 1: OpenPGP-Zertifikat als E-Mail-Text versenden

Die hier zuerst gezeigte Möglichkeit funktioniert immer, selbst wenn Sie – z.B. bei manchen E-Mail-Services im Web – keine Dateien anhängen können. Zudem bekommen Sie so Ihr Zertifikat zum ersten Mal zu Gesicht und wissen, was sich dahinter verbirgt und woraus der Schlüssel eigentlich besteht.

Markieren Sie nun im Texteditor den gesamten öffentlichen Schlüssel von

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

bis

```
-----END PGP PUBLIC KEY BLOCK-----
```

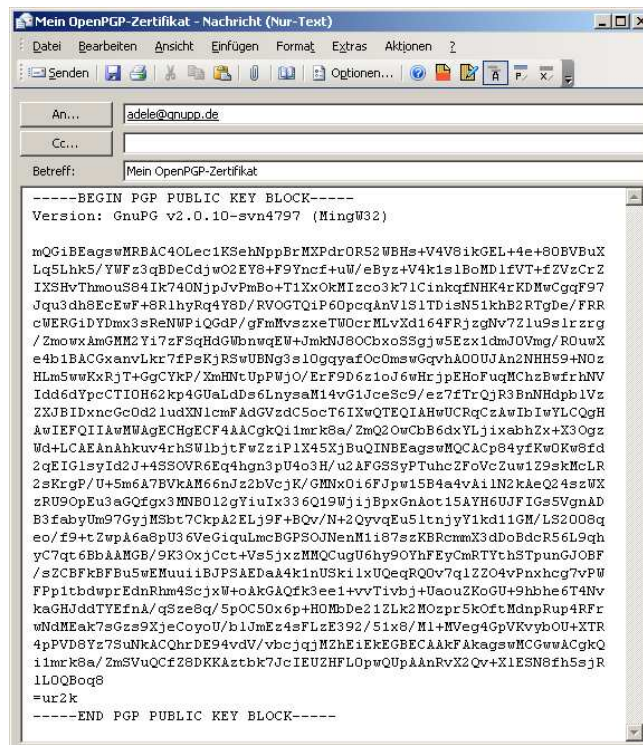
und **kopieren** Sie ihn mit dem Menübefehl oder mit dem Tastaturkürzel Strg+C. Damit haben Sie den Schlüssel in den Speicher Ihres Rechners (bei Windows Zwischenablage genannt) kopiert.

Nun starten Sie Ihr Mailprogramm – es spielt keine Rolle, welches Sie benutzen – und fügen Ihr Zertifikat in eine leere E-Mail ein. Der Tastaturbefehl zum Einfügen („Paste“) lautet bei Windows Strg+V. Es ist sinnvoll vorher das Mailprogramm so zu konfigurieren, dass reine Textnachrichten gesendet werden und keine HTML formatierte Nachrichten.

Diesen Vorgang – Kopieren und Einfügen – kennen Sie sicher als „Copy & Paste“.

Adressieren Sie nun diese E-Mail an `adele@gnupp.de` und schreiben in die Betreffzeile z.B.: *Mein OpenPGP-Zertifikat*.

So etwa sollte Ihre E-Mail nun aussehen:



Schicken Sie die E-Mail an Adele ab.

Nur zur Vorsicht: Natürlich sollten Ihre E-Mails *nicht* `heinrichh@gpg4win.de` oder ein andere Beispieladresse als Absender haben, sondern *Ihre eigene E-Mail-Adresse*. Denn sonst werden Sie nie Antwort von Adele bekommen. . .

Variante 2: OpenPGP-Zertifikat als E-Mail-Anhang versenden

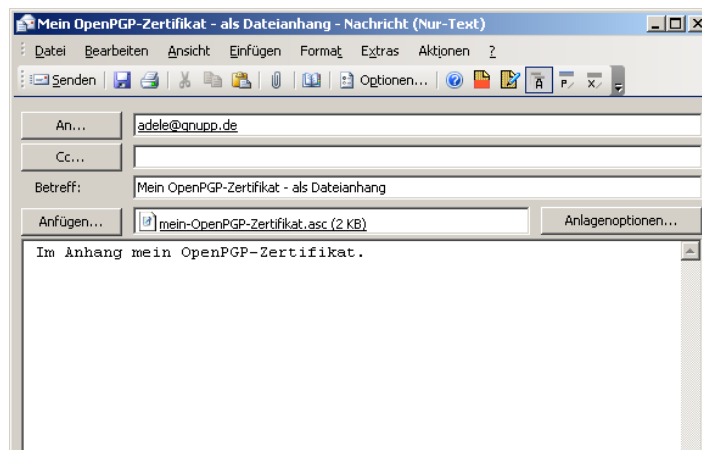
Alternativ zu Variante 1 können Sie natürlich Ihr exportiertes OpenPGP-Zertifikat auch direkt als **E-Mail-Dateianhang** versenden. Das ist oftmals das einfachere und gebräuchlichere Verfahren. Wir haben Ihnen oben die „Copy & Paste“-Methode zuerst vorgestellt, weil sie transparenter und leichter nachzuvollziehen ist.

Schreiben wir Adele nun noch einmal eine neue Mail mit der Zertifikatsdatei im Anhang:

Fügen Sie die exportierte Zertifikatsdatei als Anhang zu Ihrer neuen E-Mail hinzu – genauso wie Sie es mit jeder anderen Datei auch machen (z.B. durch Ziehen der Datei in das leere E-Mail-Fenster). Ergänzen Sie den Empfänger (`adele@gnupp.de`) und einen Betreff, z.B. *Mein OpenPGP-Zertifikat - als Dateianhang*.

Selbstverständlich dürfen Sie auch noch ein paar erklärende Sätze dazuschreiben. Adele braucht diese Erklärung jedoch nicht, denn sie ist zu nichts anderem als zu diesem Übungszweck programmiert worden.

Ihre fertige E-Mail sollte dann etwa so aussehen:



Senden Sie nun die E-Mail mit Anhang an Adele ab.

Fassen wir kurz zusammen...

Sie haben Ihr OpenPGP-Zertifikat in Kleopatra in eine Datei exportiert. Anschließend haben wir einmal den Inhalt der Datei direkt in eine E-Mail kopiert und einmal die komplette Datei als E-Mail-Anhang eingefügt. Beide E-Mails haben wir an einen Korrespondenzpartner (in unserem Fall Adele) geschickt.

Genauso gehen Sie vor, wenn Sie Ihr Zertifikat an eine echte E-Mail-Adresse senden. Sie entscheiden sich dabei natürlich für eine der beiden oben vorgestellten Varianten – in der Regel sollten Sie Ihr OpenPGP-Zertifikat per Dateianhang versenden. Dies ist für Sie und Ihren Empfänger das Einfachste. Und es hat den Vorteil, dass Ihr Empfänger Ihre Zertifikatsdatei direkt (ohne Umwege) in seine Zertifikatsverwaltung (z.B. Kleopatra) importieren kann.

Nachdem Sie gelernt haben, wie Sie Ihr OpenPGP-Zertifikat per E-Mail veröffentlichen, wird Sie sicher interessieren wie das Gleiche für **X.509-Zertifikate** funktioniert (vgl. auch Kapitel 3).



Die Antwort lautet: Genauso wie bei OpenPGP! Sie exportieren Ihr X.509-Zertifikat in Kleopatra, speichern dieses z.B. im Dateiformat **.pem* ab und versenden die Datei als E-Mail-Anhang.

Der einzige Unterschied zum oben beschriebenen OpenPGP-Vorgehen: Sie können Adele nicht benutzen! **Adele unterstützt nur OpenPGP!** Zum Üben sollten Sie sich also einen anderen Korrespondenzpartner aussuchen oder Sie schreiben testweise an sich selber.

Beim Exportieren Ihres X.509-Zertifikats haben Sie die Wahl, ob Sie die ganze (öffentliche) Zertifikatskette (in der Regel: Wurzel – Zertifizierungsstelle – Ihr Zertifikat) oder *nur* Ihr Zertifikat in eine Datei abspeichern wollen. Ersteres ist empfehlenswert, denn Ihrem Korrespondenzpartner fehlen möglicherweise Teile der Kette die er sonst zusammensuchen müsste. Klicken Sie dazu in Kleopatra alle Kettenelemente mit gedrückter Shift-Taste an und exportieren Sie diese markierten Elemente nach oben beschriebener Regel.

Hatte Ihr Korrespondenzpartner das Wurzel-Zertifikat noch nicht, so muss er dieser Wurzel das Vertrauen aussprechen bzw. durch einen Administrator aussprechen lassen um letztlich auch Ihnen zu vertrauen. Ist das bereits vorher geschehen (z.B. weil sie beide zur selben Wurzel gehören, selbst bei unterschiedlichen Zertifizierungsstellen), dann besteht das Vertrauen unmittelbar mit der Verfügbarkeit der Kette.

6.2. Veröffentlichen per Schlüsselservers

Wichtig: Die Veröffentlichung Ihres Zertifikats auf einem Schlüsselservers (Keyservers) ist nur für OpenPGP-Zertifikate möglich!



Die Veröffentlichung Ihres OpenPGP-Zertifikats auf einem internationalen Schlüsselservers bietet sich eigentlich immer an, selbst wenn Sie nur mit wenigen Partnern verschlüsselte E-Mails austauschen. Ihr Zertifikat ist dann für jedermann zugänglich auf einem Servers im Internet verfügbar. Sie ersparen sich dadurch die Versendung Ihres Zertifikats per E-Mail an jeden Ihrer Korrespondenzpartner.

VORSICHT: DIE VERÖFFENTLICHUNG IHRER E-MAIL-ADRESSE AUF EINEM KEYSERVER BIRGT LEIDER DAS RISIKO, DASS IHNEN AUCH UNGEBETENE PERSONEN E-MAILS SCHREIBEN KÖNNEN UND DIE SPAM-MENGE FÜR IHRE E-MAIL-ADRESSE DADURCH ZUNEHMEN KANN. SIE SOLLTEN DAHER IM ZWEITEN FALL EINEN AUSREICHENDEN SPAM-SCHUTZ NUTZEN. FALLS SIE KEINEN WIRKSAMEN SPAMFILTER BENUTZEN, SOLLTEN SIE U.U. VON DER VERÖFFENTLICHUNG IHRES SCHLÜSSELS AUF EINEM KEYSERVER ABSEHEN.

Und so geht's: Wählen Sie Ihr OpenPGP-Zertifikat in Kleopatra aus und klicken im Menü auf *Datei*→*Zertifikate nach Servers exportieren...*

Sofern Sie noch keinen Schlüsselservers definiert haben, bekommen Sie eine Warnmeldung:



Wie Sie an der Meldung erkennen können, ist der öffentliche Schlüsselservers `keys.gnupg.net` bereits voreingestellt. Klicken Sie auf [Fortsetzen], um Ihren ausgewählten Schlüssel an diesen Servers zu schicken. Von dort aus wird Ihr Schlüssel an alle, weltweit verbundenen Keyservers weitergereicht. Jedermann kann Ihren Schlüssel dann von einem dieser Keyservers herunterladen und dazu benutzen, Ihnen eine sichere E-Mail zu schreiben.

Wenn Sie den Ablauf im Moment nur testen, dann schicken Sie den Übungsschlüssel bitte nicht ab. Er ist wertlos und kann nicht mehr vom Schlüsselservers entfernt werden. Sie glauben nicht, wieviele Testkeys mit Namen wie „Julius Caesar“, „Helmut Kohl“ oder „Bill Clinton“ dort schon seit Jahren herumliegen. . .

Fassen wir kurz zusammen...

Sie wissen nun, wie Sie Ihr OpenPGP-Zertifikat auf einem Schlüsselservers im Internet veröffentlichen.

Wie Sie das OpenPGP-Zertifikat eines Korrespondenzpartners auf Schlüsselserversn suchen und importieren, beschreiben wir im Kapitel 15. Sie können dieses Kapitel jetzt lesen oder später, wenn Sie diese Funktion benötigen.

Die Veröffentlichung von X.509-Zertifikaten erfolgt in einigen Fällen durch die Zertifizierungsstelle. Das passiert typischerweise über LDAP-Server. Im Unterschied zu den OpenPGP-Schlüsselserversn synchronisieren sich die LDAP-Server jedoch nicht weltweit untereinander.

S/MIME

7. Sie entschlüsseln eine E-Mail

Sie bekommen verschlüsselte Nachrichten Ihrer Korrespondenzpartner und wollen diese nun entschlüsseln? Alles was Sie dazu brauchen ist Gpg4win, Ihr Schlüsselpaar und natürlich ganz wichtig: Ihre Passphrase.

In diesem Kapitel erklären wir Ihnen Schritt für Schritt, wie Sie Ihre E-Mails in Microsoft Outlook mit Gpg4win entschlüsseln.

Wir üben jetzt diesen Vorgang einmal mit Adele und Ihrem OpenPGP-Zertifikat!

Achtung: Die folgenden Übungen gelten nur für OpenPGP! Anmerkungen zur Entschlüsselung von S/MIME-E-Mails finden Sie am Ende dieses Kapitels auf Seite 53.



Im Abschnitt 6.1 haben Sie Adele Ihr OpenPGP-Zertifikat geschickt. Mit Hilfe dieses Zertifikats verschlüsselt Adele nun eine E-Mail und sendet sie an Sie zurück. Nach kurzer Zeit sollten Sie Adeles Antwort erhalten.



Nachricht mit MS Outlook und GpgOL entschlüsseln

Für die meisten Mailprogramme gibt es spezielle Erweiterungen (sogenannte Plugins), mit denen die Ver- und Entschlüsselung direkt im jeweiligen Mailprogramm erledigt werden kann. – **GpgOL** ist ein solches Plugin für MS Outlook, dass wir in diesem Abschnitt nutzen wollen, um die E-Mail von Adele zu entschlüsseln.

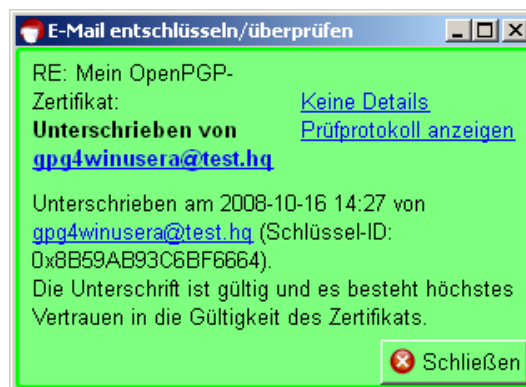
Hinweise zu weiteren Software-Lösungen finden Sie im Kapitel B. Sie können dieses Kapitel jetzt lesen oder später, wenn Sie diese Funktion benötigen.

Starten Sie MS Outlook und öffnen Sie die Antwort-E-Mail von Adele.

Kleopatra haben wir bisher nur als Zertifikatsverwaltung kennengelernt. Das Programm leistet aber weitaus mehr: Es kann die eigentliche Verschlüsselungs-Software GnuPG steuern und damit nicht nur Schlüssel und Zertifikate verwalten, sondern auch sämtliche kryptografischen Aufgaben (eben mit Hilfe von GnuPG) erledigen. Kleopatra sorgt für die graphische Benutzeroberfläche, also die Dialoge die Sie als Benutzer sehen während Sie eine E-Mail ver- oder entschlüsseln. Das heisst auch, dass Sie immer die gleichen Dialog sehen, egal ob Sie mit Outlook, einem anderen E-Mail-Programm oder auch mit dem Windows Explorer etwas verschlüsseln.

Kleopatra bearbeitet also die verschlüsselte E-Mail von Adele. Diese E-Mail hat Adele mit *Ihrem* OpenPGP-Zertifikat verschlüsselt.

Um die Nachricht zu entschlüsseln, fragt Kleopatra Sie nun nach Ihrer (zum Schlüsselpaar gehörigen) Passphrase. Geben Sie diese in den aufkommenden Dialog ein. Sofern Ihre Eingabe korrekt war, erhalten Sie einen Statusdialog (siehe nachfolgende Abbildung). Mit [Details einblenden] können Sie sich weitere Informationen der E-Mail-Überprüfung anzeigen lassen.



Die Entschlüsselung war erfolgreich! Schließen Sie den Dialog, um die entschlüsselte E-Mail zu lesen.

Möchten Sie den Prüfdialog nach dem Lesen der Mail noch einmal manuell aufrufen, so klicken Sie im Menü der geöffneten E-Mail auf *Extras*→*GpgOL Entschlüsseln/Prüfen*.

Doch nun wollen wir das Ergebnis, die entschlüsselte Nachricht, endlich einmal sehen...

Die entschlüsselte Nachricht

Die entschlüsselte Antwort von Adele sieht in etwa so aus¹:

Hallo Heinrich Heine,

hier ist die verschlüsselte Antwort auf Ihre E-Mail.

Ihr öffentlicher Schlüssel mit der Schlüssel-ID
BCFA2133DDC8CA90 und der Bezeichnung
'Heinrich Heine <heinrichh@duesseldorf.de>'
wurde von mir empfangen.

Anbei der öffentliche Schlüssel von adele@gnupp.de,
dem freundlichen E-Mail-Roboter.

Viele Grüße,
adele@gnupp.de


Der Textblock, der darauf folgt, ist das Zertifikat von Adele.

Wir werden im nächsten Kapitel dieses Zertifikat importieren und an Ihrem Schlüsselbund befestigen. Importierte Zertifikate können Sie jederzeit zum Verschlüsseln von Nachrichten an Ihren Korrespondenzpartner benutzen oder dessen signierte Mails überprüfen.

¹Abhängig von der Softwareversion von Adele kann dies auch etwas unterschiedlich aussehen.

Fassen wir kurz zusammen...

1. Sie haben eine verschlüsselte E-Mail mit Ihrem geheimen Schlüssel entschlüsselt.
2. Der Korrespondenzpartner hat sein eigenes Zertifikat beigelegt, damit Sie ihm verschlüsselt antworten können.

Nachdem Sie gelernt haben, wie Sie E-Mails mit Ihrem OpenPGP-Zertifikat entschlüsseln, werden Sie nun noch erfahren, wie Sie verschlüsselte **S/MIME**-E-Mails entschlüsseln. 

Die Antwort lautet auch hier: Genauso wie bei OpenPGP! Der Unterschied zu OpenPGP ist lediglich, dass S/MIME *nicht* von Adele unterstützt wird und somit die obige Übung nur für OpenPGP gilt.

Zum Entschlüsseln einer S/MIME-verschlüsselten E-Mail öffnen Sie die Nachricht in Outlook und geben im aufgehenden Dialog Ihre Passphrase ein. Sie bekommen einen ähnlichen Statusdialog wie bei OpenPGP. Nach dem Schließen dieses Dialogs sehen Sie die entschlüsselte S/MIME E-Mail.

8. Sie importieren ein Zertifikat

Ihr Korrespondenzpartner muss nicht jedes Mal sein Zertifikat mitschicken, wenn er Ihnen signiert schreibt. Sie bewahren seinen öffentlichen Schlüssel einfach an Ihrem GnuPG-„Schlüsselbund“ (oder besser: „Zertifikatsbund“) auf.

Zertifikat abspeichern

Bevor Sie ein Zertifikat in Kleopatra importieren, müssen Sie es in einer Datei abspeichern. Abhängig davon, ob Sie das Zertifikat als E-Mail-Dateianhang oder als Textblock innerhalb Ihrer E-Mail bekommen haben, gehen Sie wie folgt vor:

- Liegt das Zertifikat einer E-Mail als **Dateianhang** bei, speichern Sie es (wie Sie es in Ihrem Mailprogramm gewohnt sind) auf einem Ort Ihrer Festplatte ab.
- Sollte das Zertifikat als **Textblock** innerhalb Ihrer E-Mail übermittelt worden sein, so müssen Sie zunächst das vollständige Zertifikat markieren:

Bei OpenPGP-Zertifikaten markieren Sie den Bereich von

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

bis

```
-----END PGP PUBLIC KEY BLOCK-----
```

so wie Sie es im Abschnitt 6.1 schon getan haben.

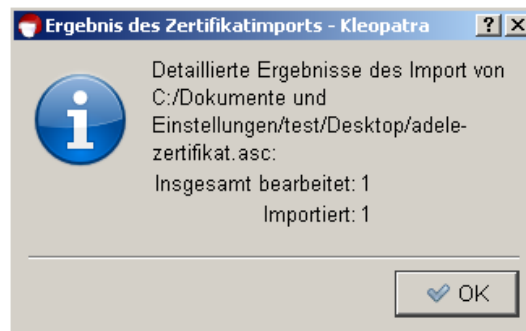
Setzen Sie den markierten Abschnitt per Copy & Paste in einen Texteditor ein und speichern Sie das Zertifikat ab. Als Dateiendung sollten Sie für OpenPGP-Zertifikate **.asc* und für X.509-Zertifikate z.B. **.pem* wählen.

Zertifikat in Kleopatra importieren

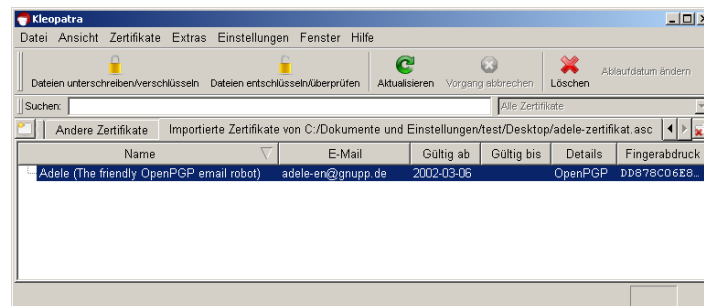
Ob Sie nun das Zertifikat als E-Mail-Anhang oder als Textblock abgespeichert haben, ist egal: In beiden Fällen importieren Sie dieses abgespeicherte Zertifikat in Ihre Zertifikatsverwaltung **Kleopatra**.

Starten Sie dafür Kleopatra, sofern das Programm noch nicht läuft.

Klicken Sie im Menü auf *Datei*→*Zertifikat importieren...*, suchen das eben abgespeicherte Zertifikat aus und laden es. Sie erhalten einen Informations-Dialog mit dem Ergebnis des Importvorgangs:



Das erfolgreich importierte Zertifikat wird nun in Kleopatra angezeigt – und zwar unter einem separaten Reiter „Importierte Zertifikate von <Pfad-zur-Zertifikatsdatei>“:



Dieser Reiter dient zur Kontrolle, weil einer Datei durchaus auch mehr als nur ein Zertifikat enthalten kann. Schließen Sie diesen Reiter mit dem rot-weißen Schließen-Button am Rechten Fensterrand.

Wechseln Sie auf den Tab „Andere Zertifikate“. Hier sollten Sie nun das von Ihnen importierte Zertifikat sehen.

Damit haben Sie ein fremdes Zertifikat – in diesem Beispiel das OpenPGP-Zertifikat von Adele – importiert und an Ihrem Schlüsselbund befestigt. Sie können dieses Zertifikat jederzeit benutzen, um verschlüsselte Nachrichten an den Besitzer dieses Zertifikats zu senden und Signaturen zu prüfen.

Sobald Sie E-Mail-Verschlüsselung häufiger und mit vielen Korrespondenzpartnern betreiben, werden Sie aus Gründen des Komforts die Zertifikate über weltweite Verteilungsdienste suchen und importieren wollen. Wie das geht, können Sie im Kapitel 15 nachlesen.

Bevor wir weitermachen, eine wichtige Frage:

Woher wissen Sie eigentlich, dass das fremde OpenPGP-Zertifikat wirklich von Adele stammt? Man kann E-Mails auch unter falschem Namen versenden – die Absenderangabe besagt eigentlich gar nichts.

Wie können Sie also sichergehen, dass ein Zertifikat auch wirklich seinem Absender gehört?

Die Kernfrage der Zertifikatsprüfung erläutern wir im Kapitel 16. Lesen Sie bitte jetzt dort weiter, bevor Sie danach an dieser Stelle fortfahren.

9. Sie verschlüsseln eine E-Mail

Jetzt wird es noch einmal spannend: Wir versenden eine verschlüsselte E-Mail!

Sie brauchen dazu Outlook, Kleopatra und natürlich ein Zertifikat Ihres Korrespondenzpartners.

Erstellen Sie zunächst in Outlook eine neue E-Mail und adressieren Sie diese an Ihren Korrespondenzpartner.

Hinweis nur für OpenPGP:

Sie können zum Üben dieses Vorgangs mit OpenPGP wieder Adele nutzen. S/MIME wird von Adele nicht unterstützt! Adressieren Sie dazu Ihre zu verschlüsselnde E-Mail an adele@gnupp.de. Der Inhalt der Nachricht ist egal – Adele kann nicht wirklich lesen...



Protokoll bestimmen – PGP/MIME oder S/MIME

Bestimmen Sie nun das Protokoll – PGP/MIME oder S/MIME – mit der Sie Ihre Nachricht verschlüsseln wollen. Die hängt davon ab, in welchem Format das Zertifikat Ihres Korrespondenzpartners bei Ihnen vorliegt. Klicken Sie dazu im Menü *Extras*→*GnuPG Protokoll* des geöffneten Outlook-Nachrichtenfensters auf: *PGP/MIME*, *S/MIME* oder *automatisch*.

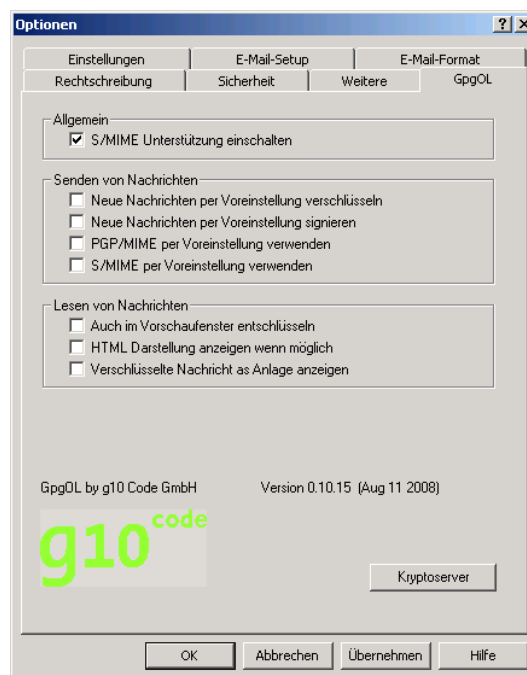
Sofern Sie diese Auswahl auf der Voreinstellung *automatisch* lassen, haben Sie später im Verschlüsselungsprozess noch die Möglichkeit zwischen PGP/MIME und S/MIME zu wählen.

Haben Sie ein bevorzugtes GnuPG-Protokoll? Dann können Sie unter den GpgOL-Optionen (*Extras*→*Optionen*→*GpgOL*) PGP/MIME oder S/MIME als Voreinstellung definieren.

Wichtiger Hinweis nur für S/MIME:

Nach der Installation von Gpg4win ist die S/MIME-Funktionalität in GpgOL deaktiviert. Wenn Sie S/MIME nutzen möchten, sollten Sie zuvor unter *Extras*→*Optionen*→*GpgOL* die Option *S/MIME Unterstützung einschalten* aktivieren:

S/MIME

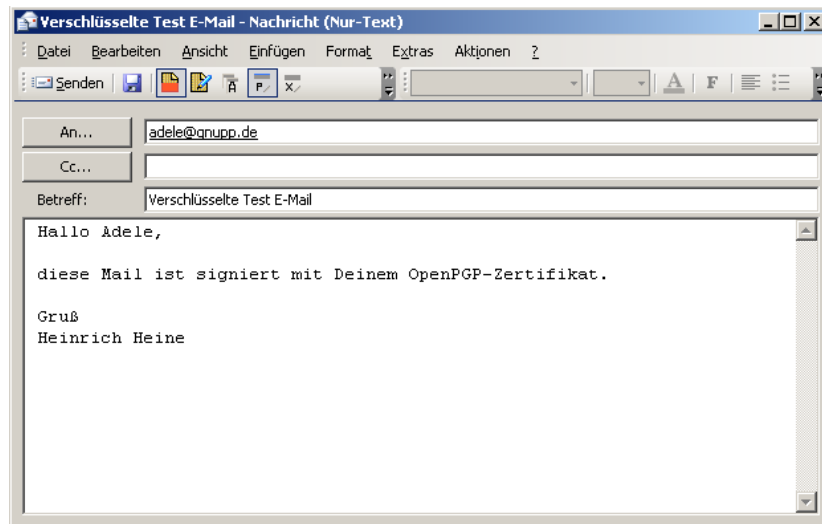


Lesen Sie sich die angezeigten Informationen sorgfältig durch, vor allem dann wenn Sie schon vorher mit S/MIME über ein anderes Plugin-Produkt gearbeitet haben.

Verschlüsselung aktivieren

Jetzt fehlt nur noch die Angabe, dass Sie Ihre Nachricht auch wirklich verschlüsselt versenden wollen: Wählen Sie *Extras*→*Nachricht mit GnuPG verschlüsseln*. Die Schaltfläche mit dem Schloss-Icon in der Symbolleiste ist aktiviert (Sie können auch gleich direkt auf diese Schaltfläche klicken).

Ihre Outlook-Nachrichtenfenster sollte nun etwa so aussehen:

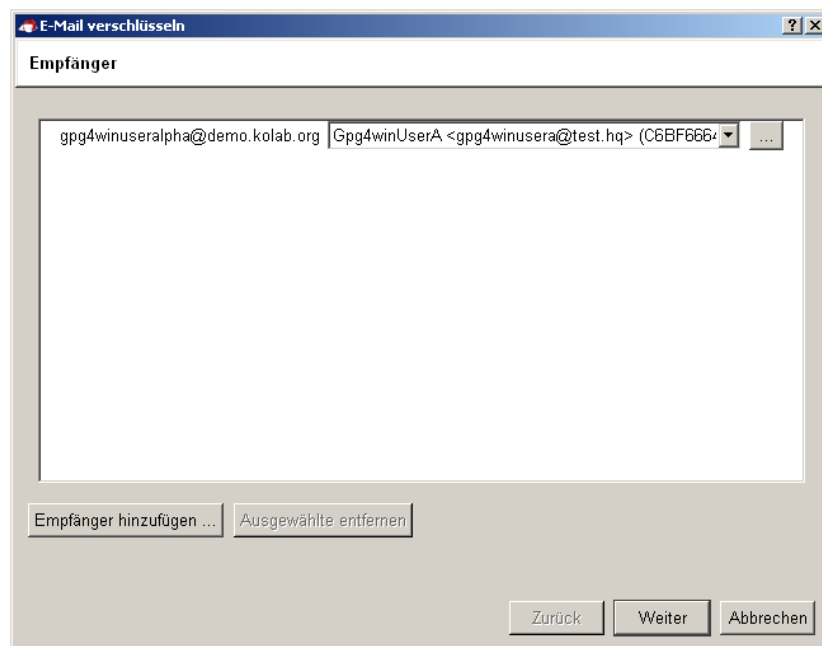


Um die Verschlüsselungsoption wieder zu deaktivieren genügt ein erneuter Klick auf die o.g. Schaltfläche.

Klicken Sie nun auf [Senden].

Zertifikatsauswahl

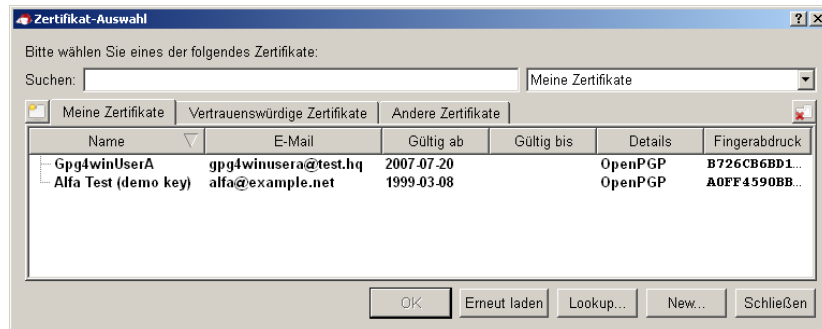
Daraufhin öffnet Kleopatra ein Fenster, in dem Sie das Zertifikat des Empfängers angeben. Kleopatra wählt – abhängig von der Empfänger-E-Mail-Adresse – in der Regel das passende Zertifikat automatisch aus.



Im Normalfall können Sie dieses vorausgewählte Zertifikat mit [Weiter] bestätigen.

Sollte es jedoch nicht das richtige Zertifikat sein – z.B. weil zu der E-Mail-Adresse mehrere Zertifikate existieren oder Sie bewusst ein anderes Zertifikat auswählen wollen – klicken Sie auf den [...]-Button neben der Drop-Down-Liste.

Sie bekommen einen Kleopatra-Dialog mit einer Auflistung aller Zertifikate des gewählten Zertifikatstyps, die in Ihrer Zertifikatsverwaltung existieren (also Zertifikate, die von Ihnen bis dahin importiert wurden). Exemplarisch sehen Sie hier eine Auswahl von verfügbaren OpenPGP-Zertifikaten:



Wählen Sie das korrekte Zertifikat Ihres Korrespondenzpartners aus, denn damit muss Ihre Nachricht ja verschlüsselt werden.

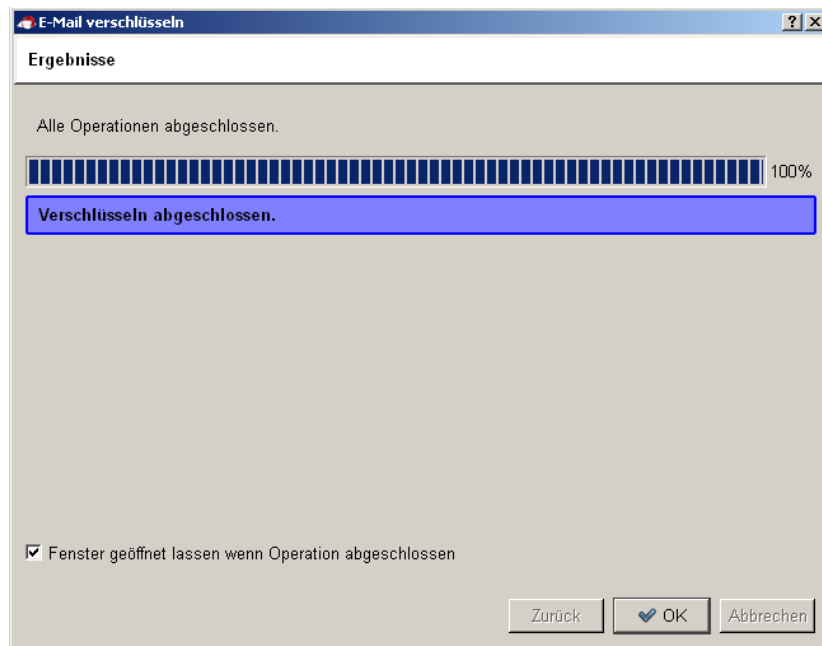
Sie erinnern sich an den Grundsatz:

Wenn Sie an jemanden verschlüsselte E-Mails schicken wollen, müssen Sie dessen Zertifikat haben und zum Verschlüsseln benutzen.

Klicken Sie auf [Weiter]. Ihre Nachricht wird nun verschlüsselt.

Verschlüsselung abschließen

Wurde Ihre Nachricht erfolgreich verschlüsselt und versandt, erhalten Sie eine Meldung, die Ihnen dies bestätigt:



Herzlichen Glückwunsch! Sie haben Ihre erste E-Mail verschlüsselt!

10. Sie signieren eine E-Mail

Sie haben in Kapitel 16 gelesen, wie Sie sich von der Echtheit eines OpenPGP-Zertifikats überzeugen und es dann mit Ihrem eigenen geheimen OpenPGP-Schlüssel signieren können.

In diesem Kapitel wollen wir uns damit beschäftigen, wie Sie nicht nur ein Zertifikat, sondern auch eine komplette **E-Mail signieren** können. Das bedeutet, dass Sie die E-Mail mit einer elektronischen Unterschrift (eine Art elektronisches Siegel) versehen.

Der Text ist dann zwar noch für jeden lesbar, aber Ihr Empfänger kann feststellen, ob die E-Mail unterwegs manipuliert oder verändert wurde.

Die Signatur garantiert Ihrem Empfänger, dass die Nachricht tatsächlich von Ihnen stammt. Und: wenn Sie mit jemandem korrespondieren, dessen Zertifikat Sie nicht haben (aus welchem Grund auch immer), können Sie so die Nachricht wenigstens mit Ihrem eigenen privaten Schlüssel „versiegeln“.

Achtung: Verwechseln Sie diese elektronische Signatur nicht mit der E-Mail-„Signatur“, die man unter eine E-Mail setzt und die zum Beispiel Ihre Telefonnummer, Ihre Adresse und Ihre Webseite enthalten. Während diese E-Mail-Signaturen einfach nur als eine Art Visitenkarte fungieren, schützt die elektronische Signatur Ihre E-Mail vor Manipulationen und bestätigt den Absender.

Übrigens ist die elektronische Unterschrift auch nicht mit der qualifizierten digitalen Signatur gleichzusetzen, wie sie im Signaturgesetz vom 22. Mai 2001 in Kraft getreten ist. Für die private oder berufliche E-Mail-Kommunikation erfüllt sie allerdings genau denselben Zweck.



10.1. Signieren mit GpgOL

Tatsächlich ist die Signierung einer E-Mail noch einfacher als die Verschlüsselung (vgl. Kapitel 9). Nachdem Sie eine neue E-Mail verfasst haben, gehen wir – analog zur Verschlüsselung – folgende Schritte durch:

- Protokoll bestimmen – PGP/MIME oder S/MIME
- Signierung aktivieren
- Zertifikatsauswahl
- Signierung abschließen

Auf den nächsten Seiten beschreiben wir diese Schritte im Detail.

Protokoll bestimmen – PGP/MIME oder S/MIME

Genauso wie beim Verschlüsseln von E-Mails müssen Sie vorher das Protokoll bestimmen, nach welchem Verfahren signiert bzw. verschlüsselt werden soll.

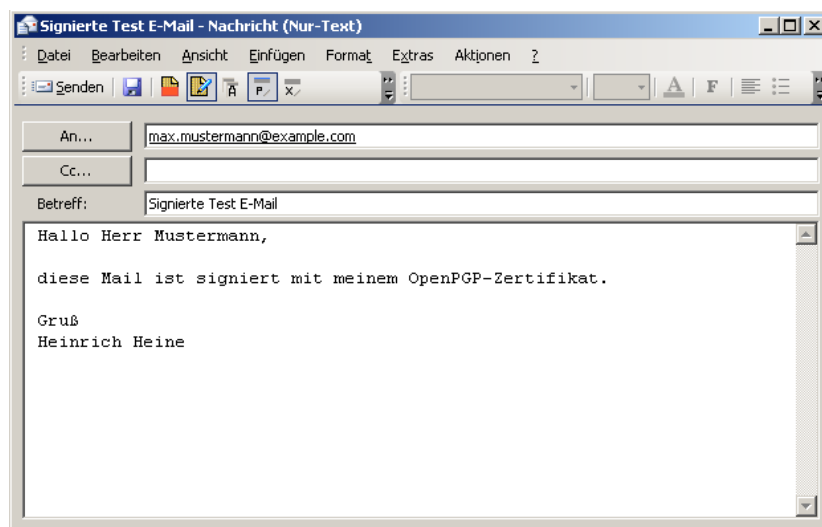
Nutzen Sie dazu das Menü *Extras*→*GnuPG Protokoll* im Outlook-Nachrichtenfenster und wählen Sie *PGP/MIME*, *S/MIME* oder *automatisch*. – Die Erklärungen und Hinweise vom Verschlüsseln (siehe Seite 58) gelten auch für das Signieren.

Signierung aktivieren

Bevor Sie Ihre Nachricht abschicken, geben Sie noch an, dass Ihre Nachricht signiert versendet werden soll:

Dazu aktivieren Sie den Menüeintrag *Extras*→*Nachricht mit GnuPG signieren*. Die Schaltfläche mit dem unterschreibenden Stift wird aktiviert.

Ihr E-Mail-Fenster sollte anschließend etwa so aussehen (als Protokoll wurde hier exemplarisch OpenPGP gewählt):

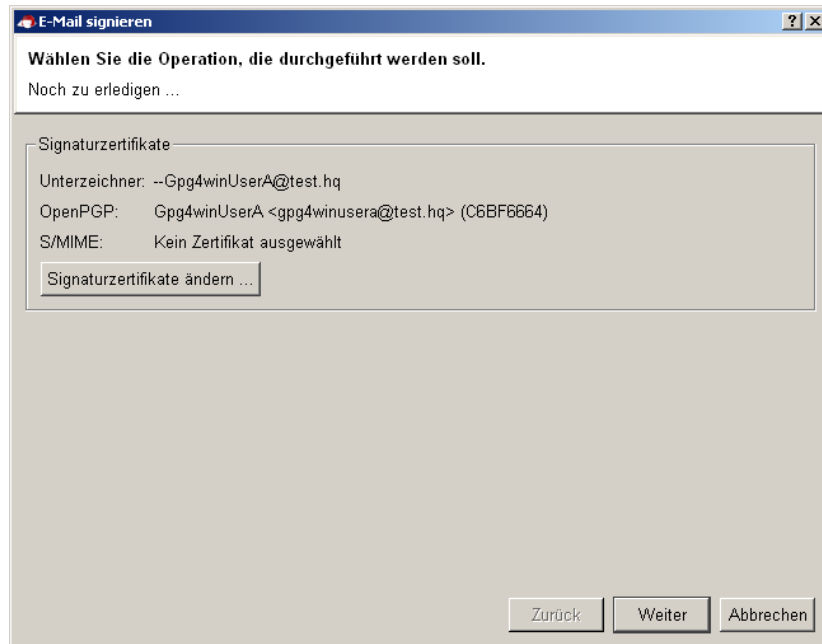


Wie beim Verschlüsseln können Sie natürlich auch die Signieroption jederzeit mit einem erneuten Klick auf die Schaltfläche wieder deaktivieren.

Klicken Sie nun auf [Senden].

Zertifikatsauswahl

Daraufhin öffnet Kleopatra ein Fenster, in dem – anders als beim Verschlüsseln – Ihre *eigenen* Zertifikate angezeigt werden.



Denn:

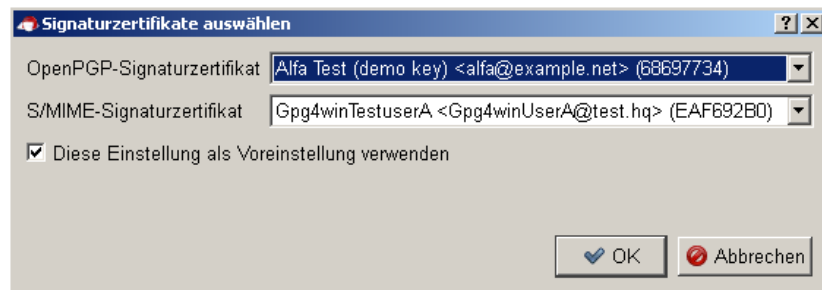
Signieren können Sie nur mit Ihrem eigenen geheimen Schlüssel.

Logisch, denn nur Ihr eigener geheimer Schlüssel bestätigt Ihre Identität. Der Korrespondenzpartner kann dann mit Ihrem Zertifikat, das er bereits hat oder sich besorgen kann, Ihre Identität überprüfen. Denn nur Ihr geheimer Schlüssel passt zu Ihrem Zertifikat.

Im Normalfall können Sie im obigen Dialog Ihr vorausgewähltes Zertifikat mit [Weiter] bestätigen. Beim ersten Durchlauf des Signierprozesses müssen Sie jedoch zunächst Kleopatra Ihr bevorzugtes Zertifikat für OpenPGP bzw. S/MIME mitteilen.

Sollte also noch kein Zertifikat ausgewählt oder nicht Ihr richtiges Zertifikat angezeigt sein – z.B. weil Sie mehrere Zertifikate besitzen – klicken Sie auf [Signaturzertifikate ändern ...].

Sie bekommen einen Auswahl-Dialog mit einer Auflistung aller Ihrer eigenen Zertifikate, die in Ihrer Zertifikatsverwaltung existieren. Nachfolgend der Dialog, gefüllt mit Beispielzertifikaten für OpenPGP und S/MIME:



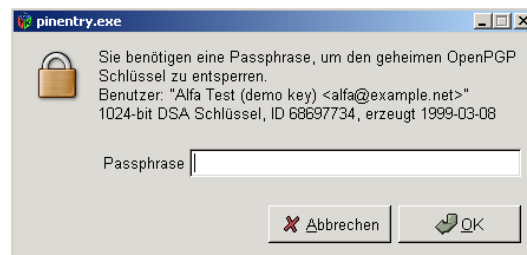
Wählen Sie Ihr korrektes Zertifikat aus, mit dem Sie Ihre Nachricht signieren wollen.

Klicken Sie anschließend auf [OK]. Ihr ausgewähltes Zertifikat wird in den letzten „E-Mail signieren“-Dialog übernommen.

Bestätigen Sie Ihr Zertifikat mit [Weiter].

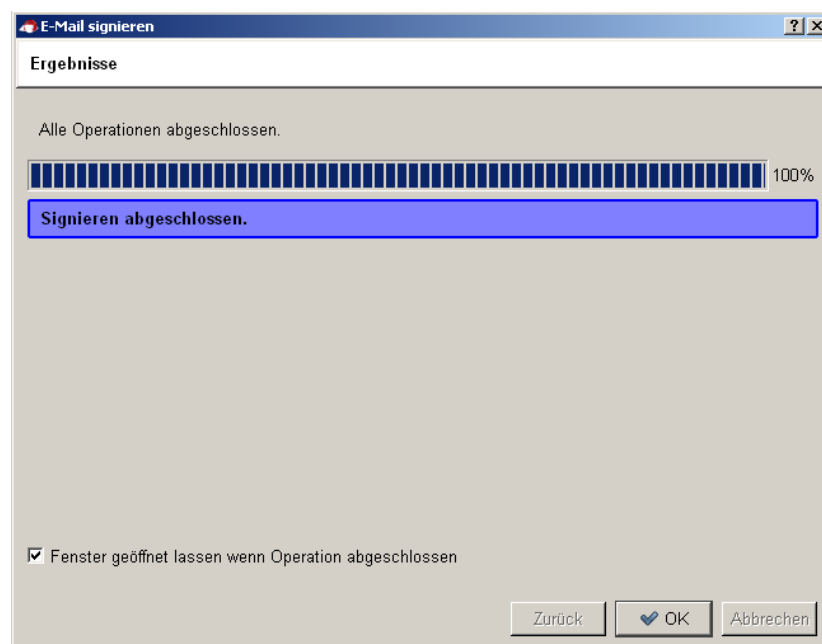
Signierung abschließen

Um die Signierung Ihrer E-Mail abzuschließen, werden Sie aufgefordert im folgenden Fenster Ihre geheime Passphrase einzugeben:



Dies ist notwendig, weil Sie mit Ihrem eignen geheimen Schlüssel signieren. Bestätigen Sie Ihre Eingabe mit [OK].

Ihre Nachricht wird nun signiert und versandt. Nach erfolgreicher Signierung Ihrer Nachricht, erhalten Sie folgenden Dialog:



Herzlichen Glückwunsch! Sie haben Ihre erste E-Mail signiert!

Fassen wir kurz zusammen...

Sie haben gelernt, wie Sie eine E-Mail mit Ihrem eigenen Zertifikat **signieren**.

Seit dem letzten Kapitel wissen Sie nun auch, wie Sie eine E-Mail mit dem Zertifikat Ihres Korrespondenzpartners **verschlüsseln**.

Damit beherrschen Sie nun die beiden wichtigsten Techniken für einen sicheren E-Mail-Versand.

Natürlich können Sie beide Techniken auch kombinieren. Entscheiden Sie ab sofort bei jeder neuen E-Mail, wie Sie Ihre Nachricht versenden wollen – je nachdem, wie wichtig und schutzbedürftig der Inhalt Ihrer E-Mail ist:

- unverschlüsselt
- verschlüsselt
- signiert
- signiert und verschlüsselt (*Mehr dazu im Abschnitt 10.4, S. 73*)

Diese vier Kombinationen können Sie entweder mit OpenPGP oder mit S/MIME realisieren.

10.2. Signatur mit GpgOL überprüfen

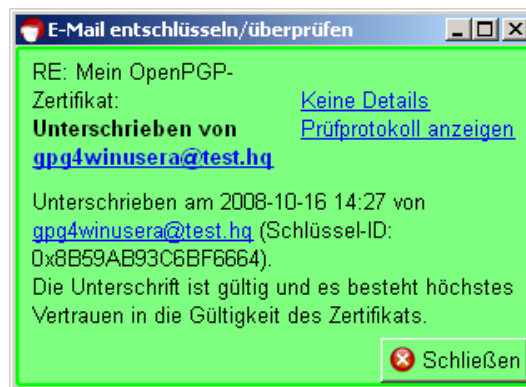
Angenommen Sie erhalten eine signierte E-Mail Ihres Korrespondenzpartners.

Die Überprüfung dieser elektronischen Signatur ist sehr einfach. Alles was Sie dazu brauchen, ist das OpenPGP- oder X.509-Zertifikat Ihres Korrespondenzpartners. Dessen Zertifikat sollten Sie vor der Überprüfung bereits in Ihre Kleopatra Zertifikatsverwaltung importiert haben (vgl. Kapitel 8).

Um eine signierte OpenPGP- oder S/MIME-E-Mail zu überprüfen, gehen Sie genauso vor, wie bei der Entschlüsselung einer E-Mail (vgl. Kapitel 7):

Starten Sie Outlook und öffnen Sie eine signierte E-Mail.

GpgOL übergibt die E-Mail automatisch an Kleopatra zur Prüfung der Signatur. Kleopatra meldet das Ergebnis in einem Statusdialog, z.B.:



Die Signaturprüfung war erfolgreich! Schließen Sie den Dialog, um die signierte E-Mail zu lesen.

Möchten Sie die Überprüfung noch einmal manuell aufrufen, so wählen Sie im Menü der geöffneten E-Mail *Extras*→*GpgOL Entschlüsseln/Prüfen*.

Sollte die Signaturprüfung fehler schlagen, wurde die Nachricht bei der Übertragung verändert. Aufgrund der technischen Gegebenheiten im Internet ist es nicht auszuschließen, dass die E-Mail durch eine fehlerhafte Übertragung verändert wurde. Das ist zunächst der wahrscheinlichste Fall. Es kann jedoch auch bedeuten, dass der Text nachträglich verändert wurde.

Wie Sie in einem solchen Fall mit der gebrochenen Signatur umgehen sollten, erfahren Sie im Abschnitt 10.3.

Übrigens...

Wenn Sie kein Gpg4win installiert haben und eine signierte E-Mail öffnen, lässt sich die Signatur natürlich nicht überprüfen. Sie sehen dann den E-Mail-Text umrahmt von merkwürdigen Zeilen – das ist die Signatur.

Exemplarisch für OpenPGP zeigen wir Ihnen, wie dann so eine OpenPGP-signierte E-Mail in Ihrem E-Mail-Programm aussieht:

Die E-Mail beginnt mit:

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

und endet unter der E-Mail-Nachricht mit:

```
-----BEGIN PGP SIGNATURE-----  
Version: GnuPG v1.4.2 (MingW32)  
  
iEYEARECAAYFAjxeqy0ACgkQcwePex+3Ivs79wCfW8u  
ytRsEXgzCrfPnjGrDDtb7QZIAN17B8l8gFQ3WIUUDCMfA5cQajHcm  
=O6lY  
-----END PGP SIGNATURE-----
```

Dies ist ein Beispiel – Abwandlungen sind natürlich möglich.



10.3. Gründe für eine gebrochene Signatur

Es gibt mehrere Gründe, die zu einem Bruch einer Signatur führen können. Wenn Sie eine E-Mail mit dem Vermerk „Bad signature“ oder „Überprüfung fehlgeschlagen“ erhalten, ist das ein Warnsignal, dass Ihre E-Mail manipuliert sein könnte; d.h. jemand hat vielleicht den Inhalt oder den Betreff der E-Mail verändert.

Eine gebrochene Signatur muss aber nicht zwangsläufig bedeuten, dass Ihre E-Mail manipuliert wurde:

1. Aufgrund der technischen Gegebenheiten ist es nicht auszuschließen, dass die E-Mail durch eine fehlerhafte Übertragung über das Internet verändert wurde.
2. Das E-Mail-Programm des Absenders oder Empfängers kann falsch eingestellt sein. Wenn man eine signierte E-Mail verschickt, sollte man unbedingt darauf achten, dass im E-Mail-Programm alle Optionen ausgeschaltet sind, die E-Mail schon beim Versand verändern. Dazu zählt „HTML-Mails“ und „Word Wrap“.

„Word Wrap“ bezeichnet den Umbruch von Zeilen in der E-Mail. Beides verändert natürlich die E-Mail und „bricht“ die Signatur, obwohl niemand sie willentlich verändert hat. Bei Outlook 2003 beispielsweise muss diese Option unter *Extras*→*Optionen*→*E-Mail-Format* das Nachrichtenformat auf *Nur Text* eingestellt sein.

Häufig sind falsche Einstellungen am E-Mail-Programm der Grund für eine gebrochene Signatur.

In jedem Fall sollten Sie die E-Mail erneut beim Absender anfordern!

10.4. Verschlüsseln und signieren

Normalerweise verschlüsseln Sie eine Nachricht mit dem Zertifikat Ihres Korrespondenzpartners, der dann mit seinem geheimen Schlüssel die E-Mail entschlüsselt.

Die umgekehrte Möglichkeit – man würde mit dem geheimen Schlüssel verschlüsseln –, ist technisch nicht möglich und macht keinen Sinn, weil alle Welt das dazugehörigen (öffentliche) Zertifikat kennt und die Nachricht damit entschlüsseln könnte.

Es gibt aber ein anderes Verfahren, um mit Ihrem geheimen Schlüssel eine Datei zu erzeugen: Die Signatur, wie wir sie oben bereits beschrieben haben. Solch eine digitale Signatur bestätigt eindeutig die Urheberschaft – denn wenn jemand Ihr Zertifikat auf diese Datei (die Signatur) anwendet und die Ausgabe dieser Prüfung ist „gültig“, so kann diese Datei nur von Ihrem privaten Schlüssel kodiert worden sein. Und zu dem dürfen ja nur Sie selbst Zugang haben.

Wenn Sie ganz sicher gehen wollen, können Sie beide Möglichkeiten kombinieren, also die E-Mail verschlüsseln und signieren:

1. Sie **signieren** die Botschaft mit Ihren eigenen geheimen Schlüssel. Damit ist die Urheberschaft nachweisbar.
2. Dann **verschlüsseln** Sie den Text mit dem Zertifikat des Korrespondenzpartners.

Damit hat die Botschaft sozusagen zwei Briefumschläge:

1. Einen Innenumschlag, der mit einem Siegel verschlossen ist (die Signatur mit Ihrem eigenen geheimen Schlüssel).
2. Einen soliden äußeren Umschlag (die Verschlüsselung mit dem Zertifikat des Korrespondenzpartners).

Ihr Brieffartner öffnet die äußere, starke Hülle mit seinem eigenen geheimen Schlüssel. Hiermit ist die Geheimhaltung gewährleistet, denn nur dieser Schlüssel kann den Text dekodieren. Die innere, versiegelte Hülle öffnet er mit Ihrem Zertifikat und hat den Beweis Ihrer Urheberschaft, denn wenn Ihr Zertifikat passt, kann er nur mit Ihrem Geheimschlüssel kodiert worden sein.

Sehr trickreich und – wenn man ein wenig darüber nachdenkt – auch ganz einfach.

11. Wie Sie Ihre E-Mails verschlüsselt archivieren

Eine wichtige Einstellung müssen Sie nun noch vornehmen, um Gpg4win richtig nutzen zu können: Es geht um Ihre E-Mails, die Sie verschlüsselt versenden.

Wie können Sie diese wichtigen Nachrichten sicher archivieren? Natürlich können Sie einfach eine Klartextversion Ihrer Texte aufbewahren, aber das wäre eigentlich nicht angebracht. Wenn Ihre Mitteilung geheimhaltungsbedürftig war, sollte sie auch nicht im Klartext auf Ihrem Rechner gespeichert sein. Sie sollten also stets Ihre verschlüsselt gesendeten E-Mails auch *verschlüsselt* aufbewahren!

Sie ahnen das Problem: zum Entschlüsseln Ihrer archivierten (versendeten) E-Mails braucht Sie aber den geheimen Schlüssel des Empfängers – und den haben Sie nicht und werden Sie nie haben. . .

Also was tun?

Ganz einfach: **Sie verschlüsseln zusätzlich auch an sich selbst!**

Die Nachricht wird einmal für Ihren eigentlichen Korrespondenzpartner (z.B. Adele) verschlüsselt und ein weiteres Mal auch für Sie, mit Hilfe Ihres eigenen Zertifikats. So können Sie die E-Mail später einfach mit Ihrem eigenen geheimen Schlüssel wieder lesbar machen.

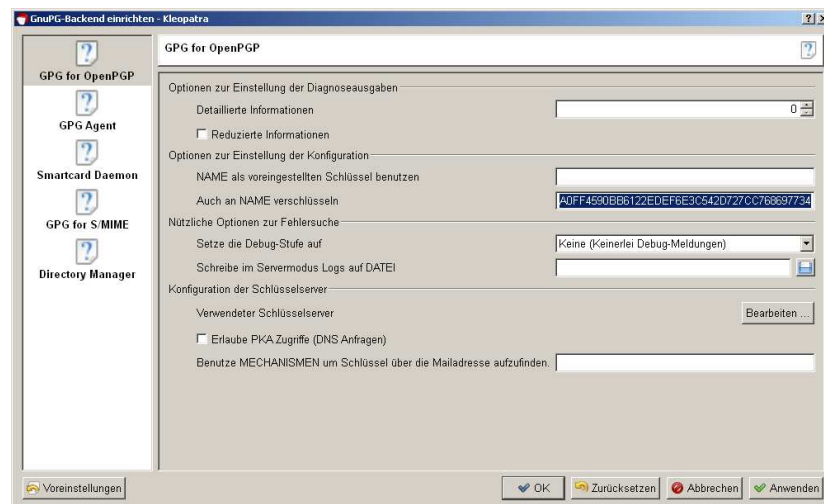
Da Gpg4win nicht wissen kann, welchen Schlüssel Sie benutzen (Sie können ja auch mehrere haben) müssen Sie dem Programm dies mitteilen.

Wie? – Das erfahren Sie auf der nächsten Seite.

Um diese Option zu nutzen, genügen einige wenige Mausklicks:

Öffnen Sie Kleopatra und dort das GnuPG-Backend über *Extras*→*GnuPG-Backend einrichten...*

Wählen Sie *GPG for OpenPGP* bzw. *GPG for S/MIME* und geben Sie in das Textfeld an der Stelle ***Auch an NAME verschlüsseln*** den vollständigen Fingerabdruck Ihres Zertifikats an, z.B.:



Den Fingerabdruck finden Sie in den Zertifikatsdetails.

Eine entsprechende Option finden Sie auch bei allen E-Mail-Programmen, die GnuPG direkt unterstützen.

Fassen wir kurz zusammen...

1. Sie haben mit dem Zertifikat Ihres Korrespondenzpartners eine E-Mail verschlüsselt und ihm damit geantwortet.
2. Kleopatra verschlüsselt Ihre gesendeten verschlüsselten E-Mails auch zusätzlich mit Ihrem eigenen Zertifikat, so dass die Nachrichten für Sie lesbar bleiben.

Das war's! Zum Ende dieses ersten Teils des Kompendiums werden Sie nun ein sehr fundiertes Einsteigerwissen über Gpg4win besitzen.

Willkommen in der Welt der freien und sicheren E-Mail-Verschlüsselung!

Für ein noch besseres Verständnis, wie Gpg4win im Hintergrund wirklich funktioniert, empfehlen wir Ihnen sich nun mit dem zweiten, fortgeschrittenen Teil von Gpg4win zu beschäftigen. Wir versprechen Ihnen, Sie werden viele spannende Dinge darin entdecken!

Genau wie das Kryptographiesystem Gpg4win wurden dieses Kompendium nicht nur für Mathematiker, Geheimdienstler und Kryptographen geschrieben, sondern **für jedermann**.

Teil II.

Fortgeschrittene

12. Wie funktioniert Gpg4win?

Das Besondere an Gpg4win und der zugrundeliegenden Public-Key Methode ist, dass sie jeder verstehen kann und soll. Nichts daran ist Geheimwissen – es ist nicht einmal besonders schwer zu verstehen.

Die Benutzung von Gpg4win ist sehr einfach, seine Wirkungsweise dagegen ziemlich kompliziert. Wir werden in diesem Kapitel erklären, wie Gpg4win funktioniert – nicht in allen Details, aber so, dass die Prinzipien dahinter deutlicher werden. Wenn Sie diese Prinzipien kennen, werden Sie ein hohes Vertrauen in die Sicherheit von Gpg4win gewinnen.

Ganz am Ende dieses Buches, in Kapitel 24, können Sie – wenn Sie wollen – auch noch die letzten Geheimnisse um die Public-Key Kryptographie lüften und entdecken, warum Gpg4win nicht zu knacken ist.

Der Herr der Schlüsselringe

Wenn man etwas sehr Wertvolles sichern will, schließt man es am besten ein – mit einem Schlüssel. Noch besser mit einem Schlüssel, den es nur einmal gibt und den man ganz sicher aufbewahrt.



Denn wenn dieser Schlüssel in die falschen Hände fällt, ist es um die Sicherheit des wertvollen Gutes geschehen. Dessen Sicherheit steht und fällt mit der Sicherheit des Schlüssels. Also hat man den Schlüssel mindestens genauso gut abzusichern, wie das zu sichernde Gut selbst. Die genaue Form des Schlüssels muss völlig geheim gehalten werden.

Geheime Schlüssel sind in der Kryptographie ein alter Hut: schon immer hat man Botschaften geheimzuhalten versucht, indem man den Schlüssel geheimhielt. Dies wirklich sicher zu machen ist sehr umständlich und dazu auch sehr fehleranfällig.



Das Grundproblem bei der „normalen“ geheimen Nachrichtenübermittlung ist, dass für Ver- und Entschlüsselung derselbe Schlüssel benutzt wird und dass sowohl der Absender als auch der Empfänger diesen geheimen Schlüssel kennen.

Dies führt zu einer ziemlich paradoxen Situation: Bevor man mit einem solchen System ein Geheimnis (eine verschlüsselte Nachricht) mitteilen kann, muss man schon vorher ein anderes Geheimnis (den Schlüssel) mitgeteilt haben. Und da liegt der Hase im Pfeffer: man muss sich ständig mit dem Problem herumärgern, dass der Schlüssel unbedingt ausgetauscht werden muss, aber auf keinen Fall von einem Dritten abgefangen werden darf.

Gpg4win dagegen arbeitet – außer mit dem Geheimschlüssel – mit einem weiteren Schlüssel (engl. „key“), der vollkommen frei und öffentlich (engl. „public“) zugänglich ist.

Man spricht daher auch von Gpg4win als einem „Public-Key“ Verschlüsselungssystem.

Das klingt widersinnig, ist es aber nicht. Der Witz an der Sache: es muss kein Geheimschlüssel mehr ausgetauscht werden. Im Gegenteil: der Geheimschlüssel darf auf keinen Fall ausgetauscht werden! Weitergegeben wird nur der öffentliche Schlüssel (das Zertifikat) – und den kennt sowieso jeder.

Mit Gpg4win benutzen Sie also ein Schlüsselpaar – eine geheime und eine zweite öffentliche Schlüsselhälfte. Beide Hälften sind durch eine komplexe mathematische Formel untrennbar miteinander verbunden. Nach heutiger wissenschaftlicher und technischer Kenntnis ist es unmöglich, einen Schlüsselteil aus dem anderen zu berechnen und damit den Code zu knacken. In Kapitel 24 erklären wir, warum das so ist.



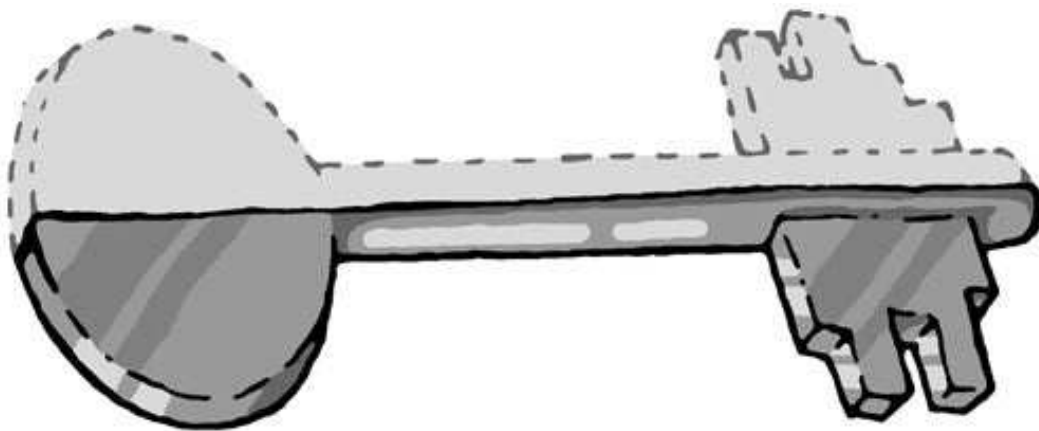
Das Gpg4win-Prinzip ist wie gesagt recht einfach:

Der **geheime Schlüssel**, auch **private Schlüssel** genannt (engl. „secret key“ oder „private key“), muss geheim gehalten werden.

Der **öffentliche Schlüssel**, auch **Zertifikat** genannt (public key), soll so öffentlich wie möglich gemacht werden.

Beide Schlüsselteile haben ganz und gar unterschiedliche Aufgaben:

Der geheime Schlüsselteil **entschlüsselt** Nachrichten.



Der öffentliche Schlüsselteil **verschlüsselt** Nachrichten.

Der öffentliche Safe

In einem kleinen Gedankenspiel wird die Methode des Public-Key Verschlüsselungssystems und ihr Unterschied zur „nicht-public-key“ Methode deutlicher...

Die „nicht-Public-Key Methode“ geht so:

Stellen Sie sich vor, Sie stellen einen Briefkasten vor Ihrem Haus auf, über den Sie geheime Nachrichten übermitteln wollen.

Der Briefkasten ist mit einem Schloss verschlossen, zu dem es nur einen einzigen Schlüssel gibt. Niemand kann ohne diesen Schlüssel etwas hineinlegen oder herausnehmen. Damit sind Ihre geheimen Nachrichten zunächst einmal gut gesichert.



Da es nur einen Schlüssel gibt, muss Ihr Korrespondenzpartner denselben Schlüssel wie Sie haben, um den Briefkasten damit auf- und zuschließen und eine Geheimnachricht deponieren zu können.

Diesen Schlüssel müssen Sie Ihrem Korrespondenzpartner auf geheimem Wege übergeben.



Erst wenn der andere den Geheimschlüssel hat, kann er den Briefkasten öffnen und die geheime Nachricht lesen.

Alles dreht sich also um diesen Schlüssel: wenn ein Dritter ihn kennt, ist es sofort aus mit den Geheimbotschaften. Sie und Ihr Korrespondenzpartner müssen ihn also genauso geheim austauschen wie die Botschaft selbst.

Aber – eigentlich könnten Sie ihm bei dieser Gelegenheit ja auch gleich die geheime Mitteilung übergeben. . .

Übertragen auf die E-Mail-Verschlüsselung: Weltweit müssten alle E-Mail-Teilnehmer geheime Schlüssel besitzen und auf geheimem Wege austauschen, bevor sie geheime Nachrichten per E-Mail versenden könnten.

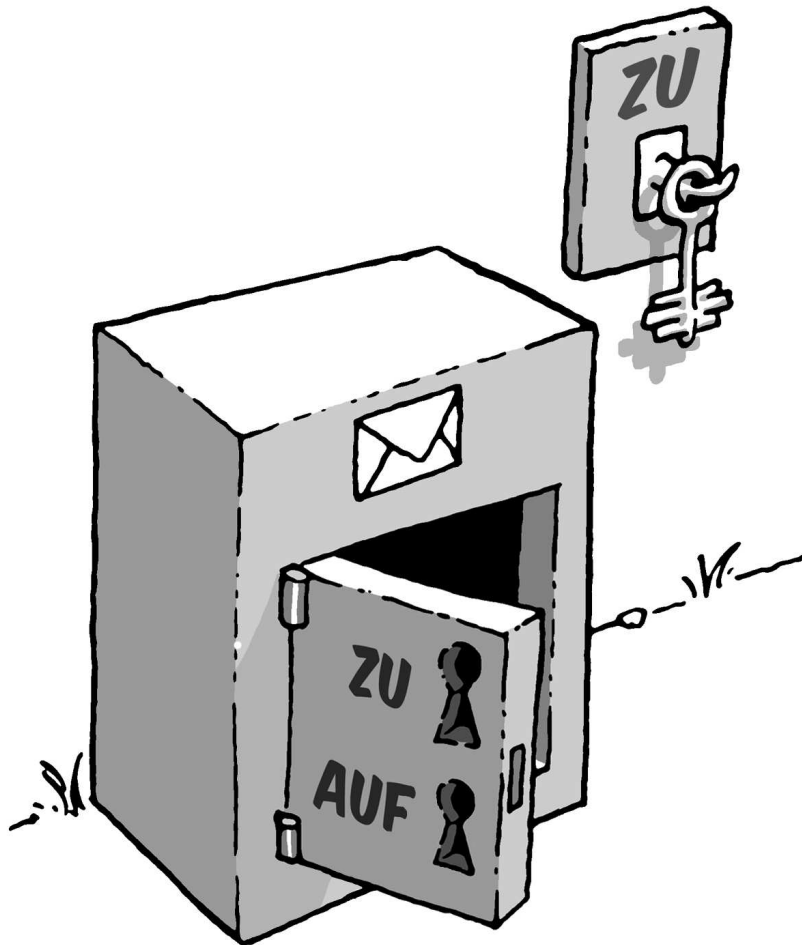
Vergessen wir diese Möglichkeit am besten sofort wieder. . .



Jetzt die Public-Key Methode:

Sie installieren wieder einen Briefkasten vor Ihrem Haus. Aber: dieser Briefkasten ist – ganz im Gegensatz zu dem ersten Beispiel – stets offen. Direkt daneben hängt – weithin öffentlich sichtbar – ein Schlüssel, mit dem jedermann den Briefkasten zuschließen kann.

Zuschließen, aber nicht aufschließen: das ist der Trick!



Dieser Schlüssel gehört Ihnen, und – Sie ahnen es: es ist Ihr öffentlicher Schlüssel.

Wenn jemand Ihnen eine geheime Nachricht hinterlassen will, legt er sie in den Briefkasten und schließt mit Ihrem öffentlichen Schlüssel ab. Jedermann kann das tun, denn der Schlüssel dazu ist ja völlig frei zugänglich.

Kein anderer kann den Briefkasten nun öffnen und die Nachricht lesen. Selbst derjenige, der die Nachricht in dem Briefkasten eingeschlossen hat, kann ihn nicht wieder aufschließen, zum Beispiel um die Botschaft nachträglich zu verändern.

Denn die öffentliche Schlüsselhälfte taugt ja nur zum Abschließen.

Aufschließen kann man den Briefkasten nur mit einem einzigen Schlüssel: Ihrem eigenen geheimen oder privaten Schlüsselteil.

Wieder übertragen auf die E-Mail-Verschlüsselung: Jedermann kann eine E-Mail an Sie verschlüsseln. Er benötigt dazu keineswegs einen geheimen, sondern ganz im Gegenteil einen vollkommen öffentlichen, „ungeheimen“ Schlüssel. Nur ein einziger Schlüssel entschlüsselt die E-Mail wieder: Ihr privater, geheimer Schlüssel.

Spielen wir das Gedankenspiel noch einmal anders herum:

Wenn Sie einem anderen eine geheime Nachricht zukommen lassen wollen, benutzen Sie dessen Briefkasten mit seinem öffentlichen, frei verfügbaren Schlüssel.

Sie müssen Ihren Briefpartner dazu nicht persönlich kennen, ihn getroffen oder je mit ihm gesprochen haben, denn sein öffentlicher Schlüssel ist überall und jederzeit zugänglich. Wenn Sie Ihre Nachricht hinterlegt und den Briefkasten des Empfängers mit seinem öffentlichem Schlüssel wieder verschlossen haben, ist sie völlig unzugänglich für jeden anderen, auch für Sie selbst. Nur der Empfänger kann den Briefkasten mit seinem privaten Schlüssel öffnen und die Nachricht lesen.



Was ist nun eigentlich gewonnen: Es gibt immer noch einen geheimen Schlüssel!?

Der Unterschied gegenüber der „nicht-Public-Key Methode“ ist allerdings ein gewaltiger:

Ihren privater Schlüssel kennen und benutzen nur Sie selbst. Er wird niemals einem Dritten mitgeteilt – die Notwendigkeit einer geheimen Übergabe entfällt, sie verbietet sich sogar.

Es muss überhaupt nichts Geheimes mehr zwischen Absender und Empfänger ausgetauscht werden – weder eine geheime Vereinbarung noch ein geheimes Codewort.

Das ist – im wahrsten Sinne des Wortes – der Knackpunkt: alle „alten“ Verschlüsselungsverfahren können geknackt werden, weil ein Dritter sich beim Schlüsselaustausch in den Besitz des Schlüssels bringen kann.

Dieses Risiko entfällt, weil der Geheimschlüssel nicht ausgetauscht wird und sich nur an einem einzigen Ort befindet: Ihrem eigenen Schlüsselbund.

13. Die Passphrase

Wie Sie im letzten Kapitel gelesen haben, ist der private Schlüssel eine der wichtigsten Komponenten im Public-Key Verschlüsselungssystem. Man muss (und darf) ihn zwar nicht mehr auf geheimem Wege mit seinen Korrespondenzpartnern austauschen, aber nach wie vor ist seine Sicherheit der Schlüssel zur Sicherheit des „ganzen“ Systems.

Es ist deswegen eminent wichtig, diesen privaten Schlüssel sicher abzuspeichern. Dies geschieht auf zweierlei Weise:



Jeder andere Benutzer des Rechners, auf dessen Festplatte dieser Schlüssel gespeichert ist, darf keinen Zugriff auf ihn erhalten – weder zum Schreiben noch zum Lesen. Es ist deswegen unbedingt zu vermeiden, den Schlüssel in einem öffentlichen Ordner (z.B. `c:\Temp` oder `c:\WINNT`) abzuliegen. Gpg4win speichert den Schlüssel deswegen im sogenannten „Heimverzeichnis“ („Homedir“) von GnuPG ab. Dies kann sich je nach System an unterschiedlichen Orten befinden; für einen Benutzer mit dem Anmeldenamen „Harry“ könnte es z.B.:

```
C:\Dokumente und Einstellungen\harry\Anwendungsdaten\gnupg  
sein. Der geheime Schlüssel befindet sich dort in einer Datei mit dem Namen secring.gpg.
```

Dieser Schutz allein ist allerdings nicht ausreichend: Zum einen kann der Administrator des Rechners immer auf alle Dateien zugreifen – also auch auf Ihren geheimen Schlüssel. Zum anderen könnte der Rechner abhanden kommen oder durch „Malware“ (Viren-, Würmer-, Trojanersoftware) kompromittiert werden.

Ein weiterer Schutz ist deswegen notwendig. Dieser besteht aus einer Passphrase.

Die Passphrase sollte aus einem Satz und nicht nur aus einem Wort bestehen. Sie müssen diese Passphrase wirklich „im Kopf“ haben und niemals aufschreiben müssen.

Trotzdem darf sie nicht erraten werden können. Das klingt vielleicht widersprüchlich, ist es aber nicht. Es gibt einige erprobte Tricks, mit deren Hilfe Sie sich eine völlig individuelle, leicht zu merkende und nur sehr schwer zu erratende Passphrase ausdenken können.

Eine **gute Passphrase** kann so entstehen:

Denken Sie an einen Ihnen gut bekannten Satz, z.B.:

Ein blindes Huhn findet auch einmal ein Korn.

Aus diesem Satz nehmen Sie beispielsweise jeden dritten Buchstaben:

nieufdahnlnr (Ein blindes Huhn findet auch einmal ein Korn.)

Diesen Buchstabensalat können Sie sich zunächst sicher nicht gut merken, aber Sie werden ihn eigentlich nie vergessen, solange Sie den ursprünglichen Satz im Kopf haben. Im Laufe der Zeit und je öfter Sie ihn benutzen, prägt sich so eine Passphrase in Ihr Gedächtnis. Erraten kann diese Passphrase niemand.

Denken Sie an ein Ereignis, das sich bereits fest in Ihrem persönlichen Langzeitgedächtnis verankert hat. Vielleicht gibt es einen Satz, mit dem sich Ihr Kind oder Ihr Partner „unvergesslich“ gemacht hat. Oder eine Ferienerinnerung, oder der Titel eines für Sie wichtigen Liedes.

Verwenden Sie kleine und große Buchstaben, Nummern, Sonder- und Leerzeichen durcheinander. Im Prinzip ist alles erlaubt, auch „Ö“, „ß“, „\$“ usw.

Aber Vorsicht – falls Sie Ihren geheimen Schlüssel im Ausland an einem fremden Rechner benutzen wollen, bedenken Sie, dass fremdsprachige Tastaturen diese Sonderzeichen oft nicht haben. Beispielsweise werden Sie kein „ä“ auf einer englischen Tastatur finden.

Machen Sie Rechtschreibfehler, z.B. „feLer“ statt „Fehler“. Natürlich müssen Sie sich diese „feLer“ gut merken können. Oder wechseln Sie mittendrin die Sprache. Aus dem schönen Satz:

In München steht ein Hofbräuhaus.

könnten man beispielsweise diese Passphrase machen:

inMinschen stet 1h0f breuhome

Denken Sie sich einen Satz aus, der möglichst unsinnig ist, den Sie sich aber doch merken können, wie z.B.:

Es blaut so garstig beim Walfang, neben Taschengeld, auch im Winter.

Eine Passphrase in dieser Länge ist ein sicherer Schutz für Ihren geheimen Schlüssel.

Sie darf auch kürzer sein, wenn Sie einige Buchstaben groß schreiben, z.B. so:

Es blAut nEBen TaschengeLd auch im WiNter.

Das ist nun kürzer, aber nicht mehr so leicht zu merken. Wenn Sie eine noch kürzere Passphrase verwenden, indem Sie hier und da Sonderzeichen benutzen, haben Sie zwar bei der Eingabe weniger zu tippen, aber die Wahrscheinlichkeit, dass Sie Ihre Passphrase vergessen, wird dabei noch größer.

Ein extremes Beispiel für eine möglichst kurze, aber dennoch sehr sichere Passphrase ist dieses hier:

R!Qw"s,UIb *7\§

In der Praxis haben sich solche Zeichenfolgen allerdings als recht wenig brauchbar herausgestellt, da man einfach zu wenig Anhaltspunkte für die Erinnerung hat.

Eine **schlechte Passphrase** ist blitzschnell geknackt, wenn sie:

- schon für einen anderen Zweck benutzt wird; z.B. für einen E-Mail-Account oder Ihr Handy
- aus einem Wörterbuch stammt. Cracker lassen in Minutenschnelle komplette Wörterbücher elektronisch über eine Passphrase laufen.
- aus einem Geburtsdatum oder einem Namen besteht. Wer sich die Mühe macht, Ihre E-Mail zu entziffern, kann auch ganz leicht an diese Daten herankommen.
- ein landläufiges Zitat ist; wie z.B. „das wird böse enden“ oder „to be or not to be“. Auch mit derartigen gängigen Zitaten testen Cracker routinemäßig und blitzschnell eine Passphrase.
- aus nur einem Wort oder aus weniger als 8 Zeichen besteht. Denken Sie sich eine längere Passphrase aus.

Wenn Sie nun Ihre Passphrase zusammenstellen, nehmen Sie *auf gar keinen Fall* eines der oben angeführten Beispiele. Denn es liegt auf der Hand: Wenn sich jemand ernsthaft darum bemüht Ihre Passphrase herauszubekommen, würde er zuerst ausprobieren, ob Sie nicht eines dieser Beispiele genommen haben.

Seien Sie kreativ! Denken Sie sich jetzt eine Passphrase aus! Unvergesslich und unknackbar.

Lesen Sie dann im Kapitel 5 weiter, um Ihre neue Passphrase bei der Erzeugung Ihres Schlüsselpaars festzulegen.

14. Zertifikat im Detail

Auf Seite 41 haben Sie sich schon den Detaildialog Ihres erzeugtes Zertifikats angesehen. Viele Angaben zu Ihrem Zertifikat sind dort aufgelistet. Einige werden wir im folgenden Abschnitt ansprechen (beachten Sie die Unterschiede für OpenPGP- und X.509-Zertifikate):

- die Benutzer-ID
- den Fingerabdruck
- die Gültigkeit
- das Inhabervertrauen (*nur OpenPGP*)
- die Beglaubigungen (*nur OpenPGP*)

Die Benutzer-ID besteht aus dem Namen und der E-Mail-Adresse, die Sie während der Zertifikatserzeugung eingegeben haben, also z.B.: Heinrich Heine <heinrichh@gpg4win.de>
Für OpenPGP-Zertifikate können Sie mit Kleopatra über den Menüpunkt *Zertifikate*→*Benutzer-ID hinzufügen...* Ihr Zertifikat um weitere Benutzerkennungen erweitern. Das ist dann sinnvoll, wenn Sie z.B. für eine weitere E-Mail-Adressen Ihr gleiches Zertifikat nutzen möchten.
Beachten Sie: Hinzufügen neuer Benutzer-IDs ist in Kleopatra nur für OpenPGP-Zertifikate (nicht aber für X.509) möglich.

Der Fingerabdruck wird verwendet, um mehrere Zertifikate voneinander zu unterscheiden. Mit dieser Kennung können Sie nach (öffentlichen) Zertifikaten suchen, die z.B. auf einem weltweit verfügbaren OpenPGP-Schlüsselservern oder auf einem X.509-LDAP-Verzeichnisdienst-Server liegen. Was Schlüsselservern sind, erfahren Sie im folgenden Kapitel.

Die Gültigkeit ist normalerweise auf „Unbegrenzt“ gesetzt. Für OpenPGP-Zertifikate können Sie die Gültigkeit selbständig mit Kleopatra ändern, indem Sie auf die Schaltfläche „Ablaufdatum ändern“ in den Zertifikatsdetails klicken (oder den Menü *Zertifikate*→*Ablaufdatum ändern* auswählen) und ein neues Datum eintragen. Damit können Sie Zertifikate nur für eine begrenzte Zeit gültig erklären, zum Beispiel, um sie an externe Mitarbeiter auszugeben.

Die Gültigkeit von X.509-Zertifikaten wird bei der Zertifikatsausstellung von der Zertifizierungsstelle (CA) festgelegt und kann nicht vom Nutzer geändert werden.

Das Inhabervertrauen beschreibt das Maß an Zuversicht, das Sie subjektiv in den Besitzer des OpenPGP-Zertifikats setzen, andere OpenPGP-Zertifikate korrekt zu signieren/beglaubigen. Sie können das Vertrauen über die Schaltfläche [Vertraue durch dieses Zertifikat ausgestellte Beglaubigungen] in den Zertifikatsdetails (oder über das Menü *Zertifikate*→*Inhabervertrauen ändern*) einstellen.
Beachten Sie: Das Inhabervertrauen ist nur für OpenPGP-Zertifikate relevant. Für X.509-Zertifikate gibt es diese Vertrauensmethode nicht.

Die Beglaubigungen Ihres OpenPGP-Zertifikats beinhalten die Benutzer-IDs derjenigen Zertifikatsinhaber, die sich von der Echtheit Ihres Zertifikats überzeugt und es dann auch signiert haben. Das Vertrauen in die Gültigkeit Ihres Zertifikats steigt mit der Anzahl an Beglaubigungen, die Sie von anderen Nutzern erhalten.

Beachten Sie: Beglaubigungen sind nur für OpenPGP-Zertifikate relevant. Für X.509-Zertifikate gibt es diese Vertrauensmethode nicht.

Diese Zertifikatsdetails sind für die tagtägliche Benutzung von Gpg4win nicht unbedingt erforderlich. Sie werden relevant, wenn Sie neue Zertifikate erhalten oder ändern.

Wie Sie fremde Zertifikate prüfen und signieren und was genau das „Netz des Vertrauens“ ist, erfahren Sie im Kapitel 16.

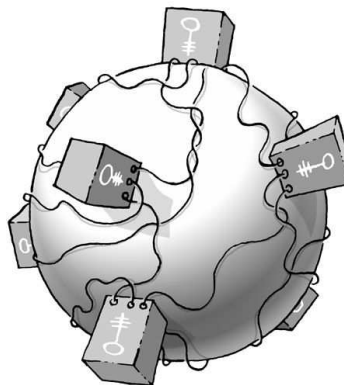
15. Die OpenPGP-Schlüsselserver

Die Nutzung eines Schlüsselservers zum Verbreiten Ihres OpenPGP-Zertifikats haben wir Ihnen bereits im Abschnitt 6.2 einführend erläutert. Dieses Kapitel beschäftigt sich mit den Details von Schlüsselservern.



Schlüsselserver können von allen Programmen benutzt werden, die den OpenPGP-Standard unterstützen.

In Kleopatra ist ein Keyserver bereits voreingestellt: `hkp://keys.gnupg.net`. Ein Mausklick unter *Datei*→*Zertifikate nach Server exportieren...* genügt, und Ihr öffentlicher Schlüssel ist unterwegs rund um die Welt. Es genügt, den Schlüssel an irgendeinen der verfügbaren Keyserver zu senden, denn fast alle synchronisieren sich weltweit miteinander. Es kann ein, zwei Tage dauern, bis Ihr OpenPGP-Zertifikat wirklich überall verfügbar ist, aber dann haben Sie ein globales Zertifikat!



Die Schlüsselserver sind dezentral organisiert, aktuelle Statistiken über ihre Zahl oder die Anzahl der dort liegenden Schlüssel gibt es nicht. Dieses verteilte Netz von Keyservern sorgt für eine bessere Verfügbarkeit und verhindert dass einzelne Systemadministratoren Schlüssel löschen um so die Kommunikation unmöglich zu machen („Denial of Service“-Angriff).

Wir raten dazu, nur moderne Keyserver zu verwendet (auf denen die SKS Software läuft), da nur diese mit den neueren Merkmalen von OpenPGP umgehen können.

Adressen einiger Schlüsselserver

Hier eine Auswahl von gut funktionierenden Schlüsselservern:

- <hkp://blackhole.pca.dfn.de>
- <hkp://pks.gpg.cz>
- <hkp://pgp.cns.ualberta.ca>
- <hkp://minsky.surfnet.nl>
- <hkp://keyserver.ubuntu.com>
- <hkp://keyserver.pramberger.at>
- <http://gpg-keyserver.de>
- <http://keyserver.pramberger.at>

Sollte Sie Probleme mit einer Firewall haben, so versuchen Sie am besten die Keyserver, deren URL mit `http://` beginnen.

Die Keyserver unter den Adressen

- <hkp://keys.gnupg.net>
- <hkp://subkeys.pgp.net>

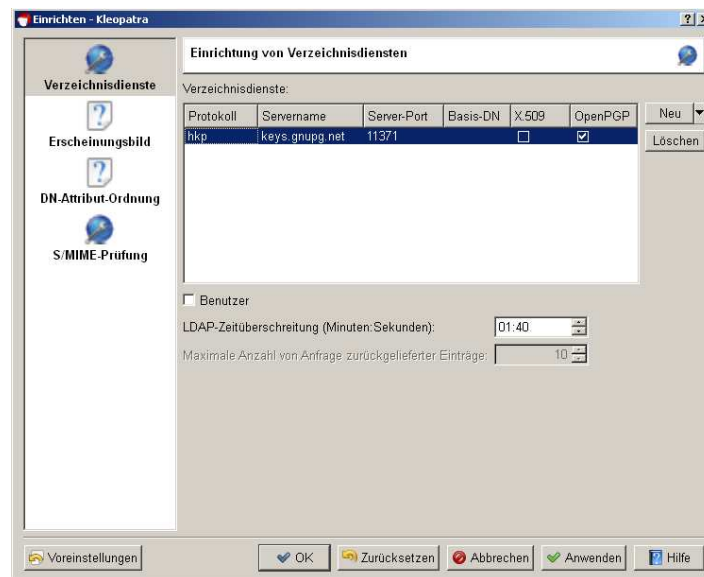
sind ein Sammelpunkt für ein ganzes Netz dieser Server; es wird dann zufällig ein konkreter Server ausgewählt.

Achtung: Der Keyserver `ldap://keyserver.pgp.com` synchronisiert sich nicht mit den anderen Servern und sollte i.d.R. nicht benutzt werden.

Schlüsselservern einrichten

Öffnen Sie die Kleopatra-Einstellungen: *Einstellungen*→*Kleopatra einrichten...*

Legen Sie unter der Rubrik „Verzeichnisdienste“ einen neuen Schlüsselserver an, indem Sie auf *Neu*→*OpenPGP* klicken. Ein voreingestellter OpenPGP-Keyserver mit dem Servernamen `keys.gnupg.net` wird in die Liste hinzugefügt:



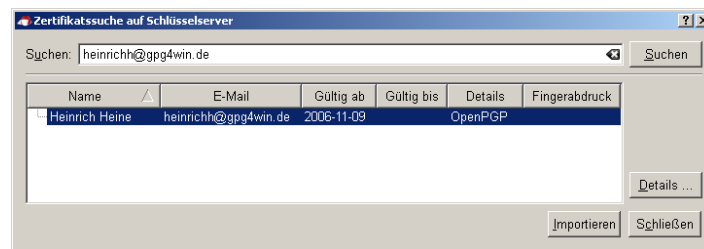
Passen Sie den Eintrag nach Bedarf an (z.B. indem Sie eine Serveradresse von der letzten Seite hier abändern).

Bestätigen Sie abschließend Ihre Konfiguration mit [OK]. Ihr OpenPGP-Schlüsselserver ist nun erfolgreich eingerichtet. Testen Sie nun Ihre Konfiguration indem Sie eine Zertifikatssuche auf dem Server starten (siehe nächster Abschnitt).

Zertifikate auf Schlüsselservern suchen

Genauso einfach wie Sie ein Zertifikat auf Schlüsselserver hochladen (vgl. Abschnitt 6.2), können Sie auch nach Zertifikaten suchen und diese später importieren.

Klicken Sie dazu in Kleopatra auf *Datei*→*Zertifikate auf Server suchen...*. Sie erhalten ein Zertifikatssuchdialog, in dessen Suchfeld Sie den Namen des Zertifikatsbesitzers oder seine E-Mail-Adresse eingeben können:



Um die Details eines ausgewählten Zertifikats zu sehen, klicken Sie auf die Schaltfläche [Details ...].

Zertifikate vom Schlüsselservern importieren

Möchten Sie eines der gefundenen Zertifikate in Ihre lokale Zertifikatssammlung hinzufügen? Dann selektieren Sie das Zertifikat aus der Liste der Suchergebnisse und klicken Sie auf [Importieren].

Kleopatra zeigt Ihnen anschließend einen Dialog mit den Ergebnissen des Importvorgangs an. Bestätigen Sie diesen mit [OK].

War der Import erfolgreich, finden Sie das ausgewählte Zertifikat in der Kleopatra-Zertifikatsverwaltung.

16. Die Zertifikatsprüfung

Woher wissen Sie eigentlich, dass das fremde Zertifikat wirklich vom genannten Absender stammt? Und umgekehrt – warum sollte Ihr Korrespondenzpartner glauben, dass das Zertifikat, das Sie ihm geschickt haben, auch wirklich von Ihnen stammt? Die Absenderangabe auf einer E-Mail besagt eigentlich gar nichts.

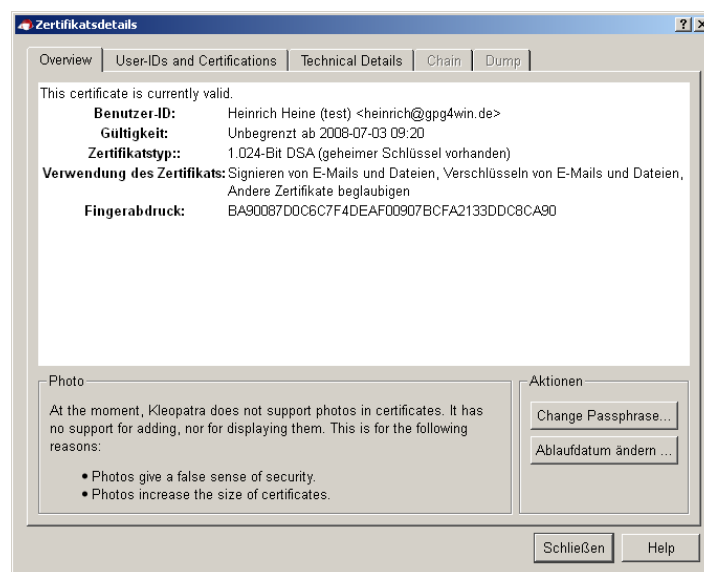
Wenn Ihre Bank z.B. eine E-Mail mit Ihrem Namen und der Anweisung erhält, Ihr sämtliches Guthaben auf ein Nummernkonto auf den Bahamas zu überweisen, wird sie sich hoffentlich weigern – E-Mail-Adresse hin oder her. Eine E-Mail-Adresse besagt überhaupt nichts über die Identität des Absenders.

Der Fingerabdruck

Wenn Sie nur einen kleinen Kreis von Korrespondenzpartnern haben, ist die Sache mit der Identität schnell geregelt: Sie prüfen den Fingerabdruck des anderen Zertifikats.

Jedes Zertifikat trägt eine einmalige Kennzeichnung, die es zweifelsfrei identifiziert; besser noch als ein Fingerabdruck eines Menschen. Deshalb bezeichnet man diese Kennzeichnung eben als „Fingerprint“.

Wenn Sie sich zu einem Zertifikat die Details in Kleopatra anzeigen lassen (z.B. durch Doppelklick auf das Zertifikat), sehen Sie u.a. dessen 40-stelligen Fingerabdruck:



Der Fingerprint von Adeles OpenPGP-Zertifikat ist also:

BA90 087D 0C6C 7F4D EAF0 0907 BCFA 2133 DDC8 CA90

Wie gesagt – der Fingerabdruck identifiziert das Zertifikat und seinen Besitzer eindeutig.

Rufen Sie Ihren Korrespondenzpartner einfach an, und lassen Sie sich von ihm den Fingerprint seines Zertifikats vorlesen. Wenn die Angaben mit dem Ihnen vorliegenden Zertifikat übereinstimmen, haben Sie eindeutig das richtige Zertifikat.

Natürlich können Sie sich auch persönlich mit dem Eigentümer des Zertifikats treffen oder auf jedem anderen Wege mit ihm kommunizieren, solange Sie ganz sicher sind, dass Zertifikat und Eigentümer zusammen gehören. Häufig ist der Fingerprint auch auf Visitenkarten abgedruckt; wenn Sie also eine authentische Visitenkarte haben, so können Sie sich den Anruf ersparen.

OpenPGP-Zertifikat signieren

Nachdem Sie sich „per Fingerabdruck“ von der Echtheit des Zertifikats überzeugt haben, haben Sie nun die Möglichkeit, dieses Zertifikat zu signieren.

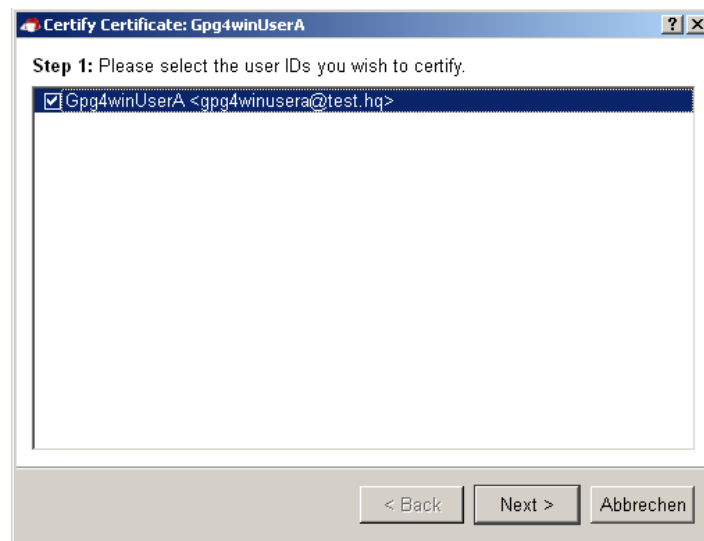
Beachten Sie: Signieren von Zertifikaten durch Benutzer ist nur mit OpenPGP möglich. Bei X.509 ist das autorisierten Stellen vorbehalten!

Durch das Signieren eines (fremden) Zertifikats teilen Sie anderen (Gpg4win-)Benutzern mit, dass Sie dieses Zertifikat für echt halten: Sie übernehmen so etwas wie die „Patenschaft“ über dieses Zertifikat und erhöhen das allgemeine Vertrauen in seiner Echtheit.

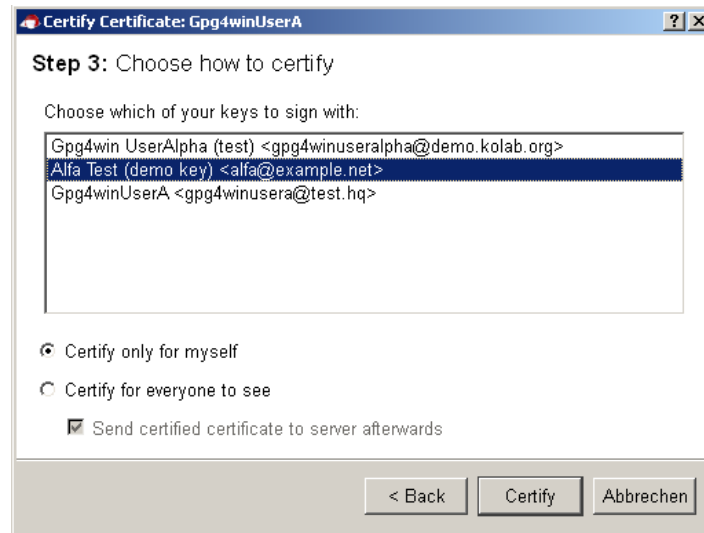
Wie funktioniert das Signieren nun genau?

Selektieren Sie in Kleopatra das OpenPGP-Zertifikat, das Sie für echt halten und signieren möchten. Wählen Sie anschließend im Menü: *Zertifikate*→*Zertifikat beglaubigen...*

Im nachfolgenden Dialog bestätigen Sie nun noch einmal das zu signierende OpenPGP-Zertifikat mit [Weiter]:



Im nächsten Schritt wählen Sie *Ihr* OpenPGP-Zertifikat aus, mit dem Sie das (im letzten Schritt ausgewählte) Zertifikat signieren wollen:

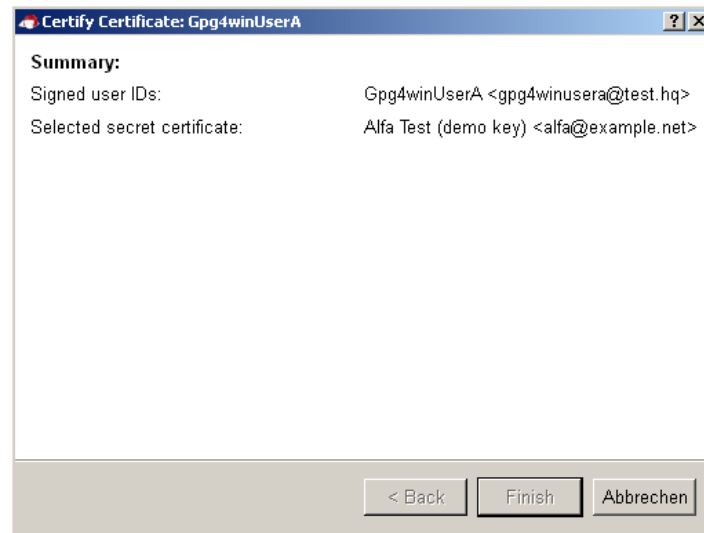


Entscheiden Sie hier, ob Sie [Nur für mich selbst beglaubigen] oder [Für alle sichtbar beglaubigen] wollen. Bei letzterer Variante haben Sie die Option, das signierte Zertifikat anschließend auf einen Keyserver hochzuladen und damit der Welt ein mit Ihrer Beglaubigung versehenes, aktualisiertes Zertifikat zur Verfügung zu stellen.

Bestätigen Sie Ihre Auswahl mit [Beglaubigen].

Wie beim Signieren einer E-Mail müssen Sie auch beim Signieren eines Zertifikats (mit Ihrem privaten Schlüssel) Ihre Passphrase eingeben. Erst nach korrekter Eingabe ist die Beglaubigung abgeschlossen.

Nach erfolgreicher Signierung erhalten Sie folgendes Fenster:



Wollen Sie nun einmal die erfolgte Beglaubigung überprüfen?

Dann öffnen Sie die Zertifikatsdetails des eben signierten Zertifikats. Wählen Sie den Reiter *Benutzer-IDs und Beglaubigungen* und klicken auf die Schaltfläche [Hole Beglaubigungen ein].

Sortiert nach den Benutzer-IDs sehen sie alle Beglaubigungen, die in diesem Zertifikat enthalten sind. Hier sollte Sie auch Ihre Zertifikat wiederfinden, mit dem Sie eben signiert haben.

Das Netz des Vertrauens

Durch das Signieren/Beglaubigen von Zertifikaten entsteht – auch über den Kreis von Gpg4win-Benutzern und Ihrer täglichen Korrespondenz hinaus – ein „Netz des Vertrauens“, bei dem Sie nicht mehr zwangsläufig darauf angewiesen sind, ein OpenPGP-Zertifikat direkt zu prüfen.



Natürlich steigt das Vertrauen in die Gültigkeit eines Zertifikats, wenn mehrere Leute ihn signieren. Ihr eigenes OpenPGP-Zertifikat wird im Laufe der Zeit die Signaturen vieler anderer GnuPG-Benutzer tragen. Damit können immer mehr Menschen darauf vertrauen, dass dieses Zertifikat wirklich Ihnen und niemandem sonst gehört.

Wenn man dieses „Web of Trust“ weiterspinnt, entsteht eine flexible Beglaubigungs-Infrastruktur.

Eine einzige Möglichkeit ist denkbar, mit dem man diese Zertifikatsprüfung aushebeln kann: Jemand schiebt Ihnen ein falsches Zertifikat unter. Also einen öffentlichen OpenPGP-Schlüssel, der vorgibt, von X zu stammen, in Wirklichkeit aber von Y ausgetauscht wurde. Wenn ein solches gefälschtes Zertifikat signiert wird, hat das „Netz des Vertrauens“ natürlich ein Loch. Deshalb ist es so wichtig, sich zu vergewissern, ob ein Zertifikat, wirklich zu der Person gehört, der er zu gehören vorgibt.

Was aber, wenn eine Bank oder Behörde überprüfen möchte, ob die Zertifikate ihrer Kunden echt sind? Alle anzurufen, kann hier sicher nicht die Lösung sein. . .

Zertifizierungsinstanzen

Hier braucht man eine „übergeordnete“ Instanz, der alle Benutzer vertrauen können. Sie überprüfen ja auch nicht persönlich den Personalausweis eines Unbekannten durch einen Anruf beim Einwohnermeldeamt, sondern vertrauen darauf, dass die ausstellende Behörde diese Überprüfung korrekt durchgeführt und beglaubigt hat.

Solche Zertifizierungsinstanzen gibt es auch bei der Public-Key Verschlüsselung für OpenPGP. In Deutschland bietet unter anderem z.B. die Zeitschrift c't schon lange einen solchen Dienst kostenlos an, ebenso wie viele Universitäten.



Wenn man also ein OpenPGP-Zertifikat erhält, dem eine Zertifizierungsstelle per Signatur seine Echtheit bestätigt, kann man sich darauf verlassen.

Derartige Beglaubigungsinstanzen oder „Trust Center“ sind auch bei anderen Verschlüsselungssystemen – wie z.B. S/MIME – vorgesehen. Im Gegensatz zum „Web of Trust“ sind sie hierarchisch strukturiert: Es gibt eine „Oberste Beglaubigungsinstanz“, die weitere „Unterinstanzen“ beglaubigt und ihnen das Recht vergibt, Benutzerzertifikate zu beglaubigen (vgl. Kapitel 3).



Am besten ist diese Infrastruktur mit einem Siegel vergleichbar: die Plakette auf Ihrem Autonommerschild kann Ihnen nur eine dazu berechtigte Institution geben, die die Befugnis dazu wiederum von einer übergeordneten Stelle erhalten hat.

Mit der hierarchischen Zertifizierungs-Infrastruktur entspricht dieses Modell natürlich wesentlich besser den Bedürfnissen staatlicher und behördlicher Instanzen als das lose, auf gegenseitigem Vertrauen beruhende „Web of Trust“ von GnuPG. Der Kern der Beglaubigung selbst ist allerdings völlig identisch: Gpg4win unterstützt neben dem „Web of Trust“ (OpenPGP) zusätzlich auch eine hierarchische Zertifizierungsstruktur (S/MIME). Demnach bietet Gpg4win eine Grundlage um dem strengen Signaturgesetz der Bundesrepublik Deutschland zu entsprechen.

Wenn Sie sich weiter für dieses Thema interessieren, dann können Sie sich an der Quelle informieren: die Website „Sicherheit im Internet“ (www.bsi-fuer-buerger.de) des Bundesministeriums für Wirtschaft und Technologie hält Sie über dieses und viele andere Themen aktuell auf dem Laufenden.

Eine weitere exzellente, mehr technische Informationsquelle zum Thema der Beglaubigungsinfrastrukturen bietet das Original GnuPG Handbuch, das Sie ebenfalls im Internet finden (www.gnupg.org/gph/de/manual).

17. Dateianhänge verschlüsseln

Wenn Sie eine verschlüsselte E-Mail versenden und Dateien anhängen, so wollen Sie in der Regel sicherlich auch, dass die Anhänge verschlüsselt werden.

Bei einer komfortablen Integration von GnuPG in Ihr E-Mail-Programm sollten Anhänge genauso behandelt werden wie der eigentlichen Text Ihrer E-Mail, also signiert, verschlüsselt oder beides zusammen.

GpgOL übernimmt die Verschlüsselung und Signierung von Anhängen automatisch.

Bei weniger komfortablen Integrationen müssen Sie aufpassen: Die Anhänge werden oft unverschlüsselt mitgesendet.

Was kann man in so einem Fall tun? Ganz einfach: Sie verschlüsseln den Anhang getrennt und hängen ihn dann in verschlüsseltem Zustand an die E-Mail an. Es läuft also auf ein ganz normales verschlüsseln von Dateien hinaus. Und das ist in Kapitel 18 beschrieben.

18. Dateien signieren und verschlüsseln

Nicht nur E-Mails, sondern auch einzelne Dateien können Sie mit Gpg4win signieren und verschlüsseln. Das Prinzip ist das gleiche:

- Sie **signieren** eine Datei mit Ihrem Zertifikat, um sicherzugehen, dass die Datei unverändert bei Ihrem Empfänger ankommt.
- Sie **verschlüsseln** eine Datei mit dem Zertifikat des Empfängers, um die Datei vor unbefugten Personen geheim zu halten.

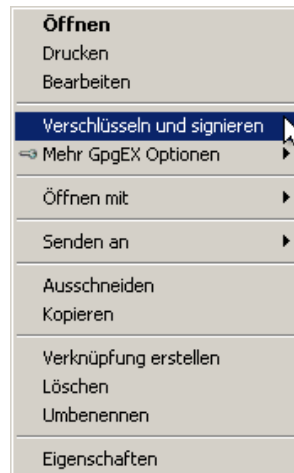
Mit der Anwendung **GpgEX** können Sie Dateien ganz einfach aus Ihrem Windows Explorer heraus signieren oder verschlüsseln – egal ob OpenPGP oder S/MIME. Dieses Kapitel erläutert Ihnen, wie das genau funktioniert.

Sollten Sie eine Datei als E-Mail-Anhang verschicken, übernimmt z.B. GpgOL automatisch die Signierung bzw. Verschlüsselung der Datei zusammen mit Ihrer E-Mail. Sie brauchen sich in diesem Fall nicht gesondert darum kümmern.

18.1. Dateien signieren und überprüfen

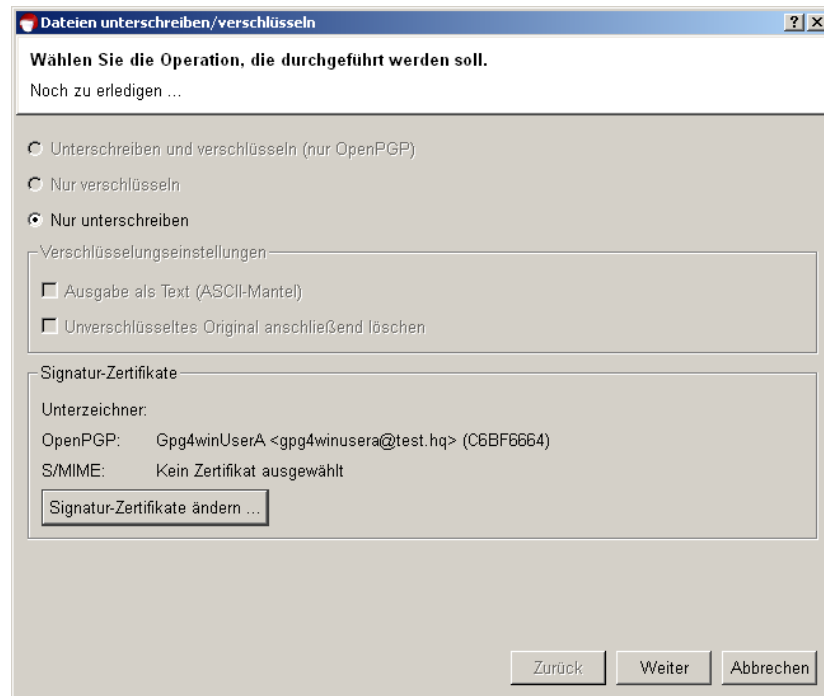
Beim Signieren einer Datei kommt es nicht vorrangig auf die Geheimhaltung, sondern auf die Unverändertheit der Datei an.

Die Signierung können Sie bequem mit **GpgEX** aus dem Kontextmenü des Windows Explorer ausführen. Selektieren Sie eine Datei und öffnen Sie mit der rechten Maustaste das Kontextmenü:



Dort wählen Sie *Verschlüsseln und Signieren* aus.

Selektieren Sie im erscheinenden Fenster die Option *Nur unterschreiben*:

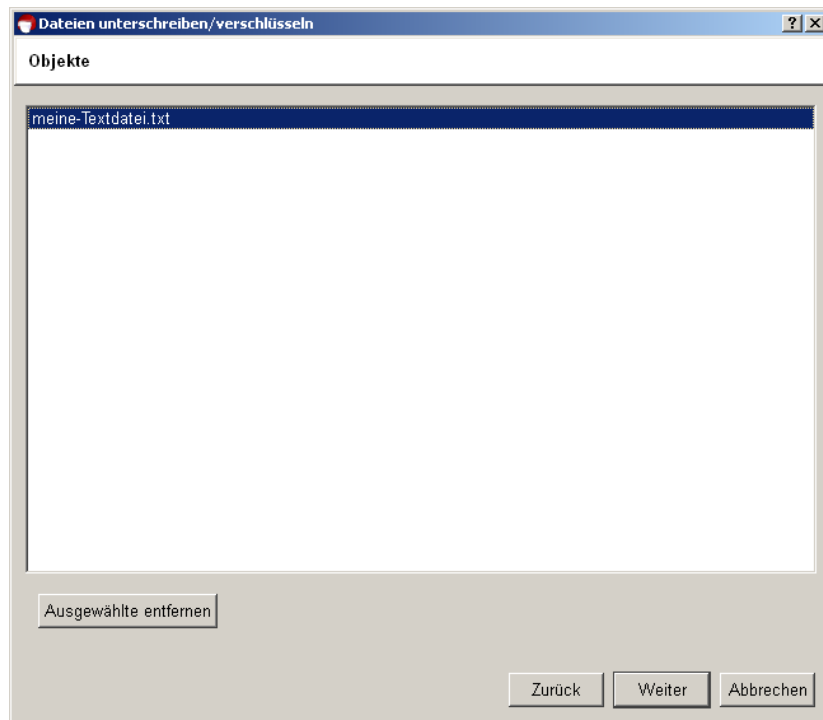


Der darunter liegende Rahmen *Verschlüsselungseinstellungen* wird dadurch ausgegraut, da sie ja lediglich signieren möchten.

Im letzten Abschnitt *Signatur-Zertifikate* wählen Sie – sofern nicht schon vorausgewählt – Ihr (OpenPGP oder S/MIME) Standardzertifikat aus, mit dem Sie die Datei signieren möchten.

Klicken Sie nun auf [Weiter].

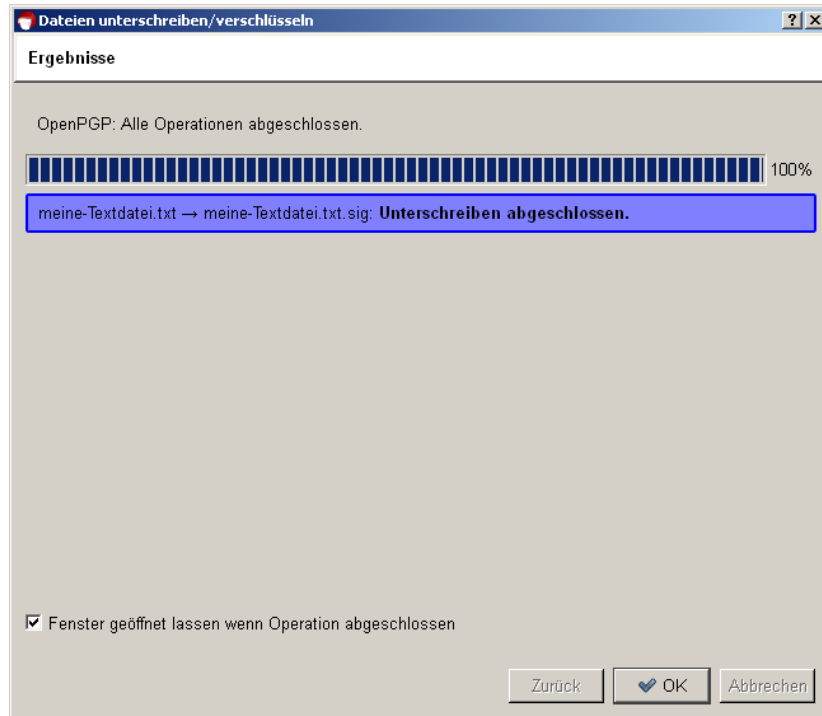
Nun haben Sie die Möglichkeit, die Auswahl der zu bearbeitenden Dateien zu bestätigen oder ggf. die Auswahl zu korrigieren.



Bestätigen Sie Ihre Auswahl mit [Weiter].

Sie müssen nun Ihre Passphrase in den aufkommenden Pinentry-Dialog eingeben.

Nach erfolgreicher Signierung erhalten Sie folgendes Fenster:



Sie haben damit Ihre Datei erfolgreich signiert.

Abhängig davon, ob Sie OpenPGP oder S/MIME zum Signieren genutzt haben, erhalten Sie als Ergebnis eine Datei mit der Endung `*.sig` (bei OpenPGP) oder `*.p7s` (bei S/MIME).

Beim Signieren einer Datei wird stets eine „abgetrennte“ Signatur verwendet. Dies bedeutet, dass Ihre zu signierende Datei unverändert bleibt und eine zweite Datei mit der eigentlichen Signatur erzeugt wird. Um die Signatur später zu überprüfen, sind beide Dateien notwendig.

Folgendes Beispiel zeigt noch einmal, welche neue Datei Sie erhalten, wenn Sie Ihre ausgewählte Datei (hier `<dateiname>.txt`) mit OpenPGP bzw. S/MIME signieren:

OpenPGP:

`<dateiname>.txt → <dateiname>.txt.sig`

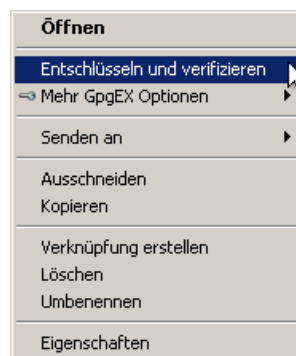
S/MIME:

`<dateiname>.txt → <dateiname>.txt.p7s`

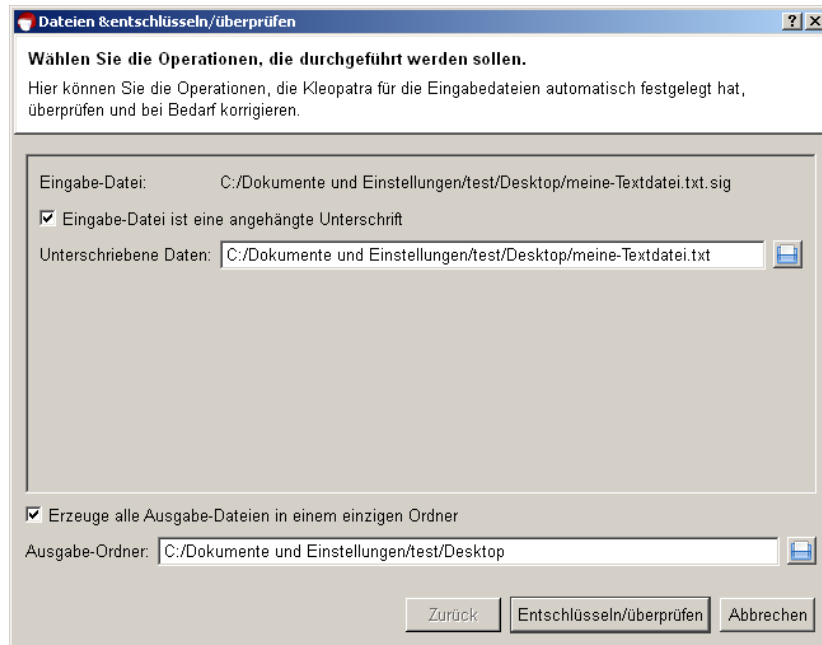
Signatur überprüfen

Wir wollen nun überprüfen, ob die eben signierte Datei korrekt ist.

Zum Überprüfen der Unverändertheit und der Authentizität müssen die Signatur-Datei und die unterschriebene Datei (Originaldatei) im selben Verzeichnis liegen. Selektieren Sie die Signatur-Datei – also die mit der Endung **.sig* oder **.p7s* – und wählen Sie aus dem Kontextmenü des Windows Explorer den Eintrag *Entschlüsseln und Verifizieren*:



Daraufhin erhalten Sie folgendes Fenster:



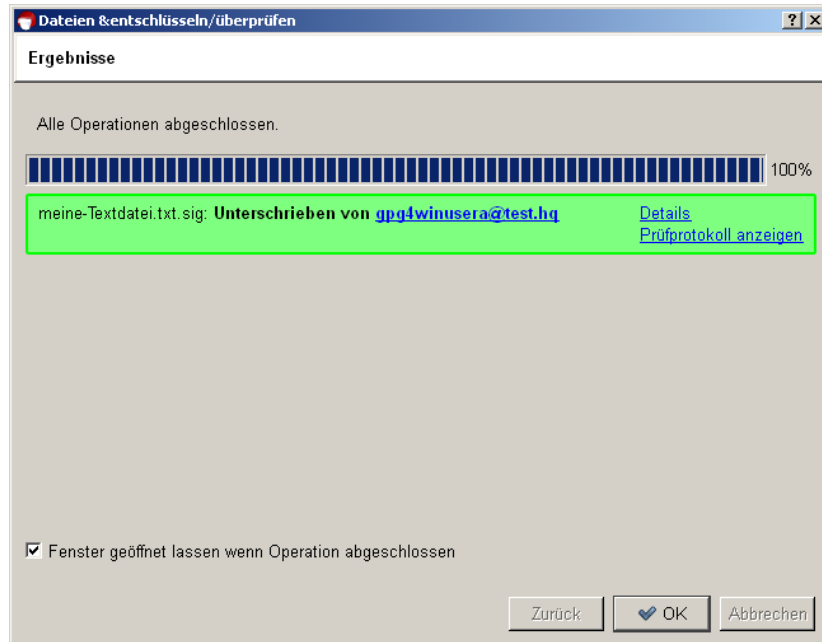
Kleopatra zeigt unter *Eingabe-Datei* den vollständigen Pfad zur ausgewählten Signatur-Datei an.

Die Option *Eingabe-Datei ist eine angehängte Unterschrift* ist aktiviert, da wir ja unsere Originaldatei (hier: *unterschiedene Daten*) mit der Eingabe-Datei signiert haben. Kleopatra findet automatisch die zugehörige unterschriebene Originaldatei.

Automatisch ist auch der *Ausgabe-Ordner* auf den gleichen Pfad ausgewählt. Der wird aber erst relevant wenn Sie mehr als eine Datei gleichzeitig verarbeiten.

Bestätigen Sie die gegebenen Operationen mit [Entschlüsseln/überprüfen].

Nach erfolgreicher Überprüfung der Signatur erhalten Sie folgendes Fenster:

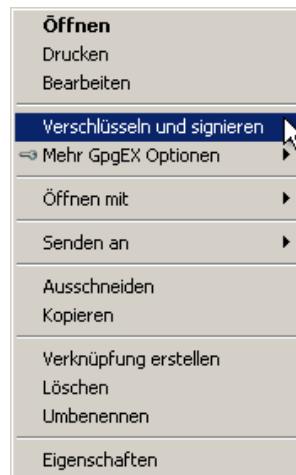


Das Ergebnis zeigt, dass die Signatur gültig ist – also die Datei nicht verändert wurde. Selbst wenn nur ein Zeichen hinzugefügt, gelöscht oder geändert wurde, wird die Signatur als ungültig angezeigt.

18.2. Dateien verschlüsseln und entschlüsseln

Genauso wie E-Mails lassen sich Dateien nicht nur signieren, sondern auch verschlüsseln. Das wollen wir im folgenden Abschnitt mit GpgEX und Kleopatra einmal durchspielen.

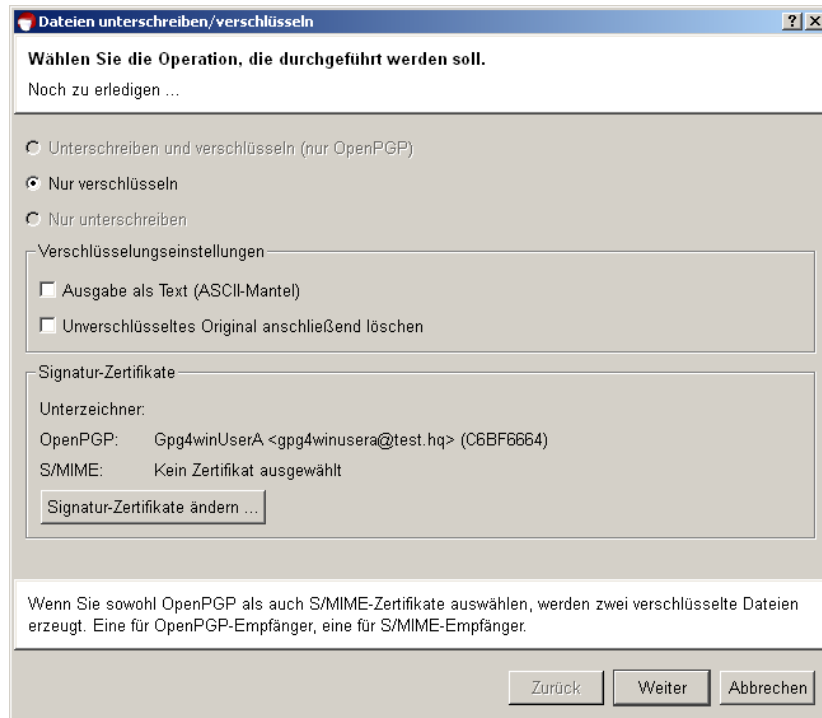
Selektieren Sie eine Datei und öffnen Sie mit der rechten Maustaste das Kontextmenü:



Wählen Sie hier *Verschlüsseln und Signieren* aus.

Sie erhalten den Dialog, den Sie vom Signieren einer Datei (vgl. Abschnitt 18.1) schon kennen.

Entscheiden Sie sich im oberen Feld für die Option *Nur Verschlüsseln*:



Die Verschlüsselungseinstellungen sollten Sie nur bei Bedarf umstellen:

Ausgabe als Text (ASCII-Mantel): Bei Aktivierung dieser Option erhalten Sie die verschlüsselte Datei mit der Dateiendung **.asc* (OpenPGP) bzw. **.pem* (S/MIME). Diese Dateitypen sind mit jedem Texteditor lesbar – Sie würden dort den Buchstaben- und Ziffernsalat sehen, den Sie schon kennen.

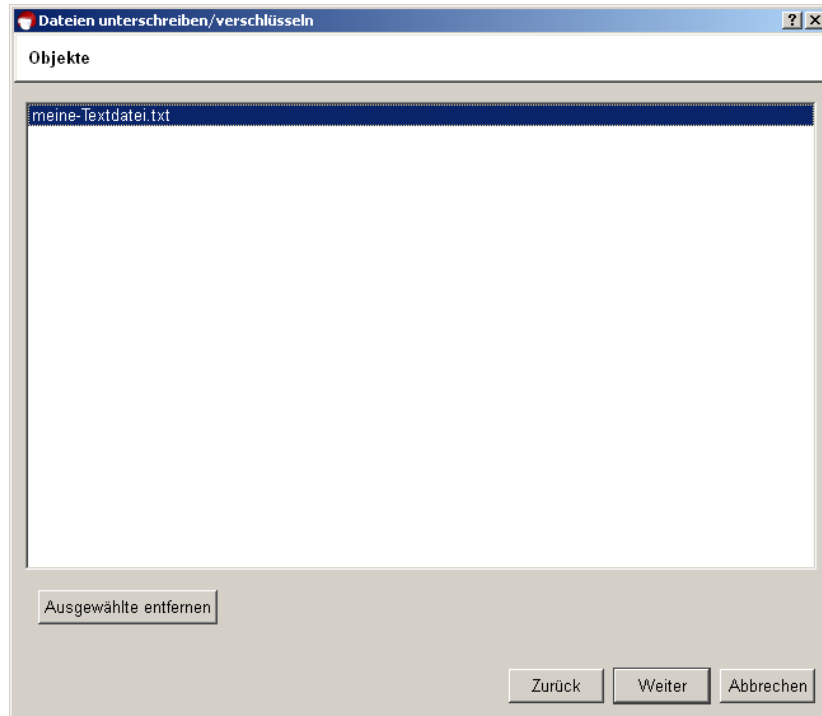
Ist diese Option nicht ausgewählt (Voreinstellung), so wird eine verschlüsselte Datei mit der Endung **.gpg* (OpenPGP) bzw. **.p7m* (S/MIME) angelegt. Diese Dateien sind Binärdateien – eine Betrachtung im Texteditor ist also sinnlos.

Was Sie hier benutzen ist eigentlich gleichgültig; Gpg4win kommt mit beiden Arten klar.

Unverschlüsseltes Original anschließend löschen: Ist diese Option aktiviert, wird Ihre ausgewählte Originaldatei nach dem Verschlüsseln gelöscht.

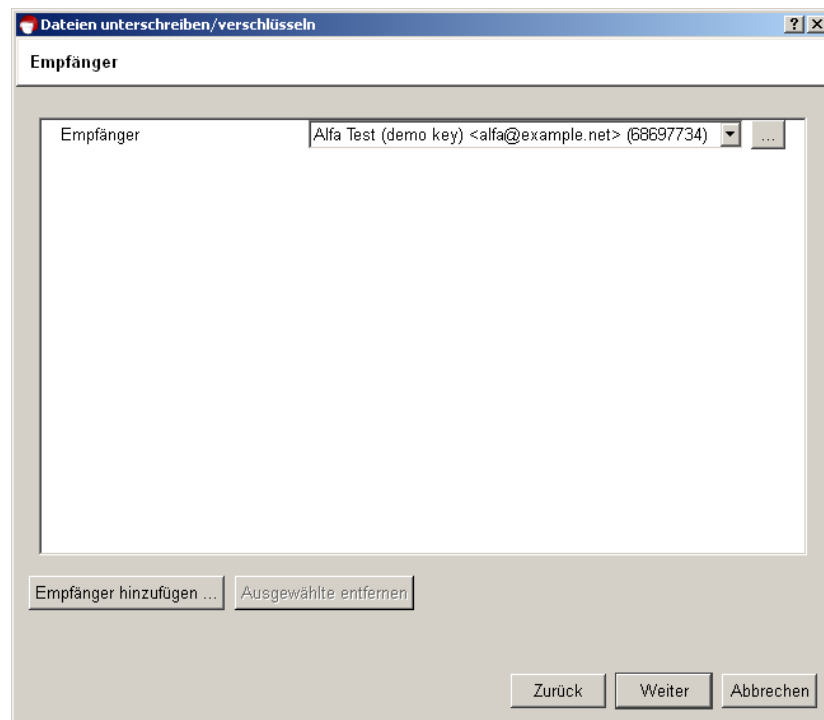
Klicken Sie auf [Weiter].

Nun haben Sie wieder die Möglichkeit, die Auswahl zu bestätigen oder zu korrigieren:



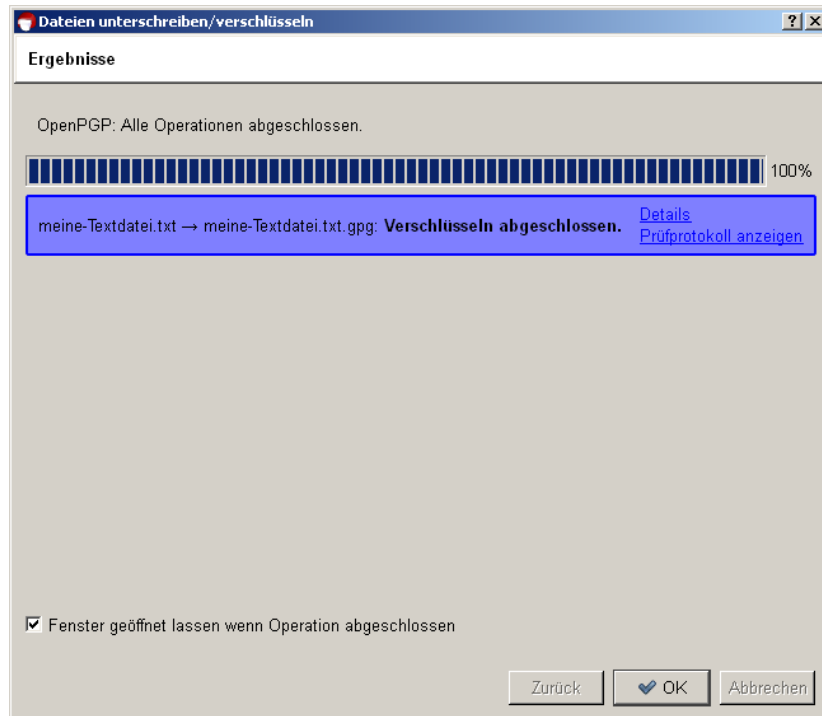
Bestätigen Sie mit [Weiter].

An wen soll die Datei verschlüsselt werden? Wählen Sie im folgenden Dialog einen oder mehrere Empfänger aus:



Klicken Sie dazu auf [Empfänger hinzufügen...] und wählen aus Ihrer Zertifikatsliste den oder die Empfänger aus. Abhängig vom gewählten Empfänger-Zertifikat und deren Typ (OpenPGP oder S/MIME) wird Ihre Datei anschließend für OpenPGP und/oder S/MIME verschlüsselt. Haben Sie also ein OpenPGP-Zertifikat *und* ein S/MIME-Zertifikat ausgewählt, werden Sie zwei verschlüsselte Dateien erhalten. Die möglichen Dateitypen der verschlüsselten Dateien finden Sie auf der nächsten Seite. Klicken Sie nun auf [Weiter], um Ihre Datei zu verschlüsseln.

Nach erfolgreicher Verschlüsselung sollte Ihr Ergebnisfenster etwa so aussehen:



Das war's! Sie haben Ihre Datei erfolgreich verschlüsselt!

Wie beim Signieren einer Datei hängt das Ergebnis von der gewählten Verschlüsselungsmethode (OpenPGP oder S/MIME) ab.

Beim Verschlüsseln Ihrer Originaldatei (hier <dateiname>.txt) sind insgesamt vier Dateitypen als Ergebnis möglich:

OpenPGP:

- <dateiname>.txt → <dateiname>.txt.gpg
- <dateiname>.txt → <dateiname>.txt.asc (bei Ausgabe als Text/ASCII-Mantel)

S/MIME:

- <dateiname>.txt → <dateiname>.txt.p7m
- <dateiname>.txt → <dateiname>.txt.pem (bei Ausgabe als Text/ASCII-Mantel)

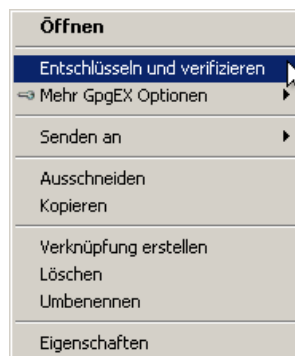
Eine der vier verschlüsselten Ergebnisdateien geben Sie nun an Ihren ausgewählten Empfänger weiter. Anders als beim Signieren einer Datei wird die unverschlüsselte Originaldatei natürlich *nicht* weitergegeben.

Datei entschlüsseln

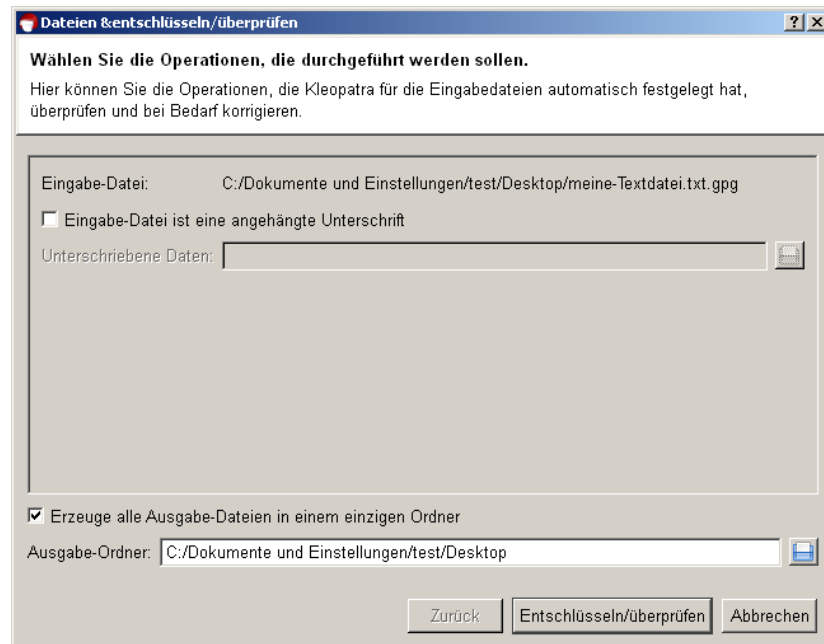
Wir wollen nun die zuvor verschlüsselte Datei zum Testen einmal entschlüsseln.

Dazu sollten Sie vorher beim Verschlüsseln auch an Ihr eigenes Zertifikat verschlüsselt haben – andernfalls können Sie die Datei nicht mit Ihrer Passphrase entschlüsseln (vgl. Kapitel 11).

Selektieren Sie die verschlüsselte Datei – also die mit der Endung *.gpg, *.asc, *.p7m oder *.pem – und wählen Sie aus dem Kontextmenü des Windows Explorer den Eintrag *Entschlüsseln und Verifizieren*:



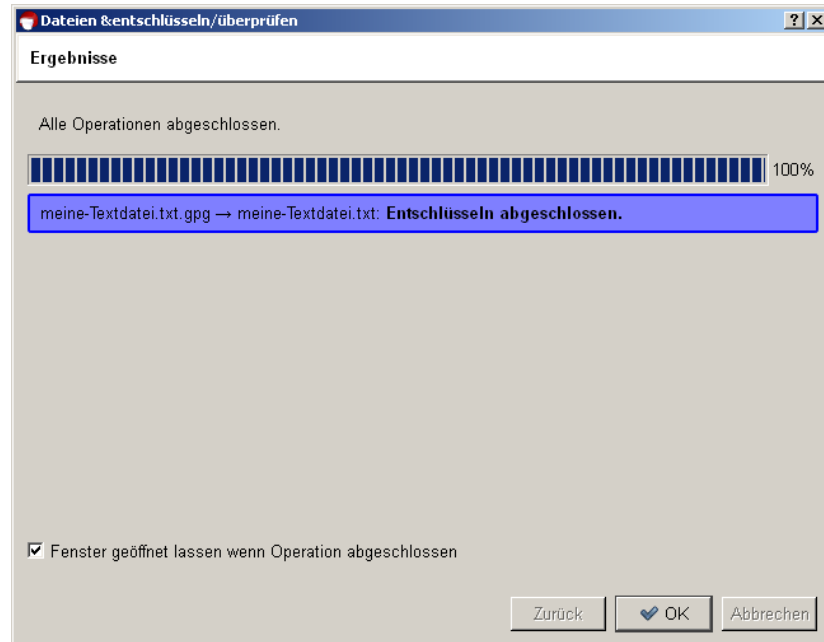
Im folgenden Entschlüsselungsdialog können Sie, bei Bedarf, noch den Ausgabe-Ordner verändern.



Klicken Sie auf [Entschlüsseln/überprüfen].

Geben Sie anschließend Ihre Passphrase ein.

Das Ergebnis zeigt, dass die Entschlüsselung erfolgreich war:



Sie sollten nun die entschlüsselte Datei problemlos entziffern oder mit einem entsprechenden Programm verwenden können.

Fassen wir kurz zusammen...

Sie haben gelernt, wie Sie mit GpgEX:

- Dateien signieren,
- signierte Dateien verifizieren,
- Dateien verschlüsseln und
- verschlüsselte Dateien entschlüsseln

können.

Gleichzeitig signieren und verschlüsseln

Ihnen ist diese Option bestimmt schon in den entsprechenden Dialogen aufgefallen. Wählen Sie sie aus, dann kombiniert GpgEX beide Aufgaben in einem Schritt.

Beachten Sie, dass immer *zuerst signiert* erst danach verschlüsselt wird.

Die Unterschrift wird also immer als geheim mitverschlüsselt. Sie kann nur von denjenigen gesehen und geprüft werden, die die Datei erfolgreich entschlüsseln konnten.

Möchten Sie Dateien signieren *und* verschlüsseln ist das derzeit nur mit OpenPGP möglich.

19. Im- und Export eines geheimen Schlüssels

In den Kapiteln 6 und 8 haben wir den Im- und Export von Zertifikaten besprochen. Wir haben Ihr eigenes Zertifikat exportiert, um es zu veröffentlichen, und wir haben das Zertifikat Ihres Korrespondenzpartners importiert und am „Schlüsselbund“ befestigt.

Dabei ging es stets um den *öffentlichen* Schlüssel – das Zertifikat. Es gibt aber auch hin und wieder die Notwendigkeit, einen *geheimen* Schlüssel zu im- oder exportieren. Wenn Sie zum Beispiel ein bereits vorhandenes (OpenPGP oder S/MIME) Schlüsselpaar mit Gpg4win weiterbenutzen wollen, müssen Sie es importieren. Oder wenn Sie Gpg4win von einem anderen Rechner aus benutzen wollen, muss ebenfalls zunächst das gesamte Schlüsselpaar dorthin transferiert werden – der öffentliche und der private Schlüssel.

19.1. Export

Immer wenn Sie einen geheimen Schlüssel auf einen anderen Rechner transferieren oder auf einer anderen Festplattenpartition bzw. einem Sicherungsmedium speichern wollen, müssen Sie mit Kleopatra eine Sicherungskopie erstellen.

So eine Sicherungskopie haben Sie evtl. schon einmal am Ende Ihrer OpenPGP-Zertifikatserzeugung angelegt. Da Ihr OpenPGP-Zertifikat aber inzwischen weitere Schlüsselunterschriften haben kann, sollte Sie die Sicherung erneut durchführen.

Öffnen Sie Kleopatra. Selektieren Sie Ihr eigenes Zertifikat und klicken Sie auf *Datei*→*Geheimen Schlüssel exportieren*.



Wählen Sie den Pfad und den Dateinamen der Ausgabedatei. Der Dateityp wird automatisch gesetzt. Abhängig davon ob Sie einen geheimen OpenPGP- oder S/MIME-Schlüssel exportieren wollen, ist standardmäßig die Dateiendung **.gpg* (OpenPGP) oder **.der* (S/MIME) ausgewählt. Bei diesen Dateien handelt es sich um Binärdateien – eine Betrachtung im Texteditor ist hier also sinnlos.

Bei Aktivierung der Option *ASCII-Mantel* erhalten Sie die Dateiendung **.asc* (OpenPGP) bzw. **.pem* (S/MIME). Diese beiden Dateitypen sind mit jedem Texteditor lesbar – Sie würden dort den üblichen Buchstaben- und Ziffernsalat sehen. Ob Sie diese Option nutzen oder nicht ist eigentlich gleichgültig; Gpg4win kommt mit beiden Arten klar.

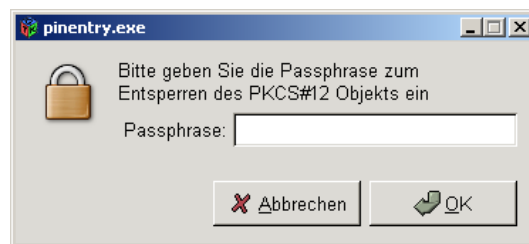
Beide Schlüsselteile – der öffentliche und der geheime – werden von Kleopatra in dieser einen Datei abgespeichert.

Achtung: Behandeln Sie diese Datei sehr sorgfältig. Sie enthält Ihren geheimen Schlüssel und damit sicherheitskritische Informationen!

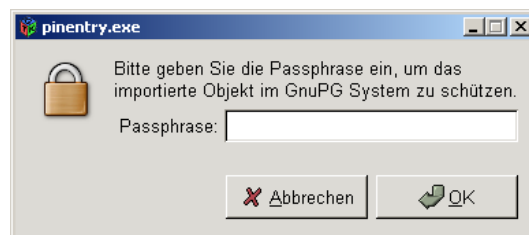
19.2. Import

Zum Importieren Ihres vorher exportierten geheimen Schlüssels in Kleopatra, gehen Sie so vor, wie Sie es beim Importieren von (fremden) Zertifikaten gewohnt sind (vgl. Kapitel 8):

Klicken Sie auf *Datei*→*Zertifikat importieren...* und wählen Sie die zu importierende Datei aus. Handelt es sich um eine PKCS12-Datei (z.B. vom Typ *.p12), so werden Sie zunächst nach der Passphrase zum Entsperren des privaten Schlüssels gefragt:



Setzen Sie nun eine (ggf. neue) Passphrase, die nach dem Importvorgang Ihrem privaten Schlüssel zugeordnet werden soll:



Wiederholen Sie Ihre Passphraseeingabe. Sollte Ihre Passphrase zu kurz oder nur aus Buchstaben bestehen, werden Sie entsprechend gewarnt.

Nach erfolgreichem Importieren erhalten Sie einen Infodialog, der Ihnen die Ergebnisse des Importvorgangs auflistet; hier am Beispiel eines geheimen OpenPGP-Schlüssels:



Kleopatra hat damit sowohl den geheimen als auch den öffentlichen Schlüssel aus der Sicherungsdatei importiert. Ihr Zertifikat ist damit unter „Meine Zertifikate“ in der Zertifikatsverwaltung von Kleopatra sichtbar.

Achtung: Löschen Sie danach unbedingt die oben erstellte Sicherungskopie Ihres geheimen Schlüssels von Ihrer Festplatte und sichern Sie vorher diese wichtige Datei möglichst irgendwo auf einem externen Medium. Denken Sie daran die gelöschte Datei aus Ihrem „Papierkorb“ zu entfernen. Andernfalls stellt diese Datei ein Sicherheitsrisiko für Ihre geheime E-Mail-Verschlüsselung dar!

Anmerkung: Es kann in einigen Fällen vorkommen, dass Sie einen importierten Schlüssel nicht direkt benutzen können. Dies äußert sich darin, dass Sie die richtige Passphrase eingeben, dieser aber nicht akzeptiert wird. Das kommt daher, dass einige Versionen von PGP intern den IDEA Algorithmus verwenden. Dieser kann von GnuPG aus rechtlichen Gründen nicht unterstützt werden. Um das Problem zu beheben, ändern Sie in PGP einfach die Passphrase und exportieren/importieren Sie den Schlüssel erneut. Sollte dies auch nicht funktionieren, so setzen Sie die Passphrase in PGP auf „leer“; d.h. auf keinen Schutz und exportieren/importieren Sie wieder – In diesem Fall müssen Sie unbedingt sicherstellen, sowohl die **Datei sicher zu löschen als auch in PGP und in Gpg4win danach wieder eine echte Passphrase zu setzen.**

Herzlichen Glückwunsch! Sie haben damit erfolgreich Ihr Schlüsselpaar exportiert und wieder importiert.

20. Systemweite Konfigurationen und Vorbelegungen für S/MIME



Im Rahmen von Softwareverteilung oder sonstigen Umgebungen in denen viele Anwender auf einem System arbeiten, ist es sinnvoll einige systemweite Vorgaben und Vorbelegungen für Gpg4win einzurichten.

Das betrifft vor allem S/MIME, denn bei streng vorgegebenen Vertrauensketten macht es Sinn dass die Anwender die Informationen dazu miteinander Teilen.

Einige typische systemweite Einrichtungen sind:

- Vertrauenswürdige Wurzel-Zertifikate

Um zu vermeiden, dass jeder Anwender selbst die notwendigen Wurzelzertifikate suchen und installieren sowie Vertrauenswürdigkeit prüfen und setzen muss, ist eine systemweite Vorbelegung der wichtigsten Wurzel-Zertifikate sinnvoll.

Dafür sind folgende Schritte durchzuführen:

1. Die Wurzel-Zertifikate ablegen wie unter Abschnitt 22.3 beschrieben.
2. Die vertrauenswürdigen Wurzeln definieren wie unter Abschnitt 22.6 beschrieben.

- Direkt verfügbare CA-Zertifikate

Um den Anwendern zusätzlich die Mühe zu ersparen, die Zertifikate der Zertifizierungsstellen (Certificate Authorities, CAs) zu suchen und zu importieren, ist auch hier eine systemweite Vorbelegung der wichtigsten CA-Zertifikate sinnvoll.

Folgen Sie dazu der Beschreibung unter Abschnitt 22.4.

- Proxy für Verzeichnisdienst-Suche

Es kommt vor, dass interne Netzwerke keine direkten Verbindungen der einzelnen Systeme nach aussen zulassen, sondern einen sogenannten Proxy vorsehen.

Ist dies in Ihrem Netzwerk auch für die bei GnuPG bzw. S/MIME wichtigen LDAP-Abfragen der Fall, so führen Sie folgende Schritte durch:

1. Stellen Sie LDAP-Verzeichnisdienst-Suchen auf Ihren Proxy wie unter Abschnitt 22.5 ein.
2. Stellen Sie Sperrlisten-Suchen auf Ihren Proxy ein, in dem Sie einen Eintrag wie z.B.:
`http-proxy http://proxy.mydomain.example:8080`
(ggf. analog für LDAP) als Administrator in die Datei
`C:\Dokumente und Einstellungen\All Users\GNU\etc\dirmgr\dirmgr.conf`
eintragen.
3. Starten Sie den DirMgr neu (siehe Abschnitt 21.3).

21. Bekannte Probleme und was man tun kann

21.1. GpgOL Menüs und Dialog nicht mehr in Outlook zu finden

Es kann vorkommen, dass trotz Aktualisierung von Gpg4win die Menüs und Dialog die von GpgOL zu Outlook hinzugefügt werden nicht mehr zu finden sind.

Das ist dann der Fall wenn ein technisches Problem auftrat und Outlook aus diesem Grund das GpgOL Element deaktiviert.

Reaktivieren Sie GpgOL über das Outlook-Menü *Hilfe*→*Info*→*Deaktivierte Elemente*.

21.2. GpgOL Icons zum Signieren/Verschlüsseln fehlen in der Symbolleiste

Wenn bereits viele Icons in der Symbolleiste des Nachrichtenfensters vorhanden sind, so zeigt Outlook die Signieren/Verschlüsseln-Icons nicht unbedingt direkt an.

Sie können diese aber anzeigen lassen, indem Sie in der Symbolleiste auf das kleine Icon mit dem Pfeil nach unten klicken (*Optionen für Symbolleiste*): Sie erhalten eine Übersicht aller nicht angezeigten Icons. Ein Klick auf eines dieser Icons verschiebt es in den sichtbaren Teil der Symbolleiste.

21.3. Keine S/MIME Operationen mehr möglich (Systemdienst „DirMngr“ läuft nicht)



Der *Directory Manager (DirMngr)* ist ein über Gpg4win installierter Dienst, der die Zugriffe auf Verzeichnisdienste (z.B. LDAP) verwaltet. Eine der häufigsten Aufgaben ist das Laden von Sperrlisten für S/MIME Zertifikate.

Es kann vorkommen, dass die S/MIME Operationen (Signatur, Prüfung, Ver- oder Entschlüsselung) nicht durchgeführt werden können, weil „DirMngr“ nicht verfügbar ist. In der Voreinstellung von Gpg4win ist es zwingend notwendig, dass „DirMngr“ die Sperrliste prüft, geschieht das nicht, darf die jeweilige Operation nicht ausgeführt werden, da möglicherweise ein kompromittiertes Zertifikat genutzt wird.

Abhilfe schaffe ein Neustart des „DirMngr“ durch den Systemadministrator. Dies erfolgt über *Systemsteuerung*→*Verwaltung*→*Dienste*. In der Liste finden Sie „DirMngr“ – über das Kontextmenü kann der Dienst neu gestartet werden.

22. Wo finde ich die Dateien und Einstellungen von Gpg4win?

22.1. Persönliche Einstellungen der Anwender

Die persönlichen Einstellungen für jeden Anwender befinden sich im Verzeichnis %APPDATA%\gnupg. Oft entspricht das dem Verzeichnis:

```
C:\Dokumente und Einstellungen\<name>\Anwendungsdaten\gnupg\
```

Beachten Sie, dass es sich um ein verstecktes Verzeichnis handelt. Um es sichtbar zu machen, müssen Sie im Explorer über das Menü *Extras*→*Ordneroptionen* im Reiter *Ansicht* die Option *Alle Dateien und Ordner anzeigen* unter der Rubrik *Versteckte Dateien und Ordner* aktivieren.

In diesem gnupg-Verzeichnis befinden sich sämtliche persönlichen GnuPG Daten, also die persönlichen Schlüssel, Zertifikate, Vertrauenseinstellungen und Programmkonfigurationen. Bei einer Deinstallation von Gpg4win wird dieses Verzeichnis *nicht* gelöscht. Denken Sie daran, sich regelmäßig Sicherheitskopien von diesem Verzeichnis anzulegen.

22.2. Zwischengespeicherte Sperrlisten

S/MIME

Der systemweite Dienst „DirMngr“ (Directory Manager) prüft unter anderem, ob ein X.509-Zertifikat gesperrt ist und daher nicht verwendet werden darf. Dafür werden Sperrlisten von den Ausgabestellen der Zertifikate („Trust-Center“) abgeholt und für die Dauer ihrer Gültigkeit zwischengespeichert.

Abgelegt werden diese Sperrlisten unter:

```
C:\Dokumente und Einstellungen\LocalService\Lokale  
Einstellungen\Anwendungsdaten\GNU\cache\dirmngr\crs.d\
```

Hierbei handelt es sich um *geschützte* Dateien, die standardmäßig vom Explorer nicht angezeigt werden. Sollten Sie dennoch die Anzeige dieser Dateien wünschen, deaktivieren Sie die Option *Geschützte Systemdateien ausblenden* in den Ansicht-Einstellungen des Windows Explorer.

In diesem Verzeichnis sollten keine Änderungen vorgenommen werden.

22.3. Vertrauenswürdige Wurzeln von DirMngr

S/MIME

Für eine vollständige Prüfung von X.509-Zertifikats-Gültigkeiten muss auch den Wurzel-Zertifikaten vertraut werden, in deren Zertifizierungskette die Sperrlisten unterschrieben wurden.

Die Wurzel-Zertifikate, denen der DirMngr bei den Prüfungen vertrauen soll, müssen im folgenden Verzeichnis abgelegt werden:

C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\etc\dirmngr\trusted-certs\

Wichtig: Die entsprechenden Wurzel-Zertifikate müssen als DER-Dateien mit Dateinamen-Endung „.crt“ im o.g. Verzeichnis vorliegen.

Der DirMngr läuft als systemweiter Dienst und muss nach Änderungen im „trusted-certs“-Verzeichnis neu gestartet werden. Anschließend sind die dort abgelegten Wurzelzertifikaten für alle Anwender vertrauenswürdig.

Beachten Sie auch Abschnitt 22.6, um den Wurzel-Zertifikaten vollständig (systemweit) zu vertrauen.

22.4. Weitere Zertifikate von DirMngr



Um, wie oben beschrieben, die X.509-Zertifizierungskette zu prüfen sind auch die Zertifikate der Zertifizierungsstellen (Certificate Authorities, CAs) zu prüfen.

Für eine direkte Verfügbarkeit können sie in diesem (systemweiten) Verzeichnis abgelegt werden:

C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\lib\dirmngr\extra-certs\

Zertifikate, die nicht hier oder bei den Anwendern vorliegen, müssen entweder automatisch von LDAP-Servern geladen werden oder (falls so nicht verfügbar) per Hand importiert werden.

Es ist also sinnvoll im Rahmen von systemweiten Vorgaben hier die wichtigsten CA-Zertifikate abulegen.

22.5. Konfiguration zur Verwendung externer LDAP Verzeichnisdienste



GnuPG kann so konfiguriert werden, dass bei Bedarf fehlende X.509-Zertifikate oder Sperrlisten auf externen Verzeichnisdiensten gesucht werden.

Der Systemdienst „DirMngr“ verwendet dafür die Liste der Dienste, die in der Datei

C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\etc\dirmngr\ldapservers.conf angegeben sind.

Sind im internen Netz die Zugänge zu externen LDAP-Servern gesperrt, so kann man in dieser Datei einen Proxy-Dienst für entsprechende Durchleitung konfigurieren, wie folgende Zeile im Beispiel illustriert:

```
proxy.mydomain.example:389:::O=myorg,C=de
```

Die genaue Syntax für die Einträge lautet übrigens:

```
HOSTNAME:PORT:USERNAME:PASSWORD:BASE_DN
```

22.6. Systemweite vertrauenswürdige Wurzel-Zertifikate



Die systemweit als vertrauenswürdig vorgelegten Wurzel-Zertifikate werden in der Datei `C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\etc\gnupg\trustlist.txt` definiert.

Die genaue Syntax für die Einträge lautet hier:

```
FINGERPRINT S
```

also z.B.:

```
BA90087D0C6C7F4DEAF00907BCFA2133DDC8CA90 S
```

Wichtig: Damit Wurzel-Zertifikate in Kleopatra vollständig als vertrauenswürdig markiert werden (Zertifikat wird blau hinterlegt), müssen die Wurzel-Zertifikate zusätzlich für den DirMngr abgelegt werden, wie unter Abschnitt 22.3 beschrieben.

23. Fehler in den Gpg4win-Programmen aufspüren

Es kann vorkommen, dass eines der Gpg4win-Programme nicht wie erwartet zu funktionieren scheint. Nicht selten ist dabei eine Besonderheit der Arbeitsumgebung verantwortlich, so dass die Software-Entwickler von Gpg4win das beobachtete Problem gar nicht selbst nachvollziehen können.

Um die Software-Entwickler bei der Problemsuche zu unterstützen oder um auch einmal selbst in die technischen Detail-Abläufe reinzuschnuppeln, bieten die Gpg4win-Programme Unterstützung an.

In der Regel muss diese Unterstützung aber erst einmal eingeschaltet werden. Eine der wichtigsten Hilfsmittel sind Logbücher. Dort werden detaillierte Informationen zu den technischen Vorgängen vermerkt. Ein Software-Entwickler kann ein Problem und die mögliche Lösung oft leicht daran ablesen, auch wenn es auf den ersten Blick sehr unverständlich und viel zu umfangreich wirken mag.

Wenn Sie einen Fehler-Bericht an die Software-Entwickler senden wollen, so finden Sie auf dieser Web-Seite einige Hinweise:

<http://www.gpg4win.de/reporting-bugs-de.html>

Logbücher – unter o.g. URL als „Debug-Informationen“ bezeichnet – sind besonders wertvoll und sollten bei einem Fehler-Bericht mitgeschickt werden.

In diesem Kapitel wird beschrieben, wie Sie Programmablauf-Informationen (darum handelt es sich letztlich bei den Logbüchern) zu den einzelnen Gpg4win-Programmen einschalten können.

23.1. Logbuch von Kleopatra einschalten

Das Logbuch von Kleopatra besteht aus vielen Dateien, daher ist der erste Schritt ein Verzeichnis für das Logbuch zu erstellen. Denkbar ist z.B. `C:\TEMP\kleologdir`.

Bitte beachten Sie hierbei, dass es hier um Einstellungen des Anwenders, nicht des Systemadministrators geht. Die Einstellungen müssen also für jeden Anwender der ein Logbuch erstellen möchte separat vorgenommen werden und dabei insbesondere aufgefassen werden, dass unterschiedliche `kleologdir` Verzeichnisse verwendet werden.

Der Pfad zu diesem Verzeichnis muss nun in der neuen Umgebungsvariable `KLEOPATRA_LOGDIR` vermerkt werden:

Öffnen Sie dazu die Systemsteuerung, wählen dort *System*, dann den Reiter *Erweitert* und schließlich den Knopf [Umgebungsvariablen].

Fügen Sie dort folgende neue *Benutzervariable* ein:

Name der Variable: `KLEOPATRA_LOGDIR`

Wert der Variable: `C:\TEMP\kleologdir`

Beachten Sie, dass das angegebene Verzeichnis existieren muss. Sie können es auch nachträglich erstellen.

Um die Log-Funktion wirksam werden zu lassen, muss Kleopatra beendet und neu gestartet werden. Spätestens jetzt muss das Log-Verzeichnis erstellt worden sein.

Während Kleopatra nun verwendet wird, zeichnet es viele Daten in die Datei `kleo-log` (Haupt-Logbuch) sowie möglicherweise viele Dateien `pipe-input-ZEITSTEMPEL-ZUFALLSZEICHEN`.

Möglicherweise reichen diese Informationen einem Software-Entwickler nicht, um den Fehler zu erkennen. Er wird Sie dann bitten, eine weitere Umgebungsvariable anzulegen – so wie Sie es schon oben getan haben:

Name der Variable: `KLEOPATRA_LOGPTIONS`

Wert der Variable: `all`

Möglicherweise werden die Logbuch-Dateien sehr schnell sehr groß. Sie sollten diese Logbuch-Aufzeichnung nur einschalten und dann ein bestimmtes Fehlverhalten durchspielen. Anschliessend schalten Sie die Aufzeichnung wieder aus indem Sie die Umgebungsvariable löschen oder den Namen leicht variieren (für späteres leichtes reaktivieren). Vergessen Sie nicht, die Logbücher zu löschen falls sie umfangreich geworden sind oder es sehr viele Dateien sind. Am besten immer bevor Sie eine neue Aufzeichnung machen.

23.2. Logbuch von GpgOL einschalten

Für das Einschalten des Logbuches von GpgOL müssen Sie mit Ihrem normalen Benutzerkonto (also nicht als Administrator) den Registrierungs-Editor starten. Geben Sie dazu das Kommando `regedit` unter *Start*→*Ausführen* oder in einer Eingabeaufforderung ein.

Wählen Sie nun aus dem Verzeichnisbaum auf der linken Seite das folgende GpgOL-Verzeichnis aus:
`HKEY_CURRENT_USER\Software\GNU\GpgOL`

Auf der rechten Seite sehen Sie nun eine Liste von Einträgen (sogenannte Zeichenfolgen) mit teilweise bereits vordefinierten Werten. Diese Einträge werden nach dem ersten Start von Outlook mit GpgOL angelegt.

Zum Aktivieren des GpgOL-Logbuchs Doppelklicken Sie auf den Eintrag `enableDebug` und setzen Sie dessen Wert auf 1.

Je größer Sie den Wert von `enableDebug` wählen, umso mehr interne Programmablaufinformationen werden im Logbuch gespeichert. Es ist empfehlenswert mit 1 zu beginnen und nur höhere Werte einzusetzen, falls es eine tiefere Programmablaufanalyse erfordern sollte.

Als Wert für `logFile` geben Sie nun einen Dateinamen an, wohin das Logbuch geschrieben werden soll; z.B.: `C:\TEMP\gpgol.log`.

Starten Sie Outlook neu, um die Aufzeichnung zu starten.

Bedenken Sie, dass diese Datei ggf. im weiteren Verlauf sehr umfangreich werden kann. Stelle Sie `enableDebug` auf 0, sobald Sie das GpgOL-Logbuch nicht mehr benötigen.

23.3. Logbuch von DirMngr einschalten

Bei DirMngr handelt es sich um einen systemweiten Dienst und daher ist das Einschalten des Logbuches nur mit Administrator-Rechten möglich.

Um das Logbuch einzuschalten, tragen Sie folgende zwei Zeilen in die Datei
`C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\etc\dirmgr\dirmgr.conf`
ein:

```
debug-all
log-file C:\TEMP\dirmgr.log
```

Starten Sie anschließend den Dienst unter *Systemsteuerung*→*Verwaltung*→*Dienste* neu, so dass die geänderte Konfiguration eingelesen werden.

23.4. Logbuch von GnuPG einschalten

Für folgende GnuPG-Programme können Sie jeweils einzeln ein Logbuch einschalten:

- GPG Agent
- GPG für S/MIME
- GPG für OpenPGP
- Smartcard Daemon

Für diese Programme können Anwender persönliche Konfigurationen vornehmen. Dazu gehört auch das Einstellen einer Protokolldatei für den Programmablauf.

Eingeschaltet wird das jeweilige Logbuch im GnuPG Backend – erreichbar über das Kleopatra Menü *Einstellungen* → *GnuPG Backend einrichten...* Für jede der o.g. vier Programme existieren in diesem Konfigurationsfenster zwei Debug-Optionen:

- **Option „Setze die Debug-Stufe auf“**
Hier definieren Sie die Ausführlichkeit der aufzuzeichnenden Informationen. Die Debugstufe *guru* ist die höchste Stufe und erzeugt entsprechend große Dateien. Schalten Sie daher die Logbücher wieder aus (Debugstufe *keine*), wenn Sie diese nicht mehr benötigen.
- **Option „Schreibe im Servermodus Logs auf DATEI“**
Geben Sie hier die Logdatei an, in der alle Debug-Informationen gespeichert werden sollen; z.B. `C:\TEMP\gpgsm.log`.

Starten Sie anschließend Kleopatra neu (ggf. müssen Sie zuvor einen noch laufenden `gpg-agent` über den Task-Manager beenden – oder aber Sie loggen sich aus und melden sich neu an Ihrem Windows-System an).

24. Warum Gpg4win nicht zu knacken ist . . .

. . . jedenfalls nicht mit heute bekannten Methoden und sofern die Implementierung der Programme frei von Fehlern ist.

In der Realität sind genau solche Fehler in den Programmen, im Betriebssystem oder nicht zuletzt in der Benutzung der letzte Weg, um doch noch an die geheimen Informationen zu gelangen. – Auch deshalb sollte Sie dieses Kompendium bis hierhin gelesen haben.

In jedem Beispiel dieses Kompendiums haben Sie gesehen, dass zwischen dem geheimen und dem öffentlichen Schlüsselteil eine geheimnisvolle Verbindung besteht. Nur wenn beide zueinander passen, kann man Geheimbotschaften entschlüsseln.

Das Geheimnis dieser mathematischen Verbindung müssen Sie nicht unbedingt kennen – Gpg4win funktioniert für Sie auch so. Man kann diese komplexe mathematische Methode aber auch als Normalsterblicher und Nichtmathematiker verstehen. Sie müssen eigentlich nur einfache Additionen ($2 + 3$) und Multiplikationen ($5 * 7$) beherrschen. Allerdings in einer ganz anderen Rechenmethode als der, die Sie im Alltag benutzen. Es gehört sowohl zur Sicherheitsphilosophie der Kryptographie wie auch zum Prinzip der Freien Software, dass es keine geheimnisvollen Methoden und Algorithmen gibt. Letztendlich versteht man auch erst dann wirklich, warum GnuPG (die eigentliche Maschinerie hinter Gpg4win) sicher ist.

Hier beginnt also sozusagen die Kür nach dem Pflichtteil:

25. GnuPG und das Geheimnis der großen Zahlen

Kryptographie für Nicht-Mathematiker

Es ist schon versucht worden, den RSA Algorithmus, auf dem GnuPG basiert¹, zu „knacken“, also einen privaten Schlüssel zu berechnen, wenn man lediglich den öffentlichen Schlüssel kennt. Diese Berechnung ist aber noch nie für Schlüssellängen (1024 Bit und mehr), die in GnuPG verwendet werden, gelungen. Es ist zwar *theoretisch* möglich, aber *praktisch* nicht durchführbar! Denn selbst bei genügend vorhandener Zeit (viele Jahre) und Abertausenden von vernetzten Rechnern würde niemals genügend Speicher zur Verfügung stehen, um den letzten Schritt dieser Berechnung durchführen zu können.

Es kann allerdings durchaus möglich sein, dass eines Tages eine geniale Idee die Mathematik revolutioniert und eine schnelle Lösung des mathematischen Problems, welches hinter RSA steckt, liefert. Dies wird aber wohl kaum von heute auf morgen geschehen. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) veröffentlicht von Zeit zu Zeit Prognosen und Einschätzungen, welche Schlüssellängen noch wieviele Jahre für absolute Geheimhaltung benutzt werden sollen. GnuPG überschreitet mit seinen Standardeinstellungen noch weit diese Mindestanforderungen. Wie im vorigen Kapitel schon angerissen, ist die Mathematik der mit Abstand sicherste Teil an der ganzen praktisch angewandten Kryptographie.

¹Wir verwenden hier RSA als Beispiel. RSA ist einfacher zu verstehen als der Elgamal Algorithmus, der als Voreinstellung von GnuPG benutzt wird.

Im Folgenden erfahren Sie, wie diese mathematische Methode funktioniert. Nicht in allen Einzelheiten (das würde den Rahmen dieser Anleitung bei weitem sprengen), aber doch so, dass Sie bei etwas Mitrechnen selbst mathematisch korrekt ver- und entschlüsseln können und dabei das „Geheimnis der großen Zahlen“ entdecken.

Man kann diese komplexe mathematische Methode auch als Normalsterblicher und Nichtmathematiker verstehen. Sie müssen nur einfache Additionen und Multiplikationen beherrschen. Wie gesagt: Hier beginnt der Kürteil, und bei der Kür geht es immer etwas mehr zur Sache als im Pflichtprogramm. Letztendlich versteht man dann aber, warum GnuPG sicher ist.

Eine Begriffsklärung vorneweg:

*Ein **Algorithmus** ist eine mathematische Prozedur zur Veränderung oder Transformation von Daten oder Informationen.*

***Arithmetik** ist die Methode, nach der wir Zahlen addieren und multiplizieren.*

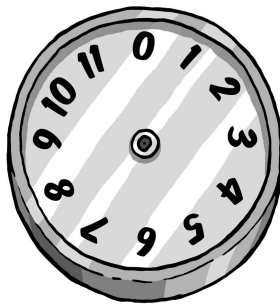
Die Verschlüsselung mit GnuPG basiert auf dem sogenannten RSA-Algorithmus². RSA steht für die Nachnamen von Ron Rivest, Ami Shamir und Ben Adleman, die diesen Algorithmus im Jahr 1978 entdeckt haben. Dieser Algorithmus verwendet einen Typ der Arithmetik, die Rechnen mit Restklassen oder „Modulo-Arithmetik“ heißt.

²RSA ist eigentlich optional, da aus Patentgründen der Elgamal Algorithmus, beruhend auf dem schwieriger zu erklärenden Problem des diskreten Logarithmus, als Standard in GnuPG verwendet wird.

25.1. Das Rechnen mit Restklassen

Wenn man mit Restklassen rechnet, so bedeutet dies, dass man nur mit dem „Rest“ rechnet, der nach einer ganzzahligen Teilung durch eine bestimmte Zahl übrigbleibt. Diese Zahl, durch die geteilt wird, nennt man den „Modul“ oder die „Modulzahl“. Wenn wir beispielsweise mit dem Teiler oder der Modulzahl 5 rechnen, sagen wir auch, „wir rechnen modulo 5“.

Wie das Rechnen mit Restklassen – auch Modulo-Arithmetik oder Kongruenzrechnung genannt – funktioniert, kann man sich gut klarmachen, wenn man sich das Zifferblattes einer Uhr vorstellt:



Diese Uhr ist ein Beispiel für das Rechnen mit modulo 12 (der Teiler ist also 12) – eine Uhr mit einem normalen Zifferblatt, allerdings mit einer 0 anstelle der 12. Wir können damit Modulo-Arithmetik betreiben, indem wir einfach den gedachten Zeiger bewegen.

Um beispielsweise $3 + 2$ zu rechnen, beginnen wir bei der Ziffer 2 und drehen den Zeiger um 3 Striche weiter (oder wir starten bei der 3 und drehen 2 Striche weiter, was natürlich auf dasselbe hinausläuft). Das Ergebnis ist 5.

Zählt man auf diese Weise $7 + 8$ zusammen, erhält man 3. Denn 3 ist der Rest, wenn man 15 (also $7 + 8$) durch 12 teilt. Um 5 mit 7 zu multiplizieren, beginnt man bei 0 und dreht 7 mal jeweils um 5 Striche weiter (oder auch bei 0 beginnend 5 mal um 7 Striche). In beiden Fällen bleibt der Zeiger bei 11 stehen. Denn 11 ist der Rest, wenn 35 (also $7 * 5$) durch 12 geteilt wird.

Beim Rechnen mit Restklassen addieren und teilen wir Zahlen also nach den normalen Regeln der Alltagsarithmetik, verwenden dabei jedoch immer nur den Rest nach der Teilung. Um anzuzeigen, dass wir nach den Regeln der Modulo-Arithmetik und nicht nach denen der üblichen Arithmetik rechnen, schreibt man den Modul (Sie wissen schon – den Teiler) dazu. Man sagt dann zum Beispiel „4 modulo 5“, schreibt aber kurz „4 mod 5“.

Bei Modulo-5 zum Beispiel hat man dann eine Uhr, auf deren Zifferblatt es nur die 0, 1, 2, 3 und 4 gibt. Also:

$$4 \text{ mod } 5 + 3 \text{ mod } 5 = 7 \text{ mod } 5 = 2 \text{ mod } 5$$

Anders ausgedrückt, ist in der Modulo-5 Arithmetik das Ergebnis aus 4 plus 3 gleich 2. Wir können also auch schreiben:

$$9 \text{ mod } 5 + 7 \text{ mod } 5 = 16 \text{ mod } 5 = 1 \text{ mod } 5$$

Wir sehen auch, dass es egal ist, in welcher Reihenfolge wir vorgehen, weil wir nämlich auch schreiben können:

$$9 \text{ mod } 5 + 7 \text{ mod } 5 = 4 \text{ mod } 5 + 2 \text{ mod } 5 = 6 \text{ mod } 5 = 1 \text{ mod } 5$$

Denn 4 ist dasselbe wie 9, und 2 dasselbe wie 7, da wir uns ja nur für den jeweiligen Rest nach der Teilung durch 5 interessieren. Daran wird deutlich, dass wir bei dieser Art der Arithmetik jederzeit 5 oder ein Vielfaches von 5, wie 10, 15 und so weiter nehmen können, und das Ergebnis stets dasselbe ist.

Das funktioniert auch beim Multiplizieren (Malnehmen).

Ein Beispiel:

$$4 \bmod 5 * 2 \bmod 5 = 8 \bmod 5 = 3 \bmod 5$$

Ebenso können wir schreiben:

$$9 \bmod 5 * 7 \bmod 5 = 63 \bmod 5 = 3 \bmod 5$$

da wir einfach 60, also $5 * 12$, abziehen können.

Man könnte aber auch schreiben:

$$9 \bmod 5 * 7 \bmod 5 = 4 \bmod 5 * 2 \bmod 5 = 8 \bmod 5 = 3 \bmod 5$$

denn 4 entspricht 9, und 2 entspricht 7, wenn wir nur den Rest nach Teilung durch 5 betrachten.

Widerum stellen wir fest, dass es egal ist, wenn wir das Vielfache von 5 einfach weglassen.

Da dadurch alles einfacher wird, machen wir das, bevor wir Zahlen addieren oder multiplizieren. Das bedeutet, dass wir uns lediglich um die Zahlen 0, 1, 2, 3 und 4 kümmern müssen, wenn wir mit der Modulo-5 Arithmetik rechnen. Denn wir können ja alles, was durch 5 teilbar ist, weglassen. Dazu noch drei Beispiele:

$$5 \bmod 11 * 3 \bmod 11 = 15 \bmod 11 = 4 \bmod 11$$

$$2 \bmod 7 * 4 \bmod 7 = 1 \bmod 7$$

$$13 \bmod 17 * 11 \bmod 17 = 7 \bmod 17$$

Das letzte Beispiel wird klar, wenn man bedenkt, dass in normaler Arithmetik gerechnet $13 * 11 = 143$ und $143 = 8 * 17 + 7$ ist.

25.2. RSA-Algorithmus und Rechnen mit Restklassen

Computer speichern Buchstaben als Zahlen. Alle Buchstaben und Symbole auf der Computertastatur werden in Wirklichkeit als Zahlen gespeichert, die zwischen 0 und 255 liegen.

Wir können also eine Nachricht auch in eine Zahlenfolge umwandeln. Nach welcher Methode (oder Algorithmus) dies geschieht, wird im nächsten Abschnitt beschrieben. Darin stellen wir Ihnen die Methode vor, nach der die Verschlüsselung mit GnuPG funktioniert: den RSA Algorithmus. Dieser Algorithmus wandelt eine Zahlenfolge (die ja eine Nachricht darstellen kann) so in eine andere Zahlenfolge um (Transformation), dass die Nachricht dabei verschlüsselt wird. Wenn man dabei nach dem richtigen Verfahren vorgeht, wird die Nachricht sicher kodiert und kann nur noch vom rechtmäßigen Empfänger dekodiert werden. Das sind die Grundlagen des RSA Algorithmus:

Sie selbst haben bei der Installation von Gpg4win während der Eingabe Ihrer Passphrase zwei große Primzahlen erzeugt, ohne es zu bemerken (dieser werden mit p und q bezeichnet). Nur Sie – oder in der Praxis Ihr Computer – kennen diese beiden Primzahlen, und Sie müssen für ihre Geheimhaltung sorgen.

Es werden daraus nun drei weitere Zahlen erzeugt:

Die erste Zahl ist das Ergebnis der Multiplikation der beiden Primzahlen, also ihr Produkt. Dieses Produkt wird als Modulus und dem Buchstaben n bezeichnet. Dies ist der Modul mit dem wir später immer rechnen werden.

Die zweite Zahl ist der sogenannte öffentliche Exponent und eine Zahl an die bestimmte Anforderungen gestellt werden (teilerfremd zu $(p - 1)(q - 1)$); sie wird mit e bezeichnet. Häufig wird hier 3, 41 oder 65537 benutzt.

Die dritte Zahl wird errechnet aus dem öffentlichem Exponent (der zweiten Zahl) und den beiden Primzahlen. Diese Zahl ist der geheime Exponent und wird mit d bezeichnet. Die komplizierte Formel zur Berechnung lautet:

$$d = e^{-1} \text{ mod } (p - 1)(q - 1)$$

Die erste und die zweite Zahl werden veröffentlicht – das ist Ihr öffentlicher Schlüssel. Beide werden dazu benutzt, Nachrichten zu verschlüsseln. Die dritte Zahl muss von Ihnen geheimgehalten werden – es ist Ihr geheimer Schlüssel. Die beiden Primzahlen werden danach nicht mehr benötigt.

Wenn eine verschlüsselte Nachricht empfangen wird, kann sie entschlüsselt werden mit Hilfe der ersten (n) und der dritten Zahl (d). Nur der Empfänger kennt beide Schlüsselteile – seinen öffentlichen und seinen geheimen Schlüssel. Der Rest der Welt kennt nur den öffentlichen Schlüssel (n und e).

Die Trick des RSA Algorithmus liegt nun darin, dass es unmöglich ist, aus dem öffentlichen Schlüsselteil (n und e) den geheimen Schlüsselteil (d) zu errechnen und damit die Botschaft zu entschlüsseln – denn: Nur wer im Besitz von d ist, kann die Botschaft entschlüsseln.

25.3. RSA Verschlüsselung mit kleinen Zahlen

Wir verwenden hier erst einmal kleine Zahlen, um deutlich zu machen, wie die Methode funktioniert. In der Praxis verwendet man jedoch viel größere Primzahlen, die aus zig Ziffern bestehen.

Nehmen wir die Primzahlen 7 und 11. Damit verschlüsseln wir Zahlen – oder Buchstaben, was für den Computer dasselbe ist – nach dem RSA Algorithmus.

Und zwar erzeugen wir zunächst den öffentlichen Schlüssel

Die erste Zahl ist 77, nämlich das Ergebnis der Multiplikation der beiden Primzahlen, 7 und 11. 77 dient uns im weiteren Verlauf als Modulus zur Ver- und Entschlüsselung.

Die zweite Zahl ist der öffentliche Exponent. Wir wählen hier 13.

Die dritte Zahl ist der geheime Schlüssel. Sie wird in einem komplizierten Verfahren errechnet, welches wir jetzt erklären:

Zunächst ziehen wir von unseren Primzahlen 7 und 11 jeweils die Zahl 1 ab (also $7 - 1$ und $11 - 1$) und multiplizieren die beiden resultierenden Zahlen miteinander. In unserem Beispiel ergibt das 60: $(7 - 1) * (11 - 1) = 60$. 60 ist unsere Modulzahl für die weiterführende Berechnung des geheimen Schlüssels (sie ist aber nicht mit dem eigentlichen Modulus 77 zu verwechseln).

Wir suchen jetzt eine Zahl, die multipliziert mit dem öffentlichen Schlüssel die Zahl 1 ergibt, wenn man mit dem Modul 60 rechnet:

$$13 \text{ mod } 60 * ? \text{ mod } 60 = 1 \text{ mod } 60$$

Die einzige Zahl, die diese Bedingung erfüllt, ist 37, denn

$$13 \text{ mod } 60 * 37 \text{ mod } 60 = 481 \text{ mod } 60 = 1 \text{ mod } 60$$

37 ist die einzige Zahl, die multipliziert mit 13 die Zahl 1 ergibt, wenn man mit dem Modul 60 rechnet.

Wir verschlüsseln mit dem öffentlichen Schlüssel eine Nachricht

Nun zerlegen wir die Nachricht in eine Folge von Zahlen zwischen 0 und 76, also 77 Zahlen, denn sowohl Verschlüsselung als auch Entschlüsselung verwenden den Modul 77 (das Produkt aus den Primzahlen 7 und 11).

Jede einzelne dieser Zahlen wird nun nach der Modulo-77 Arithmetik 13 mal mit sich selbst multipliziert. Sie erinnern sich: die 13 ist ja unser öffentlicher Schlüssel.

Nehmen wir ein Beispiel mit der Zahl 2: sie wird in die Zahl 30 umgewandelt, weil $2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 8192 = 30 \pmod{77}$ sind.

Ein weiteres Beispiel: 75 wird in die Zahl 47 umgewandelt, denn 75 wird 13 mal mit sich selbst multipliziert und durch 77 geteilt, so dass der Rest 47 entsteht.

Wenn man eine solche Rechnung für alle Zahlen zwischen 0 und 76 durchführt und die Ergebnisse in eine Tabelle einsetzt, sieht diese so aus:

	0	1	2	3	4	5	6	7	8	9
0	0	1	30	38	53	26	62	35	50	58
10	10	11	12	41	49	64	37	73	46	61
20	69	21	22	23	52	60	75	48	7	57
30	72	3	32	33	34	63	71	9	59	18
40	68	6	14	43	44	45	74	5	20	70
50	29	2	17	25	54	55	56	8	16	31
60	4	40	13	28	36	65	66	67	19	27
70	42	15	51	24	39	47	76			

Tabelle 25.1.:

In der linken Spalte stehen die 10er-Stellen, in der oberen Zeile die 1er-Stellen.

Wir entschlüsseln eine Nachricht mit dem privaten Schlüssel

Um das Beispiel mit der 2 von oben umzukehren, also die Nachricht zu dekodieren, multiplizieren wir 30 (die umgewandelte 2) unter Verwendung der Modulzahl 77 37 mal mit sich selbst. Sie erinnern sich: 37 ist der geheime Schlüssel.

Diese wiederholte Multiplikation ergibt eine Zahl die $2 \pmod{77}$ ergibt. Das andere Beispiel: die Zahl $47 \pmod{77}$ wird zur Zahl $75 \pmod{77}$ dekodiert.

Tabelle 25.2 zeigt die genaue Zuordnung der 77 Zahlen zwischen 0 und 76.

	0	1	2	3	4	5	6	7	8	9
0	0	1	51	31	60	47	41	28	57	37
10	10	11	12	62	42	71	58	52	39	68
20	48	21	22	23	73	53	5	69	63	50
30	2	59	32	33	34	7	64	16	3	74
40	61	13	70	43	44	45	18	75	27	14
50	8	72	24	4	54	55	56	29	9	38
60	25	19	6	35	15	65	66	67	40	20
70	49	36	30	17	46	26	76			

Tabelle 25.2.: Zahlentransformation modulo77, unter Verwendung des geheimen Schlüssels 37

Um eine Zahl mit Tabelle 25.2 zu transformieren, gehen wir nach der gleichen Methode vor wie bei Tabelle 25.1. Ein Beispiel: 60 wird transformiert in die Zahl in Zeile 60 und Spalte 0. Also wird 60 zu 25 transformiert.

Das überrascht nicht, denn wenn wir davon ausgehen, dass wir bei der Umwandlung von 25 mit Hilfe von Tabelle 25.1 als Ergebnis 60 erhalten, dann sollten wir auch bei der Transformation von 60 mit Hilfe von Tabelle 25.2 zum Ergebnis 25 gelangen. Dabei haben wir den öffentlichen Schlüssel, 13, zur Umwandlung bzw. Kodierung einer Zahl verwendet, und den geheimen Schlüssel 37, um sie zurückzuwandeln bzw. zu dekodieren. Sowohl für die Verschlüsselung als auch für die Entschlüsselung haben wir uns der Modulo-77 Arithmetik bedient.

Zusammenfassung

Wir haben. . .

- durch den Computer zwei zufällige Primzahlen erzeugen lassen;
- daraus das Produkt und den öffentlichen und den geheimen Subkey gebildet;
- gezeigt, wie man mit dem öffentlichen Schlüssel Nachrichten verschlüsselt;
- gezeigt, wie man mit dem geheimen Schlüssel Nachrichten entschlüsselt.

Diese beiden Primzahlen können so groß gewählt werden, dass es unmöglich ist, sie einzig aus dem öffentlich bekannt gemachten Produkt zu ermitteln. Das begründet die Sicherheit des RSA Algorithmus.

Wir haben gesehen, dass die Rechnerei sogar in diesem einfachen Beispiel recht kompliziert geworden ist. In diesem Fall hat die Person, die den Schlüssel öffentlich gemacht hat, die Zahlen 77 und 13 als öffentlichen Schlüssel bekanntgegeben. Damit kann jedermann dieser Person mit der oben beschriebenen Methode – wie im Beispiel der Tabelle 25.1 – eine verschlüsselte Zahl oder Zahlenfolge schicken. Der rechtmäßige Empfänger der verschlüsselten Zahlenfolge kann diese dann mit Hilfe der Zahl 77 und dem geheimen Schlüssel 37 dekodieren.

In diesem einfachen Beispiel ist die Verschlüsselung natürlich nicht sonderlich sicher. Es ist klar, dass 77 das Produkt aus 7 und 11 ist.

Folglich kann man den Code in diesem einfachen Beispiel leicht knacken. Der scharfsinnige Leser wird auch bemerkt haben, dass etliche Zahlen, zum Beispiel die Zahl 11 und ihr Vielfaches (also 22, 33 etc.) und die benachbarten Zahlen sich in sich selbst umwandeln.

	0	1	2	3	4	5	6	7	8	9
0	0	1	51	31	60	47	41	28	57	37
10	10	11	12	62	42	71	58	52	39	68
20	48	21	22	23	73	53	5	69	63	50
30	2	59	32	33	34	7	64	16	3	74
40	61	13	70	43	44	45	18	75	27	14
50	8	72	24	4	54	55	56	29	9	38
60	25	19	6	35	15	65	66	67	40	20
70	49	36	30	17	46	26	76			

Tabelle 25.3.:

Das erscheint als ein weiterer Schwachpunkt dieser Verschlüsselungsmethode: man könnte annehmen, dass die Sicherheit des Algorithmus dadurch beeinträchtigt würde. Doch stellen Sie sich nun vor, das Produkt zweier grosser Primzahlen, die auf absolut willkürliche Art und Weise gewählt werden, ergäbe

114,381,625,757,888,867,669,235,779,976,146,612,010,
218,296,721,242,362,562,561,842,935,706,935,245,733,
897,830,597,123,563,958,705,058,989,075,147,599,290,
026,879,543,541

Hier ist überhaupt nicht mehr ersichtlich, welche die beiden zugrunde liegenden Primzahlen sind. Folglich ist es sehr schwierig, aufgrund des öffentlichen Schlüssels den geheimen Schlüssel zu ermitteln. Selbst den schnellsten Computern der Welt würde es gewaltige Probleme bereiten, die beiden Primzahlen zu errechnen.

Man muss die Primzahlen also nur groß genug wählen, damit ihre Berechnung aus dem Produkt so lange dauert, dass alle bekannten Methoden daran in der Praxis scheitern. Außerdem nimmt der Anteil der Zahlen, die in sich selbst transformiert werden – wie wir sie oben in den Tabellen 25.1 und 25.2 gefunden haben – stetig ab, je größer die Primzahlen werden. Von Primzahlen in der Grössenordnung, die wir in der Praxis bei der Verschlüsselung verwenden, ist dieser Teil so klein, dass der RSA Algorithmus davon in keiner Weise beeinträchtigt wird.

Je größer die Primzahlen, desto sicherer die Verschlüsselung. Trotzdem kann ein normaler PC ohne weiteres das Produkt aus den beiden großem Primzahlen bilden. Kein Rechner der Welt dagegen kann aus diesem Produkt wieder die ursprünglichen Primzahlen herausrechnen – jedenfalls nicht in vertretbarer Zeit.

25.4. Die Darstellung mit verschiedenen Basiszahlen

Um zu verstehen, wie Nachrichten verschlüsselt werden, sollte man wissen, wie ein Computer Zahlen speichert und vor allem, wie sie in unterschiedlichen Zahlenbasen dargestellt werden können.

Dazu machen wir uns zunächst mit den Zahlenpotenzen vertraut.

Zwei hoch eins, das man als 2^1 darstellt, ist gleich 2; zwei hoch drei, dargestellt als 2^3 , ist $2 * 2 * 2 = 8$; zwei hoch zehn, dargestellt als 2^{10} , ist $2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 1024$.

Jede Zahl hoch 0 ist gleich 1, zum Beispiel $2^0 = 1$ und $5^0 = 1$. Verallgemeinert bedeutet dies, dass eine potenzierte Zahl so oft mit sich selbst multipliziert wird, wie es die Hochzahl (Potenz) angibt.

Das Konzept einer Zahlenbasis veranschaulicht zum Beispiel ein Kilometerzähler im Auto: das rechte Rad zählt nach jedem Kilometer eine Stelle weiter und zwar nach der vertrauten Abfolge der Zahlen 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, ...

und so weiter. Jedesmal, wenn das rechte Rad wieder 0 erreicht, zählt das Rad links davon eine Stelle hoch. Und jedesmal, wenn dieses zweite Rad die 0 erreicht, erhöht das Rad links davon um eins . . . und so weiter.



Das rechte Rad zählt die einzelnen Kilometer. Wenn es eine 8 angezeigt, dann sind dies 8 Kilometer. Das Rad links davon zeigt jeweils die vollen zehn Kilometer an: eine 5 bedeutet 50 Kilometer. Dann folgen die Hunderter: steht dort 7, dann bedeutet dies 700 Kilometer.

Nach dem gleichen Prinzip stellen wir ja auch unsere normale Zahlen mit den Ziffern 0 bis 9 dar.

„578“, zum Beispiel, bedeutet $5 * 100 + 7 * 10 + 8$, und dies entspricht 578.

Hier haben wir die „5“ stellvertretend für fünfhundert, „7“ für siebenzig und „8“ für acht. In diesem Fall ist die Basis 10, eine für uns vertraute Basis.

Also steht die rechte Ziffer für die Einer der betreffenden Zahl (d.h. sie wird mit 1 multipliziert), die Ziffer links davon steht für die Zehner (d.h. wird mit 10 multipliziert), die nächste Ziffer wiederum für die Hunderter (d.h. sie wird mit 100 multipliziert) und so weiter. Da wir Zahlen normalerweise zur Basis 10 darstellen, machen wir uns nicht die Mühe, die Basis extra anzugeben. Formal würde man dies bei der Zahl 55 mit der Schreibweise 55_{10} anzeigen, wobei die tiefgestellte Zahl die Basis anzeigt.

Wenn wir nicht zur Basis 10 darstellen, so müssen wir dies mit Hilfe einer solchen tiefgestellten Basiszahl anzeigen.

Angenommen, die Anzeige des Kilometerzählers hätte statt der Ziffern 0 bis 9 nur noch 0 bis 7. Das rechte Rädchen würde nach jedem Kilometer um eine Ziffer höher zählen, wobei die Zahlenfolge so aussehen würde:

0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, *undsoweiter*.

Unser Tacho zur Basis 8 stellt zum Beispiel folgende Zahl dar:

356

Die 6 auf dem rechten Rädchen zählt einzelne Kilometer, also 6 Kilometer.

Die 5 auf dem Rädchen daneben für $5 * 8$, also 40 Kilometer.

Die 3 links steht für je 64 Kilometer pro Umdrehung, also hier $3 * 8 * 8$ Kilometer.

So rechnet man also mit Zahlen zur Basis 8. Ein Beispiel: 728 bedeutet $7 * 8 + 2$, und das ist gleich „58“. Bei dieser Art der Darstellung steht die „2“ aus der 72 für 2, aber die „7“ steht für $7 * 8$.

Größere Zahlen werden schrittweise genauso aufgebaut, so dass 453_8 eigentlich $4 * 64 + 5 * 8 + 3$ bedeutet, was 299 ergibt.

Bei 453_8 steht die „3“ für 3, die „5“ für $5 * 8$ und die „4“ für $4 * 64$, wobei sich die „64“ wiederum aus $8 * 8$ herleitet.

Im angeführten Beispiel werden die Ziffern, von rechts nach links gehend, mit aufsteigenden Potenzen von 8 multipliziert. Die rechte Ziffer wird mit 8^0 (das ist 1) multipliziert, die links daneben mit 8^1 (das ist 8), die nächste links davon mit 8^2 (das ist 64) und so weiter.

Wenn man Zahlen zur Basis 10 darstellt, gibt es keine höhere Ziffer als 9 (also 10 minus 1). Wir verfügen also über keine Ziffer, die 10 oder eine größere Zahl darstellt. Um 10 darzustellen, brauchen wir zwei Ziffern, mit denen wir dann die „10“ schreiben können.

Wir haben also nur die Ziffern 0 bis 9.

So ähnlich ist es, wenn wir mit der Basiszahl 8 rechnen: dann haben wir nur die Ziffern 0 bis 7. Wollen wir zu dieser Basis eine höhere Zahl als sieben darstellen, müssen wir wieder zwei Ziffern verwenden. Zum Beispiel „9“ schreibt man als 11_8 , „73“ schreibt man als 111_8 .

Computer speichern Zahlen als eine Folge von Nullen und Einsen. Man nennt dies Binärsystem oder Rechnen mit der Basiszahl 2, weil wir nur die Ziffern 0 und 1 verwenden. Stellen Sie sich vor, wir würden die Kilometer mit einem Tachometer zählen, auf dessen Rädchen sich nur zwei Ziffern befinden: 0 und 1. Die Zahl 10101_2 zum Beispiel bedeutet im Binärsystem

$$1 * 16 + 0 * 8 + 1 * 4 + 0 * 2 + 1 = 21$$

.

In der Computerei verwendet man auch Gruppen von acht Binärziffern, das wohlbekannte Byte. Ein Byte kann Werte zwischen 0 - dargestellt als Byte 0000000_2 - und 255 - dargestellt als Byte 1111111_2 - annehmen. Ein Byte stellt also Zahlen zur Basis 256 dar.

Zwei weitere Beispiele:

$$10101010_2 = 170$$

und

$$00000101_2 = 5$$

.

Da der Computer die Buchstaben, Ziffern und Satzzeichen als Bytes speichert, schauen wir uns an, welche Rolle dabei die Darstellung zur Basis 256 spielt.

Nehmen wir die Silbe „un“. Das „u“ wird im Computer als 117 gespeichert und das „n“ als 110.

Diese Zahlenwerte sind für alle Computer standardisiert und werden ASCII-Code genannt. Um alle Zahlen und Symbole darstellen zu können, benötigen wir auf dem Computer die 256 Zahlen von 0 bis 255.

Wir können also die Silbe „un“ durch die Zahl $117 * 256 + 110$ darstellen.

Entsprechend würde man die Buchstabenfolge „und“ mit der Zahl $117 * 65536 + 110 * 256 + 100$ darstellen, denn das „d“ wird durch 100 repräsentiert.

Wir haben hier also Zahlen und Symbole, die auf der Computertastatur als normale Zahlen zur Basis 10 stehen, intern durch Zahlen zur Basis 256 repräsentiert.

Entsprechend können wir aus jeder Nachricht eine große Zahl machen. Aus einer langen Nachricht wird also eine gewaltig große Zahl. Und diese sehr große Zahl wollen wir nun nach dem RSA Algorithmus verschlüsseln.

Wir dürfen allerdings dabei die Zahl, zu der die Nachricht verschlüsselt wird, nicht größer werden lassen als das Produkt der Primzahlen (Modulus). Ansonsten bekommen wir Probleme, wie wir gleich noch sehen werden.

Da die folgende Prozedur mehrere Schritte umfaßt, fassen wir sie zunächst zusammen und verfolgen dann die Einzelschritte:

1. Die Nachricht *aba, cad, ada* wandeln wir – wie gesehen – in Zahlen um.
2. Diese Darstellung zur Basis 4 wandeln wir in eine Darstellung zur Basis 10 um, damit wir zur Verschlüsselung die Tabelle 25.1 benutzen können, in denen die Zahlen ja auch auf 10er-Basis dargestellt werden. Dabei entsteht eine kodierte Nachricht zur Basis 10.
3. Um die Kodierung im Vergleich zum „Klartext“ zu erkennen, rechnen wir die zur Basis 10 kodierte Nachricht auf die Basis 4 zurück und wandeln sie dann wieder in eine Buchstabensequenz.
4. So entsteht aus der Nachricht *aba, cad, ada* die verschlüsselte Nachricht *dbb, ddd, dac*.

Und nun ausführlich:

1. Die Nachricht *aba, cad, ada* wandeln wir – wie gesehen – in Zahlen um.

Angenommen, wir beschränken uns bei den Nachrichten auf die 4 Buchstaben a, b, c und d. In diesem – wirklich sehr einfachen – Beispiel können wir die vier Buchstaben durch die Zahlenwerte 0, 1, 2 und 3 darstellen, und haben dann

$$a = 0, b = 1, c = 2 \text{ und } d = 3$$

Wir wollen nun die Nachricht „abacadaca“ verschlüsseln. Wir kodieren diese Nachricht mit Hilfe der Primzahlen 7 und 11, mit dem öffentlichen Schlüssel 77 und 13 und dem dazugehörigen geheimen Schlüssel 37. Dieses Beispiel kennen wir bereits aus dem früheren Kapitel: wir haben damit die Tabellen 25.1 und 25.2 konstruiert.

2. Diese Darstellung zur Basis 4 wandeln wir in eine Darstellung zur Basis 10 um, damit wir zur Verschlüsselung die Tabelle 25.1 benutzen können, in denen die Zahlen ja auch auf 10er-Basis dargestellt werden.

Weil wir vier Buchstaben für die Nachricht verwenden, rechnen wir zur Basis 4. Für die Rechnung modulo 77 müssen wir die Nachricht in Stücke von je drei Zeichen Länge zerlegen, weil die größte dreiziffrige Zahl zur Basis 4 die 333_4 ist. Zur Basis 10 hat diese Zahl den Wert 63.

Würden wir stattdessen die Nachricht in vier Zeichen lange Stücke zerlegen, würde die Zahl zur Basis 4 den Wert 76 übersteigen und es würden unerwünschte Doppeldeutigkeiten entstehen. Folglich würde die Nachricht in dreiziffrigen Stücken nun

$$aba, cad, aca$$

ergeben. Geben wir den Zeichen nun ihre Zahlenwerte und vergessen dabei nicht, dass die Stücke dreiziffrige Zahlen zur Basis 4 darstellen.

Da wir die Buchstaben durch die Zahlen $a = 0, b = 1, c = 2, d = 3$ darstellen, wird die Nachricht zu:

$$010_4, 203_4, 020_4$$

Zur Basis 10 wird diese Nachricht durch die Zahlenfolge 4, 35, 8 dargestellt. Warum? Nehmen wir zum Beispiel das mittlere Stück 203_4 :

$$\begin{array}{lll} 3 * 4^0, & \text{also } 3 * 1, & \text{also } 3 \\ 0 * 4^1, & \text{also } 0 * 4, & \text{also } 0 \\ 2 * 4^2, & \text{also } 2 * 16, & \text{also } 32 \end{array}$$

3. Jetzt können wir zur Verschlüsselung die Tabelle 25.1 benutzen, die ja zur Basis 10 berechnet wurde. Diese Tabelle benutzen wir, weil wir mit dem schon bekannten Schlüsselpaar arbeiten wollen. Dabei entsteht eine kodierte Nachricht zur Basis 10.

Zum Verschlüsseln der Nachricht nehmen wir jetzt Tabelle 25.1 zur Hilfe. Die Nachricht wird nun zu der Zahlenfolge 53, 63, 50 (zur Basis 10).

	0	1	2	3	4	5	6	7	8	9
0	0	1	30	38	53	26	62	35	50	58
10	10	11	12	41	49	64	37	73	46	61
20	69	21	22	23	52	60	75	48	7	57
30	72	3	32	33	34	63	71	9	59	18
40	68	6	14	43	44	45	74	5	20	70
50	29	2	17	25	54	55	56	8	16	31
60	4	40	13	28	36	65	66	67	19	27
70	42	15	51	24	39	47	76			

4. Wiederum zur Basis 4 konvertiert, entsteht die verschlüsselte Nachricht.

Wird sie nun wieder zur Basis 4 konvertiert, ergibt die Nachricht nun $311_4, 333_4, 302_4$. Konvertiert man diese zu einer Buchstabensequenz, erhält man dbb, ddd, dac, was sich nun erheblich von der ursprünglichen Nachricht unterscheidet.

Man kehrt nun also den Prozeß um und transformiert die Zahlenfolge 53, 63, 50 mit Tabelle 25.2 und erhält die Sequenz 4, 35, 8. Und das entspricht, als Zahlenfolge genau der ursprünglichen Nachricht.

Anhand der Tabellen 25.1 und 25.2 können wir ebensogut Nachrichten unter Verwendung des geheimen Schlüssels (d.h. erst Tabelle 25.2 benutzen) verschlüsseln, dann mit dem öffentlichen Schlüssel (d.h. Tabelle 25.1 als zweites benutzen) dekodieren und damit unsere ursprüngliche Zahl wieder herstellen. Das bedeutet, dass der Inhaber des geheimen Schlüssels damit Nachrichten unter Verwendung des RSA Algorithmus verschlüsseln kann. Damit ist bewiesen, dass sie eindeutig nur von ihm stammen können.

Fazit:

Wie Sie gesehen haben, ist die ganze Angelegenheit zwar im Detail kompliziert, im Prinzip aber durchaus nachvollziehbar. Sie sollen schließlich nicht einer Methode einfach nur vertrauen, sondern – zumindest ansatzweise – ihre Funktionsweise durchschauen. Sehr viele tiefergehende Details sind leicht in anderen Büchern (z.B.: R. Wobst, „Abenteuer Kryptologie“) oder im Internet zu finden.

Immerhin wissen Sie nun: Wenn jemand sich an Ihren verschlüsselten E-Mails zu schaffen macht, ist er durchaus so lange damit beschäftigt, dass er dann keine Lust mehr haben dürfte diese auch noch zu lesen. . .

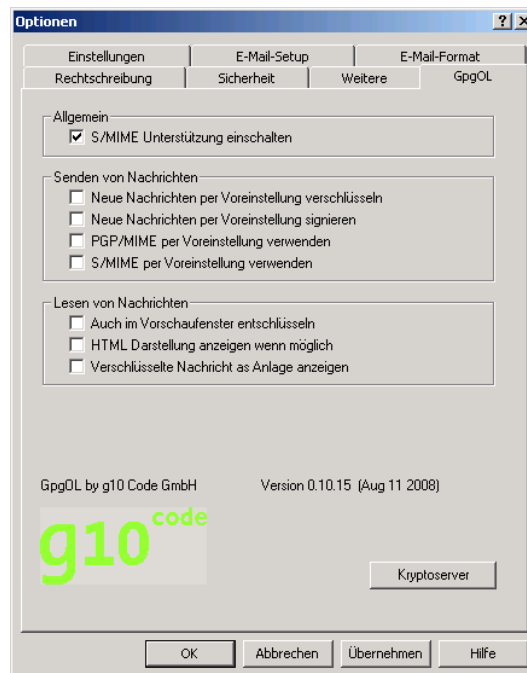
Teil III.
Anhang

A. Hinweise zum Outlook Plugin *GpgOL*

GpgOL ist ein Plugin für Microsoft Outlook, es integriert dort die Bedienung von GnuPG.

Da Outlook ein proprietäres Produkt, also nicht als Freie Software mit Quelltext verfügbar ist, hat die Integration eine Reihe von „Ecken und Kanten“. Oder mit anderen Worten: Die Bedienung ist nicht so komfortabel wie es beispielsweise die Benutzer des E-Mail Programmes Kontakt gewöhnt sind.

Die Installation wird durch den Gpg4win Installer vorgenommen. Beim nächsten Start von Outlook findet sich im Menü *Extras*→*Optionen* eine Karteikarte *GpgOL*:



Die Karteikarte *GpgOL* unterteilt sich in drei Kästen:

1. Allgemein:

Nach der Installation von Gpg4win ist die S/MIME Funktionalität in GpgOL deaktiviert. Damit ist die S/MIME Unterstützung von GnuPG gemeint. Outlook selbst unterstützt ebenfalls X.509 und S/MIME, arbeitet aber natürlich nicht mit den Gpg4win-Komponenten zusammen. Konkret heisst das, dass alle Einstellungen, das Schlüsselmanagement und die Benutzerdialoge unterschiedlich sind. Es ist zu beachten, dass Outlook keine OpenPGP Unterstützung anbietet.

S/MIME

Wichtig: Wenn Sie S/MIME nutzen möchten, müssen Sie zuvor die Option *S/MIME Unterstützung einschalten* aktivieren.

2. Senden von Nachrichten:

Die beiden ersten Optionen in diesem Kasten steuern, ob per Voreinstellung neue Nachrichten verschlüsselt und/oder signiert werden sollen. Sie können dies aber immer noch bei der Erstellung einer Nachricht individuell verändern. Lediglich die Schaltflächen sind schon entsprechend aktiviert.

Die beiden letzten Optionen definieren, ob PGP/MIME *oder* S/MIME per Voreinstellung verwendet werden soll. Auch hier können Sie diese Entscheidung immer noch vor dem Senden jeder Nachricht manuell verändern.

3. Lesen von Nachrichten:

Auch im Vorschaufenster entschlüsseln

Soll im Vorschaufenster die entschlüsselte Fassung erscheinen, so ist diese Option einzuschalten. Sie sollten dabei bedenken, dass dadurch bereits beim Durchblättern durch Ihre Nachrichten die Entschlüsselungs- und Prüfroutinen ausgeführt werden. Das heisst, es werden Dialog zum Status der E-Mails angezeigt und ggf. werden Sie nach einer Passphrase zur Entschlüsselung gefragt.

HTML Darstellung anzeigen wenn möglich

Diese Option kann benutzt werden, um die HTML Version einer Nachricht anzuzeigen. Im Normalfall oder falls keine HTML Version vorhanden ist, wird die Text Version dargestellt.

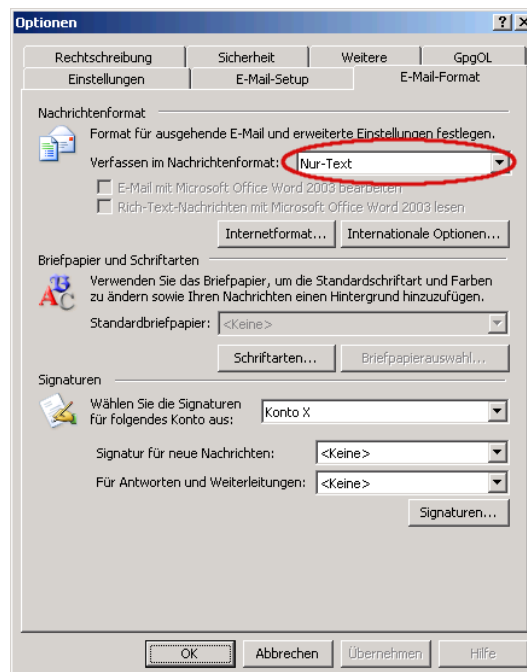
Verschlüsselte Nachricht als Anlage anzeigen

Der verschlüsselte Teil der Nachricht wird zusätzlich als Anhang angezeigt. Der Anwender kann so den verschlüsselten Teil separat speichern oder auf andere Weise weiterverarbeiten.

Alle Optionen sind nach einer Neuinstallation bereits sinnvoll vorbelegt.

Um verschlüsselte Nachrichten mit Outlook versenden zu können, sollten Sie sicherstellen, daß Sie **nicht Microsoft Word** zum Verfassen der Nachrichten benutzen.

Desweiteren ist dringend anzuraten auf HTML Nachrichten zu verzichten. Bitte kontrollieren Sie dies im Menüpunkt *Extras*→*Optionen* auf der Karteikarte *E-Mail-Format*. Das Nachrichtenformat sollte auf „**Nur-Text**“ eingestellt sein (siehe rot markierter Bereich):



B. GnuPG mit anderen E-Mail-Programme nutzen

Das Gpg4win Kompendium geht vor allem auf das E-Mail-Programm Outlook ein. GnuPG ist jedoch mit allen anderen E-Mail Programmen auch verwendbar. Unterschiede gibt es im Komfort: je besser GnuPG für ein E-Mail-Programm integriert ist, desto einfacher die Verwendung.

Die primitivste Methode, also wenn ein E-Mail-Programm überhaupt nichts über GnuPG weiss, ist die Verschlüsselung via Zwischenablage mit Hilfe von Kleopatra. Dies funktioniert nur für OpenPGP, für S/MIME und kompliziertere PGP/MIME E-Mails werden Sie über eine Zwischenspeicherung als Datei gehen müssen. Beide Methoden werden im ersten Teil dieses Kompendiums beschrieben.

Ein Integration für GnuPG wird derzeit für folgende Windows-Mailprogramme angeboten:

Thunderbird mit Plugin **Enigmail**¹.

Outlook ab Version 2003 mit Plugin **GpgOL**. Dies ist in Gpg4win fest integriert.

Claws Mail: Dieses E-Mail-Programm wird komplett mit Gpg4win mitgeliefert und kann optional mitinstalliert werden. Diese Installation über Gpg4win konfiguriert bereits Plugins für die Verwendung von PGP/MIME und S/MIME. Diese Plugins verwenden jedoch nicht Kleopatra und bieten daher nicht denselben Komfort wie es derzeit das Outlook Plugin bietet.

Kontakt: Die komfortabelste und erprobteste Integration von GnuPG bietet dieses E-Mail-Programm. Es ist für jedes System auf dem die KDE-Umgebung² installiert ist verfügbar, also für nahezu jedes GNU/Linux System.

Eine Windows-Version von Kontakt steht derzeit bereits als erste Test-Version zur Verfügung.

Desweiteren verfügen praktisch alle Mailprogramme, die unter GNU/Linux oder anderen Unix Varianten laufen, über eine komfortable und integrierte GnuPG-Unterstützung. Hervorzuheben ist hier das Programm „Kontakt“ der KDE-Umgebung. Dort wird eine umfangreiche und sehr komfortable Integration angeboten.

¹http://www.thunderbird-mail.de/wiki/Enigmail_OpenPGP

²<http://www.kde.de>

C. Automatische Installation von Gpg4win

In diesem Kapitel wird die automatisierte Installation (ohne Benutzerdialoge) erläutert.

In einigen Fällen, wie zum Beispiel für Software-Verteilungssysteme, ist es hilfreich, wenn die Installation von Gpg4win ohne die Interaktion über Dialoge funktioniert. Um aber trotzdem vorab alle Installationseinstellungen bestimmen zu können, unterstützt der Gpg4win Installer eine Option für den Standard-Installationspfad auf der Kommandozeile als auch eine Steuerungsdatei.

Der Standard-Installationspfad kann mit der Option `/D=<PFAD>` angegeben werden, welche als letzte Option auf der Kommandozeile erscheinen muß. Der Installerdateiname (hier: `gpg4win.exe`) kann je nach Version variieren. Die Groß-/Kleinschreibung bei der Eingabe der Optionsschalter in der Kommandozeile ist hierbei wichtig. Eventuell sind noch vorher entsprechende Schreib-/Leserechte zu setzen.

Ein Beispiel:

```
> gpg4win.exe /D=D:\Programme\Gpg4win
```

Mit der Option `/S` läuft die Installation „still“ (also automatisch ohne grafische Oberfläche) ab. Ohne Angabe von weiteren Parametern, werden alle Voreinstellungen des Installers übernommen.

Der Gpg4win-Installer unterstützt auch eine sogenannte Steuerungsdatei. Mit der Option `/C=INIFILE` kann eine Steuerungsdatei (Name endet üblicherweise auf `.ini`) angegeben werden.

Ein weiteres Beispiel:

```
> gpg4win.exe /S /C=C:\TEMP\gpg4win.ini
```

Diese `.ini` Datei sollte genau einen Abschnitt `[gpg4win]` enthalten. Dort können diverse Einstellungen vorgenommen werden, darunter absolute Pfade für die zu installierenden Konfigurationsdateien. Relative Pfade, also abhängig vom aktuellem Arbeitsverzeichnis, dürfen hier nicht angegeben werden. Absolute Pfade enthalten den vollständigen Pfad inklusive der Laufwerksangabe. In der Regel sind die Einstellungen dann anzugeben falls nicht die Voreinstellung verwendet werden soll. Ausnahmen davon sind im Beispiel auf der nächsten Seite dokumentiert.

Hier ist ein Beispiel für eine Steuerungsdatei, die zugleich alle erlaubten Schlüsselworte zeigt:

```
[gpg4win]
; Installer Einstellungen. Weg- oder leerlassen für Voreinstellung
inst_gpgol = true
inst_gpgex = true
inst_kleopatra = true
inst_gpa = true
inst_claws_mail = false
inst_compendium_de = true
inst_man_novice_de = true
inst_man_novice_en = true
inst_man_advanced_de = true
inst_man_advanced_en = true

; Die Stellen, an denen Verknüpfungen erzeugt werden sollen.
inst_start_menu = true
inst_desktop = false
inst_quick_launch_bar = false

; Im Gegensatz zu den anderen Optionen überschreibt diese Option
; die Einstellung des Benutzers im Installer.
inst_start_menu_folder = GnuPG for Windows

; Standard-Konfigurationsdateien.
gpg.conf = D:\config\gpg-site.conf
gpg-agent.conf = D:\config\gpg-agent-site.conf
trustlist.txt = D:\config\trustlist-site.txt
dirmngr.conf = D:\config\dirmngr-site.conf
dirmngr_ldapserver.conf = D:\config\dirmngr_ldapserver-site.conf
scdaemon.conf = D:\config\scdaemon-site.txt
gpa.conf = D:\config\gpa-site.conf
```

Ein entsprechender Aufruf zur automatischen Installation mit einer Steuerungsdatei `gpg4win.ini` und einem Installationspfad `D:\Programme\Gpg4win` könnte also wie folgt aussehen:

```
> gpg4win.exe /S /C=C:\TEMP\gpg4win.ini /D=D:\Programme\Gpg4win
```

D. Umstieg von anderen GnuPG Programmen

Wir werden hier erläutern, wie Sie von anderen GnuPG basierten Programmen auf Gpg4win umsteigen können. Das Installationsprogramm erkennt einige dieser Programme und warnt Sie in diesem Fall.

Generell ist es ratsam, eine vorhandene Installation eines anderen GnuPG basierten Programms zu entfernen, bevor Gpg4win installiert wird. Es ist hier wichtig, die vorhandenen Schlüssel vorher zu sichern.

Der einzige sinnvolle Weg dies zu tun, ist unter Verwendung der im alten System vorhandenen Möglichkeiten. Suchen Sie nach einem Menüpunkt um die eigenen privaten (geheimen) Schlüssel zu sichern als auch nach einem Menüpunkt um alle vorhandenen öffentlichen Schlüssel und Zertifikate zu sichern. Sichern Sie diese dann in eine oder mehrere Dateien.

Sobald Sie Gpg4win installiert haben, prüfen Sie, ob Ihre alten Schlüssel bereits vorhanden sind. Sie können dies mit Kleopatra oder GPA machen. Sind die Schlüssel schon vorhanden, so entsprach das alte System bereits den neuen Konventionen zum Speicherort für die Schlüssel und Sie müssen nichts weiter unternehmen.

Wenn die alten Schlüssel nicht erscheinen, so importieren Sie diese einfach aus den erstellten Sicherungsdateien. Lesen Sie hierzu das Kapitel 19.

Falls das alte System GPA verwendet, so können Sie die dort vorhandene Backupmöglichkeit benutzen. Diese sollte sehr ähnlich zu der Funktion in der GPA Version aus Gpg4win sein.

Falls Sie keinen anderen Weg finden, Ihre alten Schlüssel wiederzufinden, so suchen Sie bitte mit den Bordmitteln von Windows nach Dateien mit den Namen `secring.gpg` und `pubring.gpg` und importieren diese beiden Dateien mittels Kleopatra¹.

¹Dies ist nicht der offizielle Weg, funktioniert aber noch mit allen aktuellen GnuPG Versionen.

E. Deinstallation von Gpg4win

Soll Gpg4win von Ihrem System entfernt, also deinstalliert werden, dann sollten Sie zunächst alle nicht notwendigen Anwendungen beenden und alle Schlüssel und Zertifikate sichern. Falls Sie auf Ihrem Rechner mit eingeschränkten Rechten arbeiten sollten, ist es für die Deinstallation außerdem notwendig mit **Administratorenrechten** angemeldet zu sein. Wurde die Installation bereits über Ihr Benutzerkonto durchgeführt, so verfügt es über Administratorenrechte.

Es gibt zwei Möglichkeiten die Deinstallation auszuführen:

- Einmal mit den Bordmitteln von Microsoft Windows:
Öffnen Sie *Start*→*Einstellungen*→*Systemsteuerung*→*Software* und wählen Sie dann *GnuPG for Windows* aus.
Mit dem Knopf [Entfernen] deinstallieren Sie *GnuPG for Windows* bzw. *Gpg4win* von Ihrem System.
- Die zweite Möglichkeit zur Deinstallation von Gpg4win bietet Ihnen die ausführbare Datei `gpg4win-uninstall.exe`. Sie wird mit Gpg4win mitgeliefert und liegt im Installationsverzeichnis (in der Regel `C:\Programme\GNU\GnuPG\`). Falls Sie bei der Installation einen anderen als den voreingestellten Pfad gewählt hatten, werden Sie das Deinstallationsprogramm an entsprechender Stelle finden.

In beiden Fällen werden alle allgemeinen Systemdaten von Gpg4win aus dem Installationsverzeichnis sowie die Verknüpfungen im Startmenü, Desktop und Schnellstartleiste entfernt.

Nicht gelöscht werden die benutzerspezifischen und systemweiten Anwendungsverzeichnisse mit den Konfigurationseinstellungen:

- Benutzerspezifische Anwendungsdaten (`%APPDATA%\gnupg`)
entspricht in der Regel dem Verzeichnis:
`C:\Dokumente und Einstellungen\<name>\Anwendungsdaten\gnupg\`
In diesem `gnupg`-Verzeichnis befinden sich sämtliche persönlichen GnuPG Daten, also die persönlichen Schlüssel, Zertifikate, Vertrauenseinstellungen und Programmkonfigurationen.
- systemweite Anwendungsdaten (`%COMMON_APPDATA%\GNU`)
entspricht in der Regel dem Verzeichnis: `C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\`

Es kann vorkommen, dass nach der Deinstallation noch einige Systemdaten (die zwischenzeitlich von Gpg4win erstellt wurden) zurückbleiben. Falls Sie sich sicher sind, dass diese Daten nicht gesichert werden sollen, können Sie das Installationsverzeichnis, also zum Beispiel `C:\Programme\GNU\GnuPG`, über den Dateimanager vollständig löschen.

F. History

- „GnuPP für Einsteiger“, 1. Auflage März 2002 und „GnuPP für Durchblicker“, Auflage März 2002,
Autoren: Manfred J. Heinze, TextLab text+media
Beratung: Lutz Zolondz, G-N-U GmbH
Illustrationen: Karl Bihlmeier, Bihlmeier & Kramer GbR
Layout: Isabel Kramer, Bihlmeier & Kramer GbR
Fachtext: Dr. Francis Wray, e-mediate Ltd.
Redaktion: Ute Bahn, TextLab text+media
Herausgegeben vom Bundesministerium für Wirtschaft und Technologie (BMWi).
Verfügbar unter <http://www.gnupp.de/pdf/einsteiger.pdf> und
<http://www.gnupp.de/pdf/durchblicker.pdf>.
- Revidierte nicht-veröffentlichte Version von TextLab text+media.
- „Gpg4win für Einsteiger“ und „Gpg4win für Durchblicker“, Dezember 2005
Autoren: Werner Koch, g10 Code GmbH
Herausgegeben durch das Gpg4win Projekt.
- Aufgrund einer Erlaubniss des BMWi vom 14. November 2007 wurde die Invariant Section „Impressum“ entfernt und an die aktuelle Version angepasst.
- Das „Gpg4win Kompendium 3.0“ fasst „Gpg4win für Einsteiger“ und „Gpg4win für Durchblicker“ zusammen und ist umfassend für Gpg4win 2.0 aktualisiert und ergänzt worden.
Diese wurde hauptsächlich durchgeführt von:
Emanuel Schütze, Intevation GmbH
Dr. Jan-Oliver Wagner, Intevation GmbH

G. GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s

overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.