



Das Gpg4win-Kompendium

Sichere E-Mail- und Datei-Verschlüsselung mit GnuPG für Windows

Basierend auf einer Fassung von

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

Grundlegend überarbeitet von

Werner Koch, Jochen Saalfeld, Florian v. Samson, Emanuel Schütze und Dr. Jan-Oliver Wagner.

Eine Veröffentlichung der Gpg4win-Initiative

Version 4.0.1 vom 3. April 2018

(zuletzt geringfügig korrigiert am 3. April 2018)

Impressum

Copyright © 2002 Bundesministerium für Wirtschaft und Technologie¹

Copyright © 2005 g10 Code GmbH

Copyright © 2009, 2010, 2017 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>.

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

Über dieses Kompendium

Das Gpg4win-Kompendium besteht aus drei Teilen:

- **Teil I „Für Einsteiger“:** Der Schnelleinstieg in Gpg4win.
- **Teil II „Für Fortgeschrittene“:** Das Hintergrundwissen zu Gpg4win.
- **Anhang:** Weiterführende technische Informationen zu Gpg4win.

Teil I „Für Einsteiger“ führt Sie kurz und knapp durch die Installation und die alltägliche Benutzung der Gpg4win-Programmkomponenten. Der Übungsroboter **Adele** wird Ihnen dabei behilflich sein und ermöglicht Ihnen, die E-Mail-Ver- und Entschlüsselung (mit OpenPGP) so lange zu üben, bis Sie sich vertraut im Umgang mit Gpg4win gemacht haben.

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.

Teil I und II können unabhängig voneinander benutzt werden. Zu Ihrem besseren Verständnis sollten Sie aber möglichst beide Teile in der angegebenen Reihenfolge lesen.

Im **Anhang** finden Sie Details zu spezifischen technischen Themen rund um Gpg4win, unter anderem zur Outlook-Programmerweiterung GpgOL.

Genau wie das Kryptografie-Programmpaket Gpg4win, wurde dieses Kompendium nicht für Mathematiker, Geheimdienstler und Kryptografen geschrieben, sondern **für jedermann**.

Das Programmpaket Gpg4win und das Gpg4win-Kompendium sind verfügbar unter:

<http://www.gpg4win.de>

Legende

In diesem Kompendium werden folgende Textauszeichnungen benutzt:

- *Kursiv* wird dann verwendet, wenn etwas auf dem Bildschirm erscheint (z.B. in Menüs oder Dialogen). Zum Kennzeichnen von [*Schaltflächen*] werden zusätzlich eckige Klammern benutzt.

Kursiv werden vereinzelt auch einzelne Wörter im Text gesetzt, wenn deren Bedeutung in einem Satz betont, das Schriftbild aber nicht durch die Auszeichnung **fett** gestört werden soll (z.B.: *nur* OpenPGP).
- **Fett** werden einzelne Wörter oder Sätze gesetzt, die besonders wichtig und damit hervorzuheben sind. Diese Auszeichnung unterstützt den Leser bei der schnelleren Erfassung hervorgehobener Schlüsselbegriffe und wichtiger Passagen.
- `Feste Laufweite` wird für alle Dateinamen, Pfadangaben, URLs, Quellcode sowie Ein- und Ausgaben (z.B. von Kommandozeilen) verwendet.

Im folgenden werden die Ausdrücke und Kennzeichnungen verwendet:

- Sie werden im folgenden immer wieder von „Schlüsseln“ und „Zertifikaten“ lesen. In der OpenPGP-Welt hat sich der Begriff „Schlüssel“ durchgesetzt. Für die Nutzung von S/MIME wird der Begriff „Zertifikat“ verwendet. In diesem Kompendium wird primär Schlüssel verwendet. Nur wenn es explizit um S/MIME geht, wird Zertifikat genutzt.

Die Software *Kleopatra* war einst ein reines Verwaltungsprogramm für S/MIME-Zertifikate. Erst nachträglich wurde es um die Verwaltung für OpenPGP-Schlüssel erweitert.

- Wenn in einem Kapitel explizit auf die Nutzung mit S/MIME eingegangen wird, wird darauf am Rand mit diesem Symbol hingewiesen:



Inhaltsverzeichnis

I. Für Einsteiger	8
1. Gpg4win – Kryptografie für alle	9
2. E-Mails verschlüsseln: weil der Briefumschlag fehlt	11
3. So funktioniert Gpg4win	14
4. Die Passphrase	25
5. Zwei Wege, ein Ziel: OpenPGP & S/MIME	28
6. Installation von Gpg4win	30
7. Erstellung eines Schlüsselpaars	36
7.1. OpenPGP-Schlüsselpaar erstellen	38
7.2. X.509-Zertifikat erstellen	45
8. Schnellstart mit Übungen für OpenPGP	49
8.1. Dateiverschlüsselung	49
8.2. E-Mail-Verschlüsselung	54
9. Öffentliche Schlüssel importieren	56
9.1. Importieren aus Datei	56
9.2. Importieren vom Schlüsselservers	57
10. Öffentliche Schlüssel prüfen	58
11. E-Mails signieren und verschlüsseln	64
11.1. E-Mails signieren und verschlüsseln mit GpgOL	65
11.1.1. Signatur prüfen mit GpgOL	68
11.2. E-Mails signieren	69
11.3. E-Mails verschlüsselt archivieren	70
12. Dateien signieren und verschlüsseln	72
12.1. Dateien signieren, verschlüsseln und prüfen	73
13. Öffentliche Schlüssel veröffentlichen	79
13.1. Veröffentlichen auf OpenPGP-Schlüsselservers	79
13.2. Veröffentlichen von X.509-Zertifikaten	79

II. Für Fortgeschrittene	80
14. Schlüssel im Detail	81
15. Die Schlüsselserver	83
15.1. Schlüsselserver einrichten	84
15.2. X.509 Schlüsselserver einrichten	86
15.3. Schlüssel auf Schlüsselservern suchen und importieren	86
16. Dateianhänge verschlüsseln	88
17. Im- und Export eines geheimen Schlüssels	89
17.1. Export	90
17.2. Import	91
17.3. Paperkey	93
17.3.1. Export mit Paperkey	93
17.3.2. Import mit Paperkey	93
18. Konfiguration von Smartcards	94
18.1. Nutzung von Smartcards mit OpenPGP	94
18.1.1. Erstellen des OpenPGP-Schlüssels auf der SmartCard	94
18.2. Nutzung von NetKey-Cards mit X.509	97
19. Systemweite Konfiguration und Vorbelegung für S/MIME	99
20. Bekannte Probleme und Abhilfen	101
20.1. GpgOL-Menüs und -Dialoge nicht mehr in Outlook zu finden	101
20.2. GpgOL-Schaltflächen sind in Outlook2003 nicht in der Symbolleiste	101
20.3. GpgOL-Schaltflächen sind in Outlook2007 unter „Add-Ins“	101
20.4. Fehler beim Start von GpgOL	102
20.5. Installation von Gpg4win auf einem virtuellen Laufwerk	102
20.6. GpgOL überprüft keine InlinePGP-E-Mails von „CryptoEx“	102
20.7. Keine S/MIME-Operationen möglich (Systemdienst „DirMngr“ läuft nicht)	103
20.8. Keine S/MIME-Operationen möglich (CRLs nicht verfügbar)	103
20.9. Keine S/MIME-Operationen möglich (Wurzelzertifikat nicht vertrauenswürdig)	104
21. Dateien und Einstellungen von Gpg4win	105
21.1. Persönliche Einstellungen der Anwender	105
21.2. Zwischengespeicherte Sperrlisten	105
21.3. Vertrauenswürdige Wurzelzertifikate von DirMngr	106
21.4. Weitere Zertifikate von DirMngr	106
21.5. Systemweite Konfiguration zur Verwendung externer X.509-Zertifikatsserver	107
21.6. Systemweite vertrauenswürdige Wurzelzertifikate	108
21.7. Vertrauenswürdigkeit der Wurzelzertifikate durch Benutzer markieren	109
22. Probleme in den Gpg4win-Programmen aufspüren (Logdateien)	110
22.1. Logdateien von Kleopatra einschalten	111
22.2. Logdatei von GpgOL einschalten	112

22.3. Logdatei von DirMngr einschalten	113
22.4. Logdatei von GnuPG einschalten	114
22.5. Logdatei von GpgME einschalten	115
23. Warum Gpg4win nicht zu knacken ist ...	116
24. GnuPG und das Geheimnis der großen Zahlen	117
24.1. Das Rechnen mit Restklassen	119
24.2. RSA-Algorithmus und Rechnen mit Restklassen	122
24.3. RSA-Verschlüsselung mit kleinen Zahlen	123
24.4. Die Darstellung mit verschiedenen Basiszahlen	128
 III. Anhang	 135
 A. Hinweise zur Outlook-Programmerweiterung GpgOL	 136
B. GnuPG mit anderen E-Mail-Programmen nutzen	138
C. Automatische Installation von Gpg4win	139
D. Umstieg von anderen Programmen	141
E. Deinstallation von Gpg4win	143
F. Historie	145
G. GNU Free Documentation License	146
Index	153

Teil I.

Für Einsteiger

1. Gpg4win – Kryptografie für alle

Was ist Gpg4win? Die deutsche Wikipedia beantwortet diese Frage so:

Gpg4win (GNU Privacy Guard for Windows) ist ein Installationspaket für Windows zur E-Mail- und Datei-Verschlüsselung. Gpg4win ermöglicht das einfache und kostenfreie Ver- und Entschlüsseln von E-Mails, Dateien und Datei-Ordern. Ebenso kann mittels digitaler Signaturen die Integrität und die Authentizität der verschlüsselten E-Mails und Dateien überprüft werden. Das Paket besteht aus verschiedenen Programmkomponenten und einem Handbuch.

Die Handbücher „Einsteiger“ und „Durchblicker“ wurden für die vorliegende zweite Version unter der Bezeichnung „Kompendium“ zusammengeführt. Gpg4win umfasst in Version 2 die folgenden Programme:

- **GnuPG**
GnuPG ist das Kernstück von Gpg4win – die eigentliche Verschlüsselungs-Software.
- **Kleopatra**
Die zentrale Zertifikatsverwaltung von Gpg4win, die für eine einheitliche Benutzerführung bei allen kryptografischen Operationen sorgt.
- **GNU Privacy Assistent (GPA)**
ist ein alternatives Programm zum Verwalten von Zertifikaten neben Kleopatra.
- **GnuPG für Outlook (GpgOL)**
ist eine Erweiterung für Microsoft Outlook 2003 und 2007, die verwendet wird, um Nachrichten zu signieren bzw. zu verschlüsseln.
- **GPG Explorer eXtension (GpgEX)**
ist eine Erweiterung für den Windows-Explorer, mit der man Dateien über das Kontextmenü signieren bzw. verschlüsseln kann.

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 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. Jeder kann und darf den Quellcode der Programme untersuchen und – sofern er das notwendige Fachwissen dazu hat – Änderungen daran durchführen und diese weitergeben.

Für eine Sicherheits-Software ist diese Transparenz – der garantierte Einblick in den Quellcode – eine unverzichtbare Grundlage. Nur so lässt sich die Vertrauenswürdigkeit der Programmierung und des Programmes wirklich prüfen.

¹Oft auch als Open Source Software (OSS) bezeichnet.

GnuPG basiert auf dem internationalen Standard **OpenPGP** (RFC 4880), ist vollständig kompatibel zu PGP und benutzt auch die gleiche Infrastruktur (Schlüsselserver etc.) wie dieser. Seit Version 2 von GnuPG wird auch der kryptografische Standard **S/MIME** (IETF RFC 3851, ITU-T X.509 und ISIS-MTT/Common PKI) unterstützt.

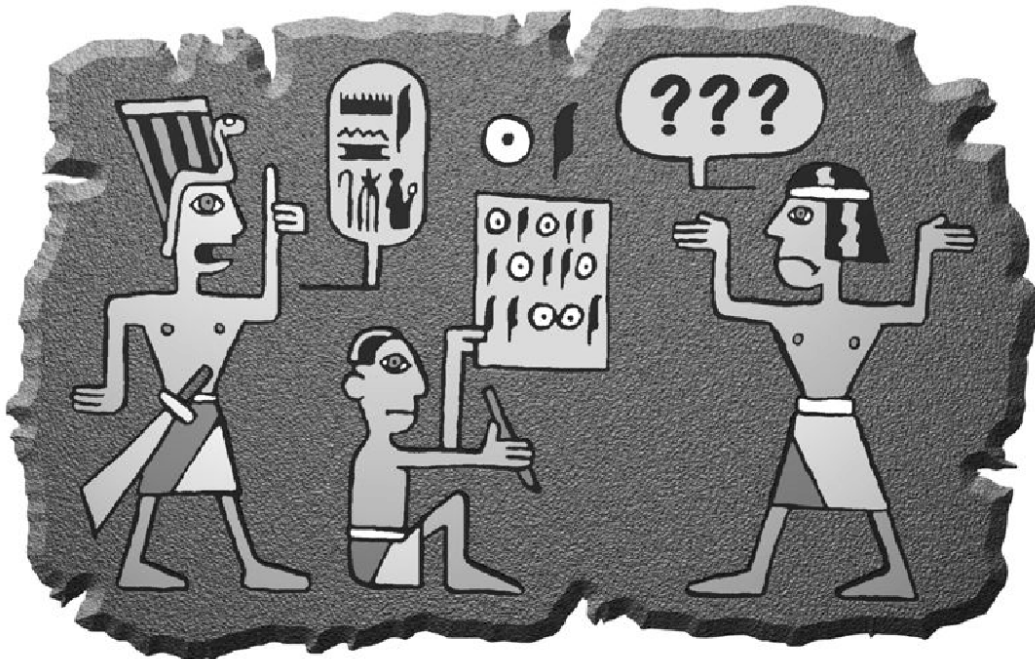
PGP („Pretty Good Privacy“) ist keine Freie Software, sie war lediglich vor vielen Jahren kurzzeitig zu ähnlichen Bedingungen wie GnuPG erhältlich. Diese Version entspricht aber schon lange nicht mehr dem Stand der Technik.

Die Vorläufer von Gpg4win wurden durch das Bundesministerium für Wirtschaft und Technologie im Rahmen der Aktion „Sicherheit im Internet“ unterstützt. Gpg4win und Gpg4win2 wurden durch das Bundesamt für Sicherheit in der Informationstechnik (BSI) unterstützt.

Weitere Informationen zu GnuPG und weiteren Projekten der Bundesregierung zum Schutz im Internet finden Sie auf den Webseiten www.bsi.bund.de und www.bsi-fuer-buerger.de des Bundesamtes für Sicherheit in der Informationstechnik.

2. E-Mails verschlüsseln: weil der Briefumschlag fehlt

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 nicht mehr nur für Könige, sondern 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, muss man sich in den neuen Technologien erst sichern. Das Internet ist so schnell und massiv über uns hereingebrochen, dass man mit der Wahrung unserer Rechte noch nicht so recht nachgekommen ist.

Beim altmodischen Briefschreiben schützen Sie die Inhalte von Mitteilungen ganz selbstverständlich mit einem Briefumschlag. Der Umschlag schützt die Nachrichten vor fremden Blicken, eine Manipulation am Umschlag kann man leicht bemerken. Nur wenn etwas nicht 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 bestimmen Sie selbst und niemand sonst.

Diese Entscheidungsfreiheit haben Sie bei E-Mails nicht. Eine normale E-Mail ist immer offen wie eine Postkarte, und der elektronische „Briefträger“ – und andere – können sie jederzeit 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 sie auch 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 es jedoch technisch möglich. Es gibt mehr als einen Hinweis darauf, dass dies genau heute schon im großen Stil mit E-Mail geschieht. Ein Artikel über das Echelon-System¹ liefert dazu interessantes Hintergrundwissen.

Denn: der Umschlag fehlt.



¹<http://www.heise.de/tp/r4/artikel/6/6928/1.html>



Was Ihnen hier vorgeschlagen wird, 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 des Programmpakets Gpg4win wahrnehmen. Sie müssen diese Software nicht benutzen – Sie müssen ja auch keinen Briefumschlag benutzen. Aber es ist Ihr gutes Recht.

Um dieses Recht zu sichern, bietet Gpg4win Ihnen eine sogenannte „starke Verschlüsselungstechnik“. „Stark“ bedeutet hier: mit keinem bekannten 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 Unterstützung von Freier Software für die Verschlüsselung. 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 Kompendium wird Ihnen dieses Vorgehen Schritt für Schritt erläutert.

3. So 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 begreifen.

Die Benutzung der einzelnen Programmkomponenten von Gpg4win ist sehr einfach, seine Wirkungsweise dagegen ziemlich kompliziert. Sie werden in diesem Kapitel erklärt bekommen, 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.

Am Ende dieses Buches, in Kapitel 23, können Sie – wenn Sie wollen – auch noch die letzten Geheimnisse um die „Public-Key“-Kryptografie lüften und entdecken, warum mit Gpg4win verschlüsselte Nachrichten nach heutigem Stand der Technik nicht zu knacken sind.

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 und Einmaligkeit des Schlüssels. Also muss man den Schlüssel mindestens genauso gut absichern, wie das zu sichernde Gut selbst. Damit er nicht kopiert werden kann, muss auch die genaue Beschaffenheit des Schlüssels völlig geheim gehalten werden.

Geheime Schlüssel sind in der Kryptografie ein alter Hut: Schon immer hat man Botschaften geheim zu halten versucht, indem man den Schlüssel verbarg. Dies wirklich sicher zu machen, ist sehr umständlich und dazu auch sehr fehleranfällig.



Das Grundproblem bei der „gewöhnlichen“ 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 müssen. Aus diesem Grund nennt man solche Verschlüsselungssysteme auch „**symmetrische Verschlüsselung**“.

Dies führt zu einer ziemlich paradoxen Situation: Bevor man mit einer solchen Methode ein Geheimnis (eine verschlüsselte Nachricht) mitteilen kann, muss man schon vorher ein anderes Geheimnis mitgeteilt haben: den Schlüssel. 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 geheimen Schlüssel – mit einem weiteren Schlüssel (engl. „key“), der vollkommen frei und öffentlich (engl. „public“) zugänglich ist. Man spricht daher auch von einem „Public-Key“-Verschlüsselungssystem.

Das klingt widersinnig, ist es aber nicht. Der Witz an der Sache: Es muss kein geheimer Schlüssel mehr ausgetauscht werden. Im Gegenteil: Der geheime Schlüssel darf auf keinen Fall ausgetauscht werden! Weitergegeben wird nur der öffentliche Schlüssel – und den darf sowieso jeder kennen.

Mit Gpg4win benutzen Sie also ein Schlüsselpaar – einen geheimen und einen zweiten öffentlichen Schlüssel. Beide Schlüsselteile 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 das Verfahren zu knacken.

In Kapitel 23 bekommen Sie erklärt, warum das so ist.



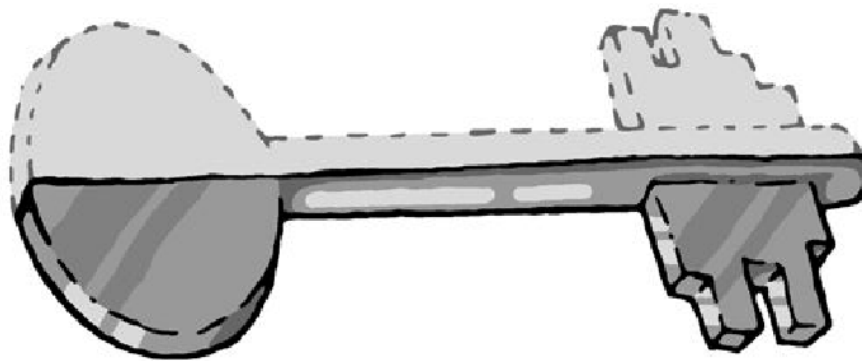
Das Prinzip der Public-Key-Verschlüsselung ist recht einfach:

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

Der **öffentliche Schlüssel** (engl. „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 Briefftresor

In einem kleinen Gedankenspiel wird die Methode des „Public-Key“-Verschlüsselungssystems und ihr Unterschied zur symmetrischen Verschlüsselung („Geheimschlüssel-Methode“ oder engl. „Non-Public-Key“-Methode) deutlicher ...

Die „Geheimschlüssel-Methode“ geht so:

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

Der Briefftresor 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 – so sicher wie in einem Tresor.



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

Diesen Schlüssel müssen Sie Ihrem Korrespondenzpartner auf geheimem Wege übergeben.



Erst wenn der andere den geheimen Schlüssel hat, kann er den Brieftresor öffnen und die geheime Nachricht lesen.

Alles dreht sich also um diesen Schlüssel: Wenn ein Dritter ihn kennt, ist es sofort aus mit den geheimen Botschaften. 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.

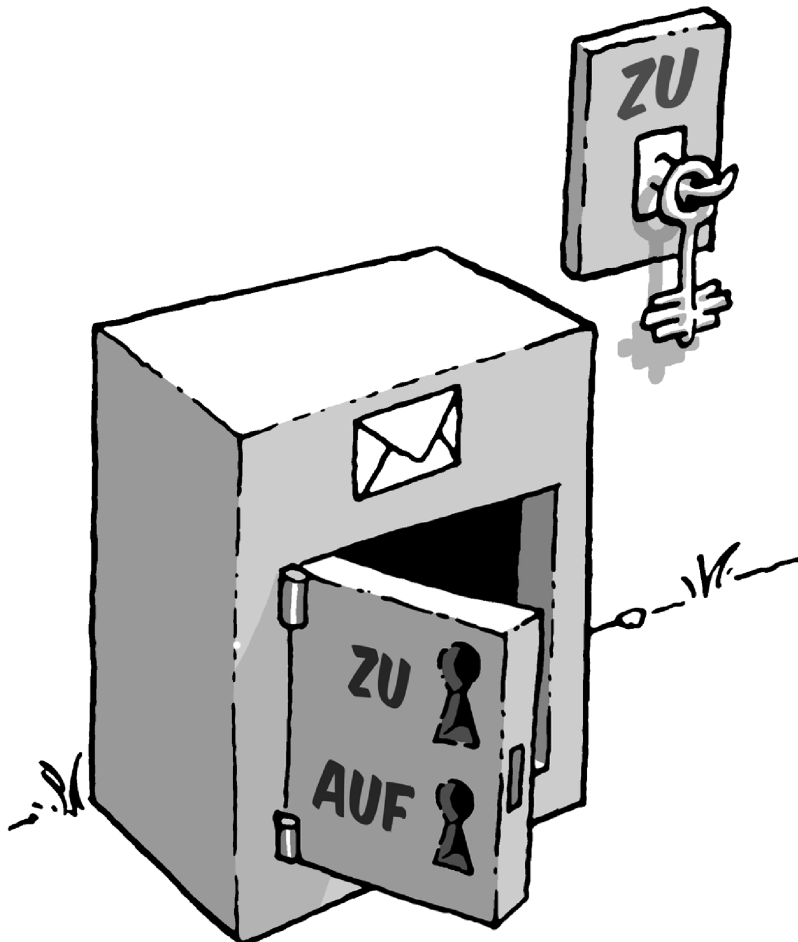
Vergessen Sie diese Möglichkeit am besten sofort wieder ...



Nun zur „Public-Key“-Methode:

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

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 Briefftresor und schließt mit Ihrem öffentlichen Schlüssel ab. Jedermann kann das tun, denn der Schlüssel dazu ist ja völlig frei zugänglich.

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

Denn die öffentliche Schlüsselhälfte taugt ja nur zum Abschießen.

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

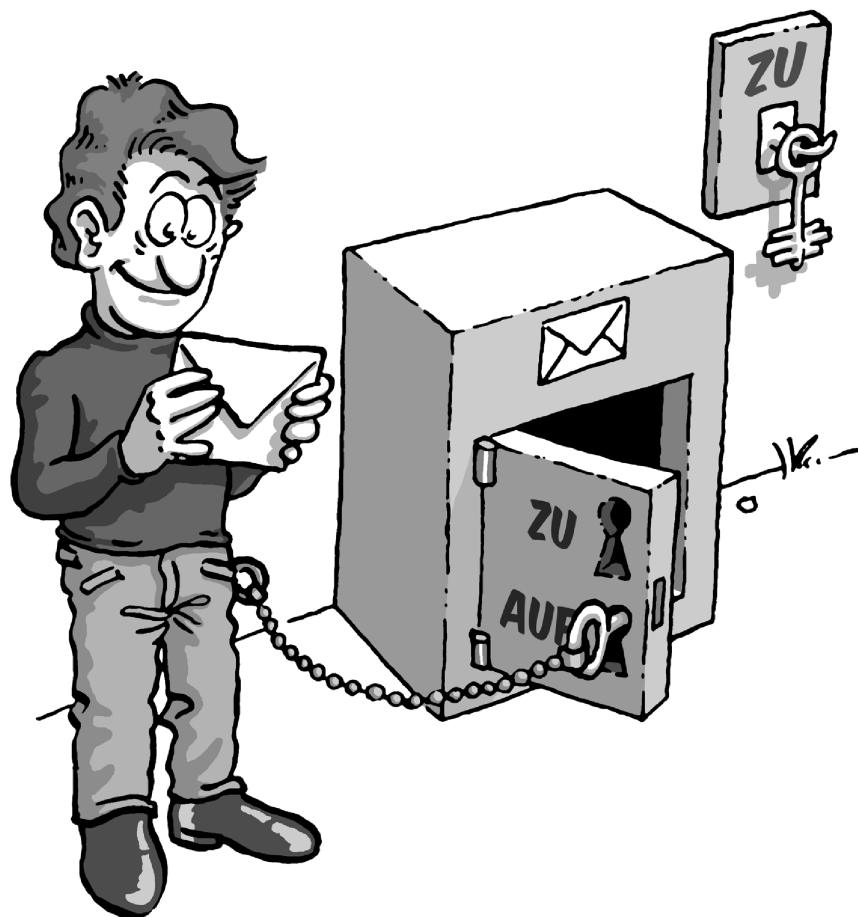
Wieder übertragen auf die E-Mail-Verschlüsselung: Jedermann kann eine E-Mail an Sie verschlüsseln.

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 Sie das Gedankenspiel noch einmal anders herum durch:

Wenn Sie einem anderen eine geheime Nachricht zukommen lassen wollen, benutzen Sie dessen Brieftresor 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 Brieftresor des Empfängers mit seinem öffentlichen Schlüssel wieder verschlossen haben, ist sie völlig unzugänglich für jeden anderen, auch für Sie selbst. Nur der Empfänger kann den Brieftresor mit seinem privaten Schlüssel öffnen und die Nachricht lesen.



Aber was ist nun eigentlich gewonnen: Es gibt doch immer noch einen geheimen Schlüssel!?

Der Unterschied gegenüber der „Non-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 symmetrischen Verschlüsselungsverfahren können geknackt werden, weil ein Dritter sich beim Schlüsselaustausch in den Besitz des Schlüssels bringen kann.

Dieses Risiko entfällt, weil ein geheimer Schlüssel nicht ausgetauscht wird und sich nur an einem einzigen, sehr sicheren Ort befindet: dem eigenen Schlüsselbund – letztendlich Ihrem eigenen Gedächtnis.

Diese moderne Methode der Verschlüsselung mit einem nicht geheimen und öffentlichen sowie einem geheimen und privaten Schlüsselteil nennt man auch „asymmetrische Verschlüsselung“.

4. Die Passphrase

Wie Sie im letzten Kapitel gelesen haben, ist der private Schlüssel eine der wichtigsten Komponenten beim „Public-Key“- oder asymmetrischen Verschlüsselungsverfahren. Man muss ihn zwar nicht mehr auf geheimem Wege mit seinen Korrespondenzpartnern austauschen, aber nach wie vor ist seine Sicherheit der Schlüssel zur Sicherheit des „ganzen“ Kryptografieverfahrens.

Technisch gesehen ist der private Schlüssel einfach eine Datei, die auf dem eigenen Rechner gespeichert wird. Um unbefugte Zugriffe auf diese Datei auszuschließen, wird sie zweifach gesichert:



Zunächst darf kein anderer Benutzer des Rechners die Datei lesen oder in sie schreiben können – was kaum zu garantieren ist, da zum einen der Administrator des Computers immer auf alle Dateien zugreifen kann, zum anderen der Rechner verloren oder durch Viren, Würmer oder Trojaner ausspioniert werden kann.

Daher ist ein weiterer Schutz notwendig: eine Passphrase. Kein Passwort – die Passphrase sollte nicht nur aus einem Wort bestehen, sondern z.B. aus einem Satz. Sie sollten diese Passphrase wirklich „im Kopf“ behalten 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.

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:

niefdahnlnr (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 eine Textzeile aus einem für Sie wichtigen Lied.

Verwenden Sie kleine und große Buchstaben, Nummern, Sonder- und Leerzeichen durcheinander. Im Prinzip ist alles erlaubt, auch Umlaute, Sonderzeichen, Ziffern 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 Umlaute (ä, ö, ü usw.) nur auf einer deutschen 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.

könnte man beispielsweise diese Passphrase machen:

inMinschen stet lh0f 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 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). Die gleiche Passphrase wäre damit bereits einer anderen, möglicherweise unsicheren Software bekannt. Falls hier ein Hacker erfolgreich zuschlägt, ist Ihre Passphrase so gut wie nichts mehr wert.
- ... aus einem Wörterbuch stammt. Passphrase-Knackprogramme können in Minutenschnelle komplette digitale Wörterbücher über ein Passwort laufen lassen – bis eines der Wörter passt.
- ... aus einem Geburtsdatum, einem Namen oder anderen öffentlichen Informationen besteht. Wer vorhat, Ihre E-Mail zu entschlüsseln, wird sich diese Daten beschaffen.
- ... ein landläufiges Zitat ist; wie z.B. „das wird böse enden“ oder „to be or not to be“. Auch mit derartigen gängigen Zitaten testen Passphrase-Knackprogramme eine Passphrase.
- ... aus nur einem Wort oder aus weniger als 8 Zeichen besteht. Denken Sie sich unbedingt 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.

In Kapitel 7 werden Sie diese Passphrase bei der Erzeugung Ihres Schlüsselpaars benötigen.

Vorher müssen Sie aber noch ein weiteres Problem aus dem Weg räumen: Irgendjemand muss beglaubigen, dass die Person, die Ihnen geheime Nachrichten schicken will, auch tatsächlich echt ist.

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

Sie haben gesehen, wie wichtig der „Umschlag“ um Ihre E-Mail ist und wie man ihn mit den Mitteln der modernen Informationstechnologie bereitstellt: ein Brieftresor, in den jedermann verschlüsselte Mails legen kann, die nur Sie als Besitzer des Brieftresors entschlüsseln können. Es ist unmöglich, die Verschlüsselung zu knacken, solange der private Schlüssel zum „Tresor“ Ihr Geheimnis bleibt.

Allerdings: Wenn man genauer darüber nachdenkt, gibt es noch ein zweites Problem. Weiter oben haben Sie gelesen, dass man – im Gegensatz zur Geheimschlüssel-Methode – den Briefpartner nicht persönlich treffen muss, damit er eine geheime Nachricht übermitteln kann. Wie kann man dann aber sicher sein, dass er auch tatsächlich derjenige ist, für den er sich ausgibt? Beim E-Mail-Verkehr kennen Sie in den seltensten Fällen alle Ihre Briefpartner persönlich – und wer sich wirklich hinter einer E-Mail-Adresse verbirgt, kann man nicht ohne Weiteres feststellen. Also muss nicht nur die Geheimhaltung der Nachricht gewährleistet sein, sondern auch die Identität des Absenders – die **Authentizität**.

Irgendjemand muss also beglaubigen, dass die Person, die Ihnen geheime Nachrichten schicken will, auch tatsächlich echt ist. Im Alltagsleben dient zu dieser „Authentisierung“ ein Ausweis, eine Unterschrift oder eine Urkunde, die von einer Behörde oder einem Notar beglaubigt wurde. Die Berechtigung zur Beglaubigung bezieht diese Institution von einer übergeordneten Behörde und letztendlich vom Gesetzgeber. Anders betrachtet, handelt es sich um eine Vertrauenskette, die sich von „oben“ nach „unten“ verzweigt: man spricht von einem **„hierarchischen Vertrauenskonzept“**.

Dieses Konzept findet sich bei Gpg4win oder anderen E-Mail-Verschlüsselungsprogrammen fast spiegelbildlich in **S/MIME** wieder. Dazu kommt **OpenPGP**, ein weiteres Konzept, das so nur im Internet funktioniert. S/MIME und OpenPGP haben beide die gleiche Aufgabe: das Verschlüsseln und Signieren von Daten. Beide benutzen die bereits bekannte Public-Key-Methode. Es gibt zwar einige wichtige Unterschiede, aber letztlich bietet keiner der Standards einen allgemeinen Vorteil gegenüber dem anderen. Deshalb können Sie mit Gpg4win beide Verfahren einsetzen.

Die Entsprechung des hierarchischen Vertrauenskonzepts hat den schönen Namen „Secure / Multipurpose Internet Mail Extension“ oder **S/MIME**. Mit S/MIME müssen Sie Ihren öffentlichen Schlüssel von einer dazu berechtigten Organisation beglaubigen lassen, bevor er wirklich nutzbar wird. Das Zertifikat dieser Organisation wurde wiederum mit dem Zertifikat einer höher stehenden Organisation beglaubigt, usw. – bis man zu einem sogenannten Wurzelzertifikat kommt. Diese hierarchische Vertrauenskette hat meist drei Glieder: das Wurzelzertifikat, das Zertifikat des Zertifikatsausstellers (auch CA für Certificate Authority genannt) und schließlich Ihr eigenes, das Anwenderzertifikat.

Als zweite, alternative, nicht kompatible Methode der Beglaubigung dient der Standard **OpenPGP**, der keine Vertrauenshierarchie aufbaut, sondern ein „**Netz des Vertrauens**“ (Web of Trust). Das Web of Trust bildet die Grundstruktur des nicht hierarchischen Internets und seiner Nutzer nach. Vertraut zum Beispiel der Teilnehmer B dem Teilnehmer A, könnte B auch dem öffentlichen Schlüssel des ihm selbst unbekannten Teilnehmers C vertrauen, wenn dieser Schlüssel durch A beglaubigt wurde.

Mit OpenPGP besteht also die Möglichkeit, ohne die Beglaubigung einer höheren Stelle verschlüsselte Daten und E-Mails auszutauschen. Es reicht aus, wenn Sie der E-Mail-Adresse und dem dazugehörigen Schlüssel Ihres Kommunikationspartners vertrauen.

Ob nun mit einer Vertrauenshierarchie oder einem Web of Trust – die Authentisierung des Absenders ist mindestens ebenso wichtig wie der Schutz der Nachricht. Im weiteren Verlauf dieses Kompendiums kommen wir auf diese wichtige Sicherheitsmaßnahme noch einmal zurück. Im Moment sollte Ihnen dieser Kenntnisstand ausreichen, um Gpg4win zu installieren und die folgenden Kapitel zu verstehen:

- Beide Verfahren – **OpenPGP** und **S/MIME** – bieten die notwendige Sicherheit.
- Die Verfahren sind **nicht kompatibel** miteinander. Sie bieten zwei alternative Methoden zur Authentisierung Ihrer geheimen Kommunikation. Man sagt somit, sie sind nicht interoperabel.
- Gpg4win ermöglicht die bequeme **parallele** Nutzung beider Verfahren – Sie müssen sich aber bei jeder Verschlüsselung/Signierung für eines der beiden entscheiden.

6. Installation von Gpg4win

In den Kapiteln 1 bis 5 haben Sie einiges über die Hintergründe der Verschlüsselung erfahren. 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.

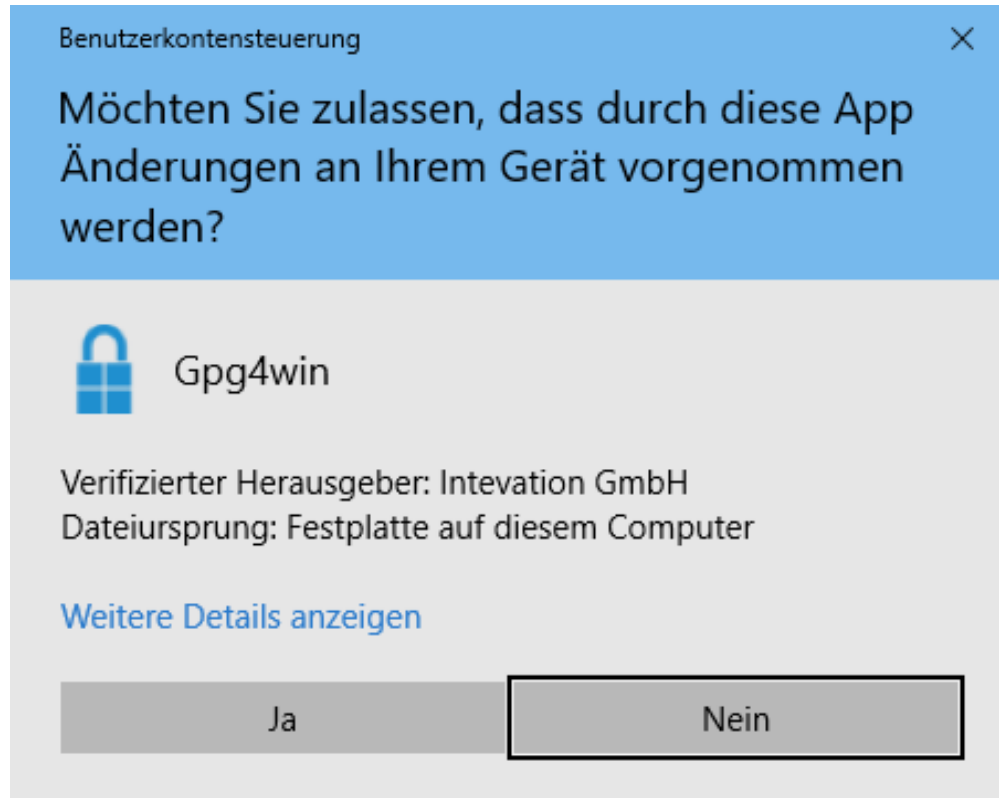
Mit diesem Wissen sind Sie nun bereit, Gpg4win zu installieren und Ihr Schlüsselpaar einzurichten.

Sollte bereits eine GnuPG-basierte Anwendung auf Ihrem Rechner installiert sein, dann lesen bitte im Anhang D nach, wie Sie Ihre vorhandenen Schlüssel übernehmen können.

Sie benötigen für die Installation auf Ihrem Windows 32 oder 64-bit System Administratorrechte.

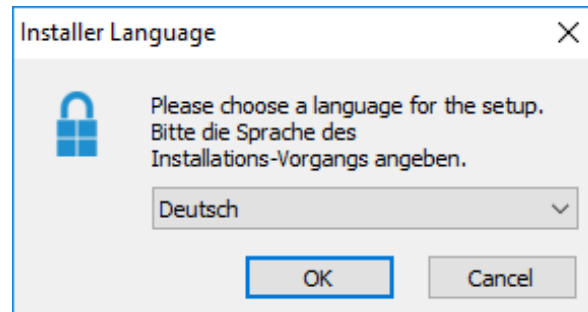
Wenn Sie Gpg4win aus dem Internet laden, achten Sie unbedingt darauf, dass Sie die Datei von einer vertrauenswürdigen Seite erhalten, z.B.: <https://www.gpg4win.de>. Zum Start der Installation klicken Sie nach dem Download auf die Datei:

`gpg4win-3.0.0.exe` (oder mit einer höheren Versionsnummer).



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

Der Installationsassistent startet und befragt Sie zuerst nach der Sprache für den Installationsvorgang:



Bestätigen Sie Ihre Sprachauswahl mit [OK].

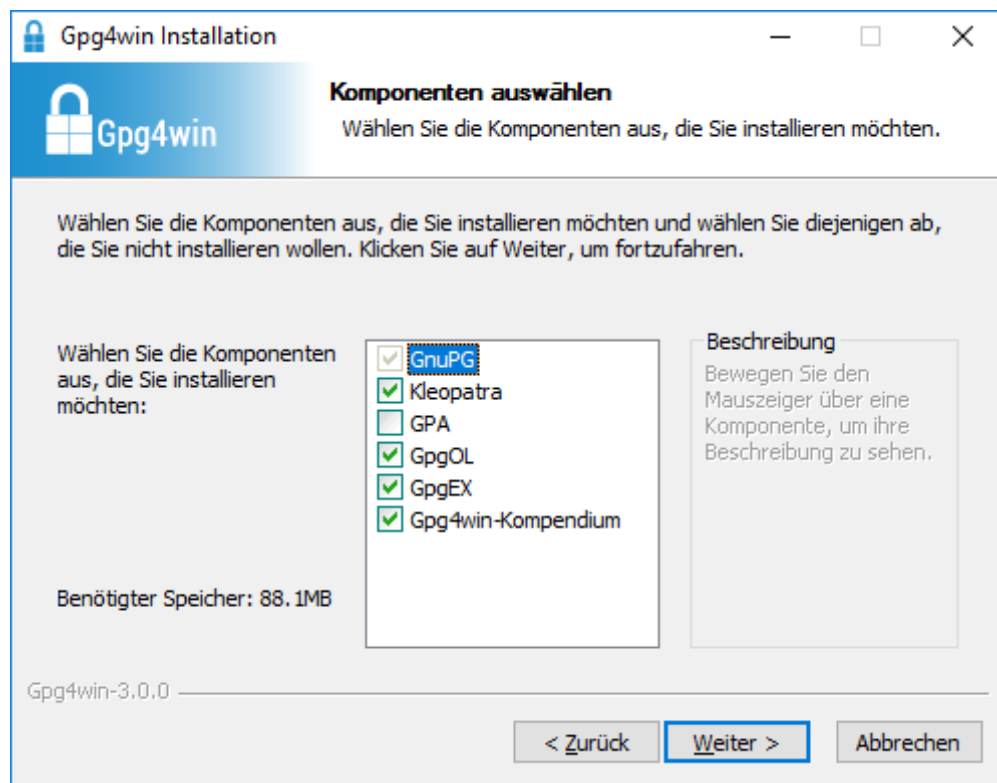
Anschließend begrüßt Sie dieser Willkommensdialog:



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

Auf der Seite mit der **Komponentenauswahl** können Sie entscheiden, welche Programme Sie installieren möchten. Eine Vorauswahl ist bereits getroffen. Sie können bei Bedarf einzelne Komponenten auch später installieren.

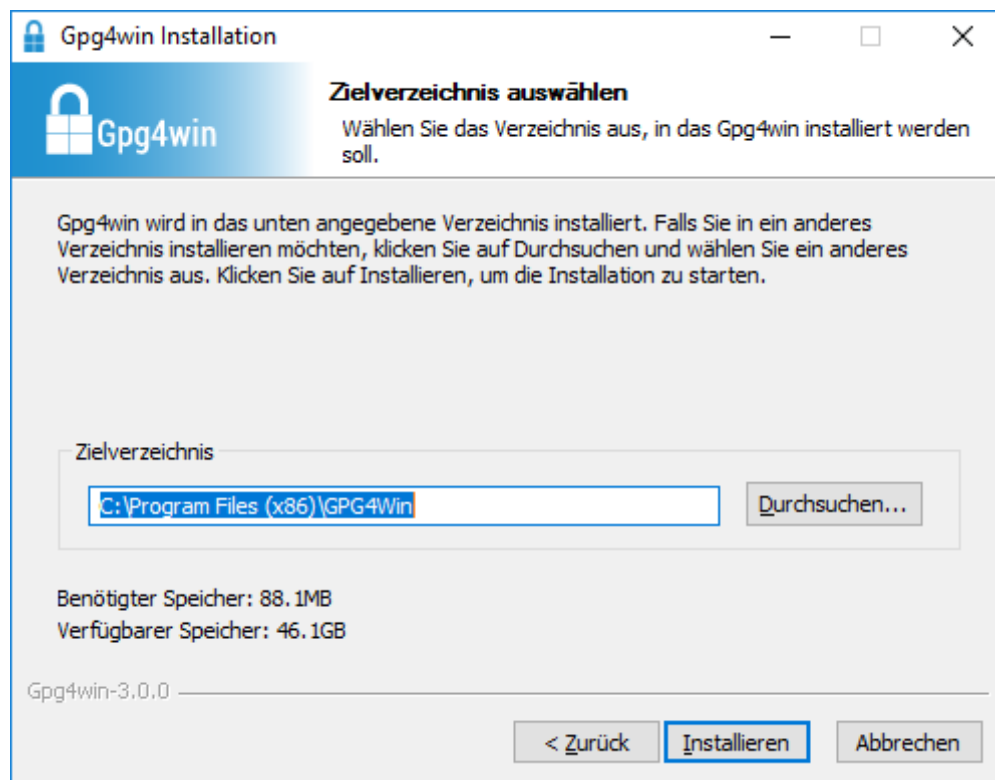
Wenn Sie die Maus über eine Komponente ziehen, erscheint eine Kurzbeschreibung.



Klicken Sie auf [*Weiter*].

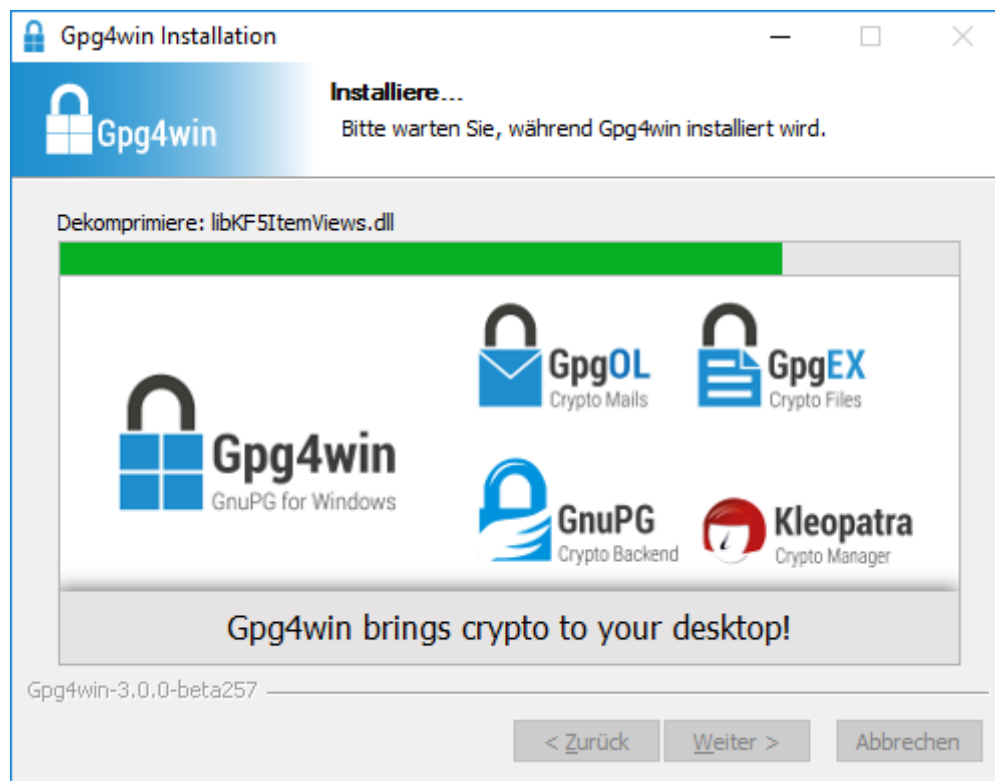
Nun wird Ihnen ein Ordner zur Installation vorgeschlagen, z.B.: C:\Program Files (x86)\Gpg4win

Übernehmen Sie den Vorschlag oder suchen Sie einen anderen Ordner aus, in dem Sie Gpg4win installieren wollen.



Klicken Sie anschließend auf [*Installieren*].

Während der nun folgenden **Installation** sehen Sie einen Fortschrittsbalken und Informationen, welche Datei momentan installiert wird.



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

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



Es wird Ihnen angeboten Kleopatra direkt zu starten. Zudem haben Sie die Möglichkeit sich die README-Datei anzeigen zu lassen, die wichtige Informationen zu der soeben installierten Gpg4win-Version enthält. Sofern Sie die README-Datei ansehen wollen, aktivieren Sie diese Option.

Klicken Sie schließlich auf [*Fertig stellen*].

Das war's schon!

Sie haben Gpg4win erfolgreich installiert und können nun loslegen.

Für Informationen zur **automatischen Installation** von Gpg4win, wie sie z.B. für Softwareverteilungssysteme interessant ist, lesen Sie bitte im Anhang C „Automatische Installation von Gpg4win“ weiter.

7. Erstellung eines Schlüsselpaars

Nachdem Sie gelesen haben, warum GnuPG eigentlich so sicher ist (Kapitel 3) und wie eine gute Passphrase als Schutz Ihres geheimen Schlüssels entsteht (Kapitel 4), können Sie nun Ihr persönliches Schlüsselpaar erzeugen.

Wie Sie im Kapitel 3 gesehen haben, besteht ein Schlüsselpaar aus einem öffentlichen und einem geheimen Schlüssel. Ergänzt durch E-Mail-Adresse, Benutzerkennung etc., die Sie bei der Erstellung angeben (den sogenannten Metadaten), erhalten Sie Ihr Schlüsselpaar mit dem öffentlichen *und* dem geheimen Schlüssel.

Diese Definition gilt sowohl für OpenPGP-Schlüssel wie auch für S/MIME-Zertifikate (S/MIME-Zertifikate entsprechen einem Standard mit der Bezeichnung „X.509“).

Eigentlich müsste man diesen wichtigen Schritt der Schlüsselpaar-Erzeugung ein paar Mal üben können ...

Genau das können Sie tun – allerdings nur für OpenPGP:

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

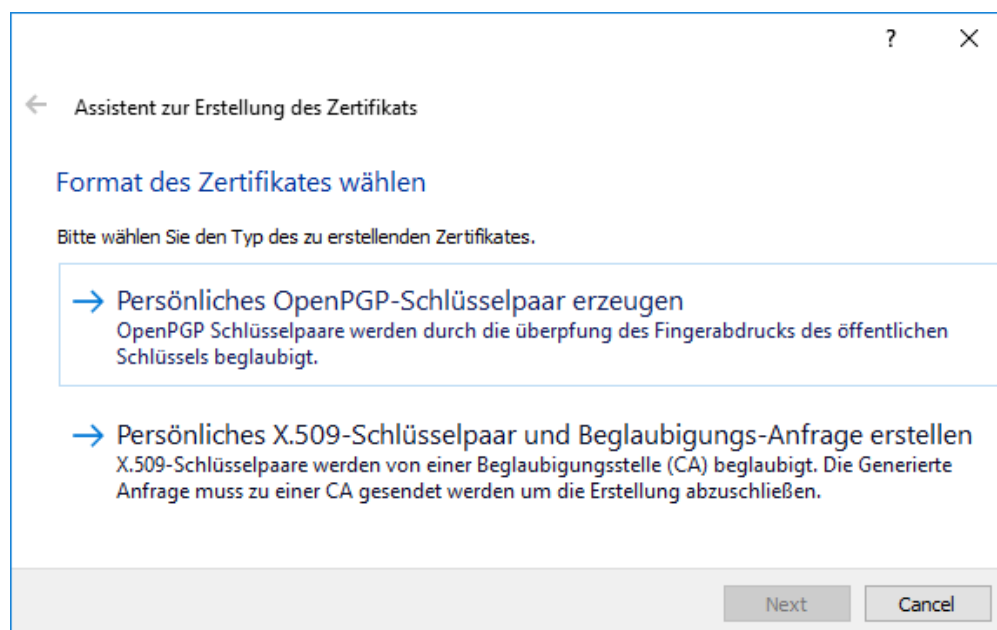
Los geht's! Rufen Sie das Programm Kleopatra über das Windows-Startmenü auf. Daraufhin sehen Sie das Hauptfenster von Kleopatra:



Zu Beginn ist diese Übersicht leer, da Sie noch keine Schlüssel erstellt (oder importiert) haben.

Klicken Sie auf [*Schlüsselpaar erstellen*] (oder alternativ *Datei*→*Neues Schlüsselpaar*).

Im folgenden Dialog entscheiden Sie sich für ein Format, in dem anschließend ein Schlüsselpaar erstellt werden soll. Sie haben die Wahl zwischen **OpenPGP** (PGP/MIME) oder **X.509** (S/MIME). Die Unterschiede und Gemeinsamkeiten beider Verfahren wurden bereits in Kapitel 5 erläutert.



Je nachdem, ob Sie sich für OpenPGP oder X.509 (S/MIME) entschieden haben, lesen Sie nun also bitte entweder:

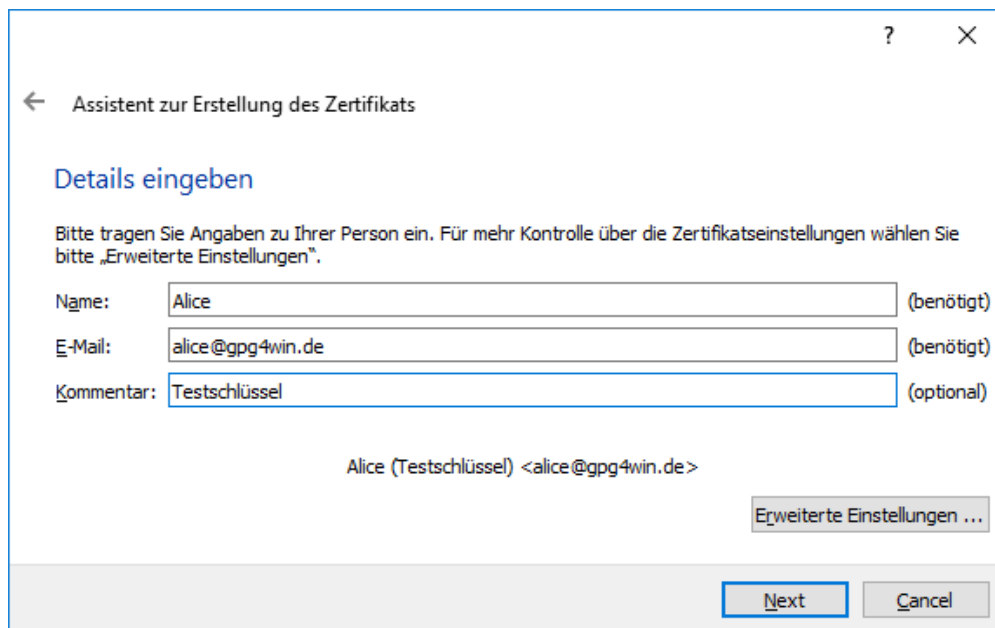
- Abschnitt 7.1: **OpenPGP-Schlüsselpaar erstellen** oder
- Abschnitt 7.2: **X.509-Zertifikat erstellen** (siehe Seite 45).

7.1. OpenPGP-Schlüsselpaar erstellen

Klicken Sie im Auswahldialog auf [*Persönliches OpenPGP-Schlüsselpaar erzeugen*].

Geben Sie im nun folgenden Dialog Ihren Namen und Ihre E-Mail-Adresse an. Name und E-Mail-Adresse sind später öffentlich sichtbar.

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



The screenshot shows a Windows-style dialog box titled "Assistent zur Erstellung des Zertifikats". It has a back arrow icon and a title bar with a question mark and a close button. The main heading is "Details eingeben". Below it, a text instruction says: "Bitte tragen Sie Angaben zu Ihrer Person ein. Für mehr Kontrolle über die Zertifikateinstellungen wählen Sie bitte „Erweiterte Einstellungen“." There are three input fields: "Name:" with the value "Alice" (marked "(benötigt)"), "E-Mail:" with the value "alice@gpg4win.de" (marked "(benötigt)"), and "Kommentar:" with the value "Testschlüssel" (marked "(optional)"). Below these fields, the text "Alice (Testschlüssel) <alice@gpg4win.de>" is displayed. At the bottom right, there is a button labeled "Erweiterte Einstellungen ...". At the very bottom, there are "Next" and "Cancel" buttons.

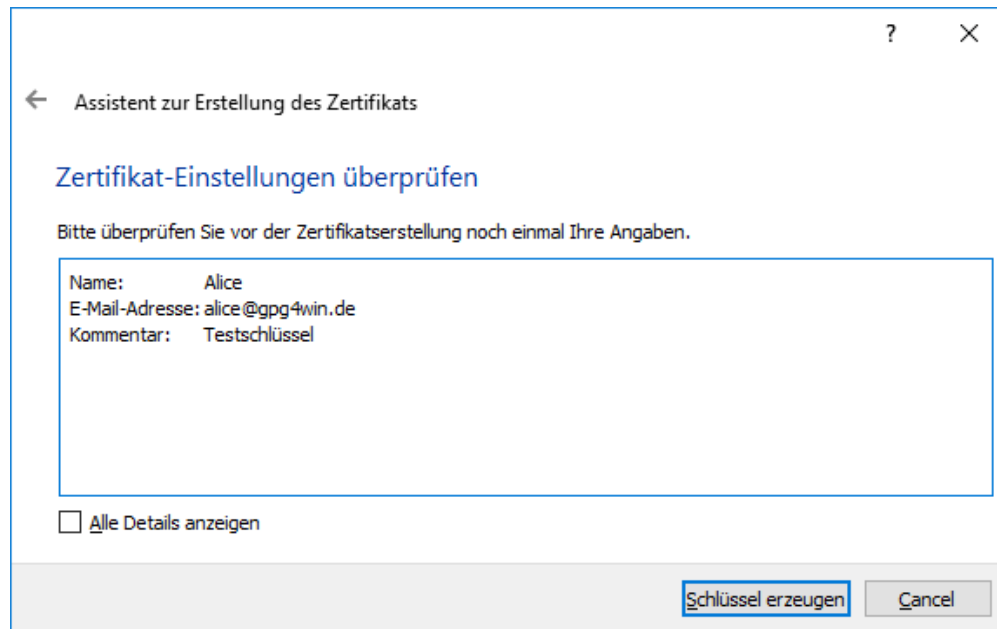
Wenn Sie die OpenPGP-Schlüsselpaar-Erzeugung zunächst einmal **testen** wollen, dann können Sie einfach einen beliebigen Namen und irgendeine ausgedachte E-Mail-Adresse eingeben, z.B.:

Alice und `alice@gpg4win.de`

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

Klicken Sie auf [*Weiter*].

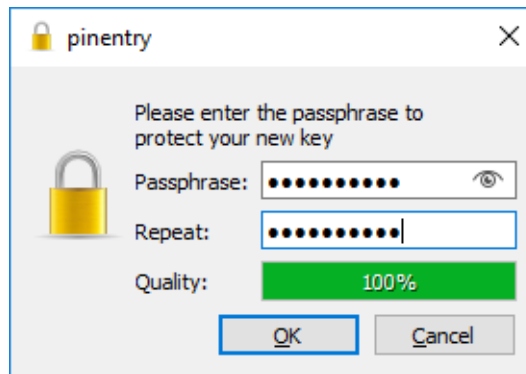
Es werden abschließend noch einmal alle wesentlichen Eingaben und Einstellungen zur **Kontrolle** aufgelistet. Falls Sie sich für die (vorbelegten) Experten-Einstellungen interessieren, können Sie diese über die Option *Alle Details* einsehen.



Wenn alles korrekt ist, klicken Sie anschließend auf [*Schlüssel erzeugen*].

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

Für die Schlüsselpaarerstellung müssen Sie Ihre persönliche Passphrase eingeben:



Wenn Sie Kapitel 4 gelesen haben, dann sollten Sie jetzt eine einfach zu merkende und schwer zu knackende geheime Passphrase parat haben. Geben Sie diese in den oben gezeigten Dialog ein!

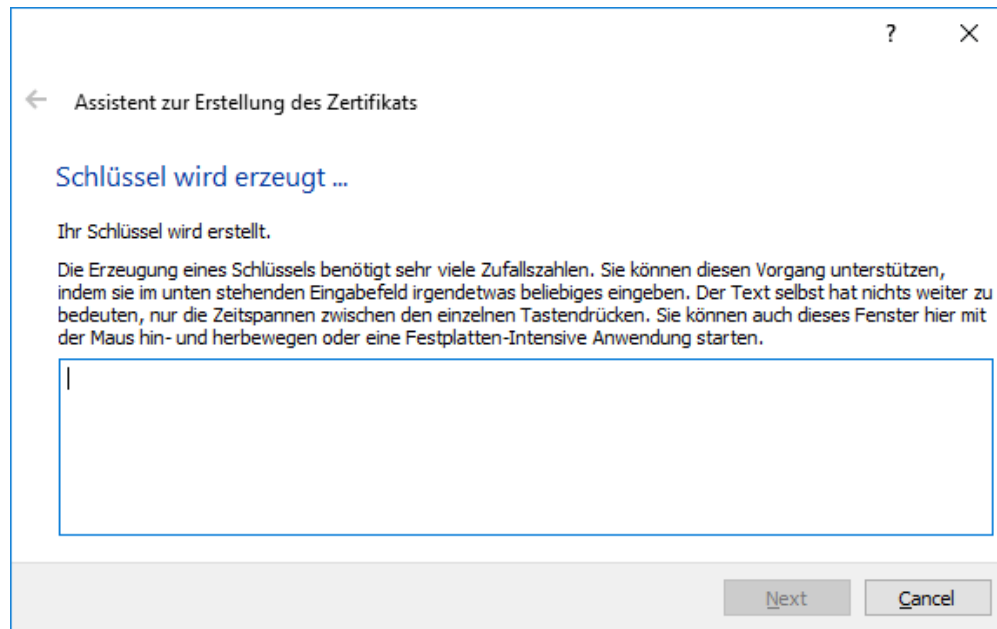
Beachten Sie bitte, dass dieses Fenster unter Umständen im Hintergrund geöffnet wurde und damit auf den ersten Blick nicht sichtbar ist.

Wenn die Passphrase nicht sicher genug ist, weil sie zu kurz ist oder keine Zahlen oder 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“.

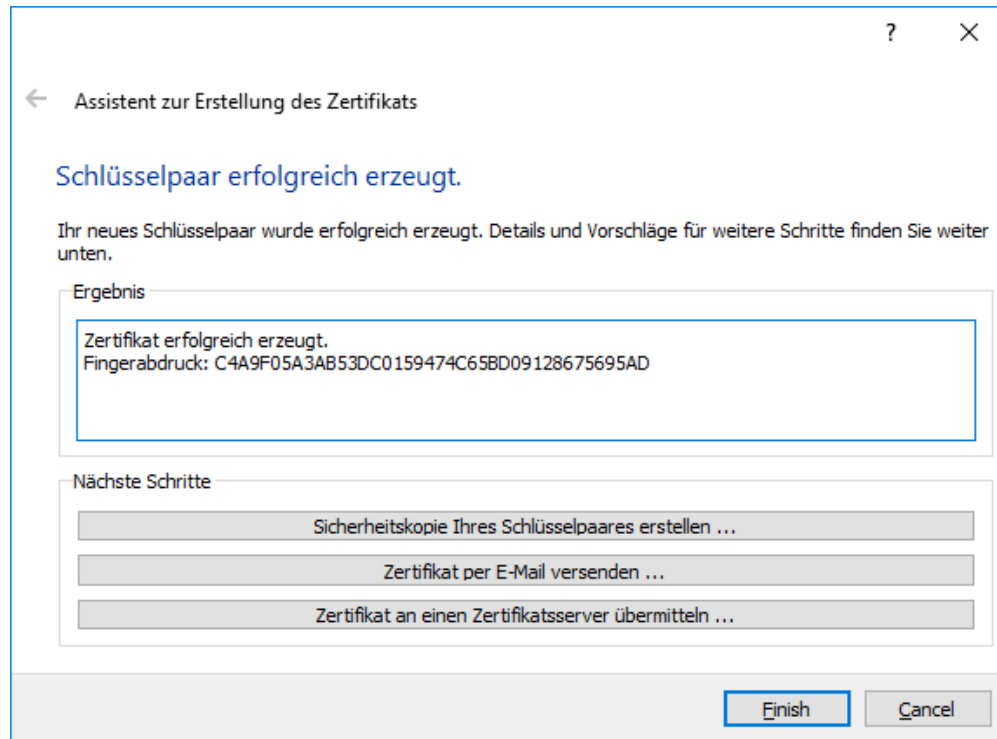
Um sicherzugehen, dass Sie sich nicht vertippt haben, müssen Sie Ihre geheime Passphrase zweimal eingeben. Bestätigen Sie Ihre Eingabe jeweils mit [*OK*].

Nun wird Ihr OpenPGP-Schlüsselpaar erzeugt:



Dies kann u.U. einige Minuten dauern. Sie können die Erzeugung der benötigten Zufallszahlen unterstützen, indem Sie im unteren Eingabefeld irgendetwas eingeben. Was Sie dort tippen, spielt keine Rolle: was Sie schreiben, wird nicht verwendet, nur die Zeitspannen zwischen den einzelnen Tastendrücken. Sie können auch mit einer anderen Anwendung Ihres Rechner weiterarbeiten und erhöhen damit ebenfalls leicht die Qualität des erzeugten Schlüsselpaars.

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



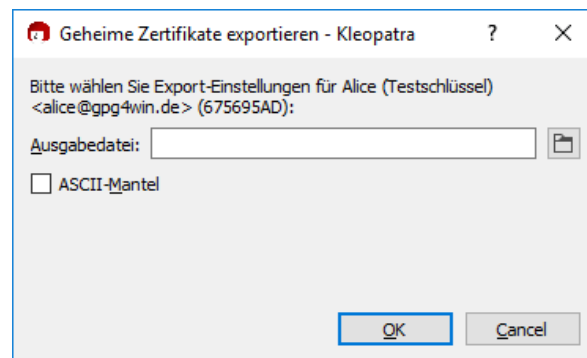
Im Ergebnis-Textfeld wird der 40-stellige „Fingerabdruck“ Ihres neu generierten OpenPGP-Schlüssels angezeigt. Dieser Fingerabdruck (engl. „Fingerprint“) ist weltweit eindeutig, d.h. keine andere Person besitzt einen Schlüssel mit identischem Fingerabdruck. Es ist sogar vielmehr so, dass es schon mit 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 und als Schlüsselkennung (oder Schlüssel-ID) bezeichnet. Dieser Fingerabdruck identifiziert die Identität des Schlüssels wie der Fingerabdruck einer Person.

Sie brauchen sich den Fingerabdruck nicht zu merken oder abzuschreiben. In den Details von Kleopatra können Sie sich ihn jederzeit später anzeigen lassen.

Als Nächstes können Sie eine oder auch (hintereinander) mehrere der folgenden drei Schaltflächen betätigen:

Sicherheitskopie Ihres (geheimen) Schlüssels erstellen...

Geben Sie hier den Pfad an, unter dem Ihr vollständiges Schlüsselpaars (also der geheime *und* öffentliche Schlüssel) exportiert werden soll:



Kleopatra wählt automatisch den Dateityp und speichert Ihren Schlüssel als `.asc` bzw. `.gpg` Datei ab – abhängig davon, ob Sie die Option **ASCII-geschützt** (engl. „ASCII armor“) ein- bzw. ausschalten.

Klicken Sie anschließend zum Exportieren auf [*OK*].

Wichtig: Falls Sie die Datei auf der Festplatte abspeichern, so sollten Sie diese Datei schnellstens auf einen anderen Datenträger (USB-Stick, Diskette oder CD-ROM) kopieren und die Originaldatei rückstandslos löschen, d.h. nicht im Papierkorb belassen! Bewahren Sie diesen Datenträger mit der Sicherheitskopie sicher auf.

Sie können eine Sicherheitskopie auch noch später anlegen; wählen Sie hierzu aus dem Kleopatra-Hauptmenü: *Datei*→*Geheimes Zertifikat exportieren...* (vgl. Kapitel 17).

Schlüssel per E-Mail versenden...

Nach dem Klick auf diese Schaltfläche sollte eine neue E-Mail erstellt werden – mit Ihrem neuen öffentlichen Schlüssel 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. Es geht aber natürlich auch manuell: Sollte sich kein neues E-Mail-Fenster öffnen, so beenden Sie den Assistenten, speichern Ihren öffentlichen Schlüssel durch *Datei*→*Zertifikat exportieren* und versenden diese Datei per E-Mail an Ihre Korrespondenzpartner.

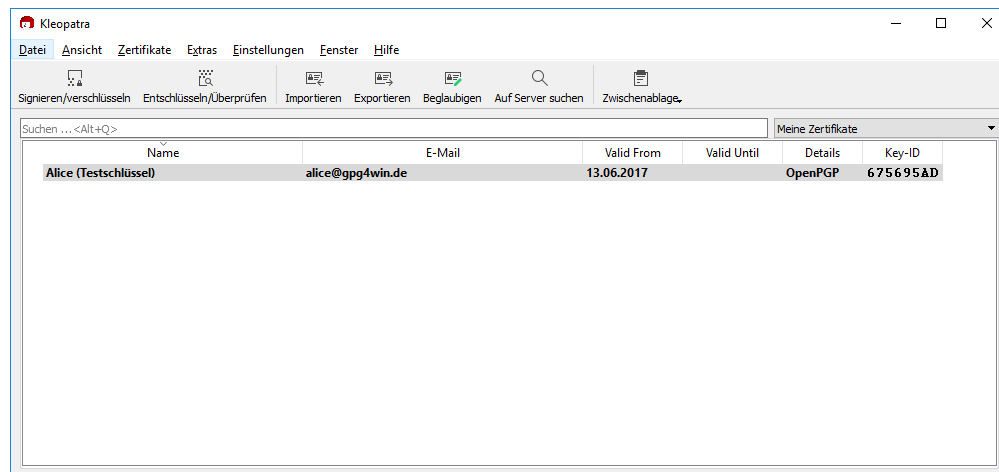
Zertifikate zu Zertifikatsservern senden...

Wie Sie einen weltweit verfügbaren OpenPGP-Zertifikatsserver in Kleopatra einrichten und wie Sie anschließend Ihr öffentliches Zertifikat auf diesem Server veröffentlichen, erfahren Sie in Kapitel 15.

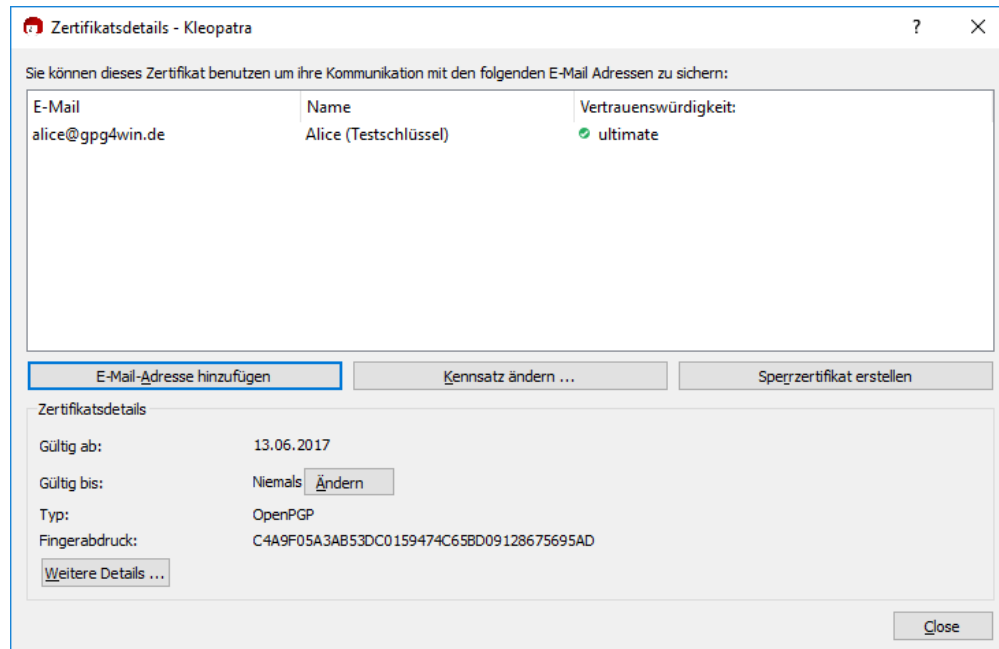
Ihr OpenPGP-Schlüsselpaar ist damit fertig erstellt. Beenden Sie anschließend den Kleopatra-Assistenten mit [*Fertigstellen*].

Damit ist die Erzeugung Ihres OpenPGP-Schlüsselpaars abgeschlossen. Sie besitzen nun einen einzigartigen elektronischen Schlüssel.

Sie befinden sich nun wieder im Hauptfenster von Kleopatra. Den soeben erzeugten OpenPGP-Schlüssel finden Sie in der Übersicht:



Doppelklicken Sie auf Ihren neuen Schlüssel, um alle Details sehen zu können:



Was bedeuten die einzelnen Details?

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

Weitere Details zum Schlüssel finden Sie im Kapitel 14.

7.2. X.509-Zertifikat erstellen

Klicken Sie im Schlüsselpaar-Auswahldialog von Seite 37 auf die Schaltfläche
[*Persönliches X.509-Schlüsselpaar und Beglaubigungs-Anfrage erstellen*].



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

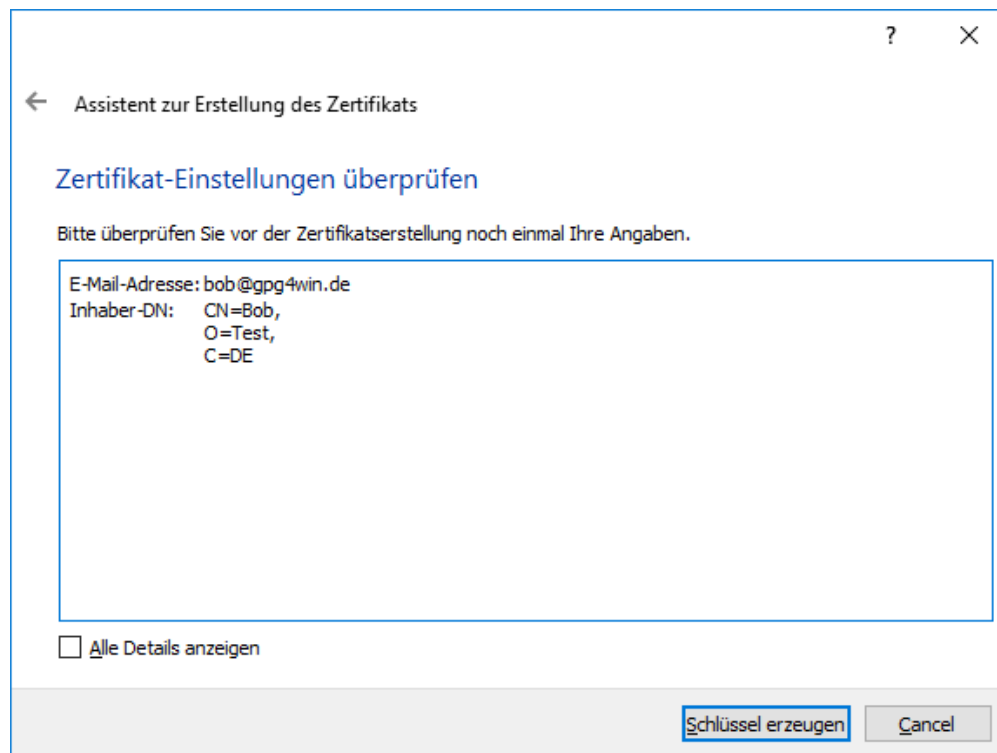
Wenn Sie die X.509-Schlüsselpaar-Erzeugung zunächst einmal **testen** wollen, dann machen Sie beliebige Angaben für Name, Organisation sowie Ländercode und geben irgendeine ausgedachte E-Mail-Adresse ein, z.B.: CN=Bob, O=Test, C=DE, EMAIL=bob@gpg4win.de

The screenshot shows a Windows-style dialog box titled 'Assistent zur Erstellung des Zertifikats'. It has a back arrow and a title bar with a question mark and a close button. The main heading is 'Details eingeben'. Below it, a text instruction says: 'Bitte tragen Sie Angaben zu Ihrer Person ein. Für mehr Kontrolle über die Zertifikateinstellungen wählen Sie bitte „Erweiterte Einstellungen“.' There are several input fields: 'Allgemeiner Name (CN):' with 'Bob' entered, 'E-Mail-Adresse (EMAIL):' with 'bob@gpg4win.de' entered, 'Ort (L):' (empty), 'Abteilung (OU):' (empty), 'Organisation (O):' with 'Test' entered, and 'Ländercode (C):' with 'DE' entered. To the right of each field is a label: '(benötigt)' for CN, EMAIL, O, and C; '(optional)' for L and OU. Below the fields, the text 'CN=Bob,O=Test,C=DE' is displayed. There is a checkbox labeled 'E-Mail-Adresse in DN aufnehmen (nötig für fehlerhafte CAs)' which is currently unchecked. At the bottom right, there is a button labeled 'Erweiterte Einstellungen ...'. At the very bottom, there are 'Next' and 'Cancel' buttons.

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

Klicken Sie auf [*Weiter*].

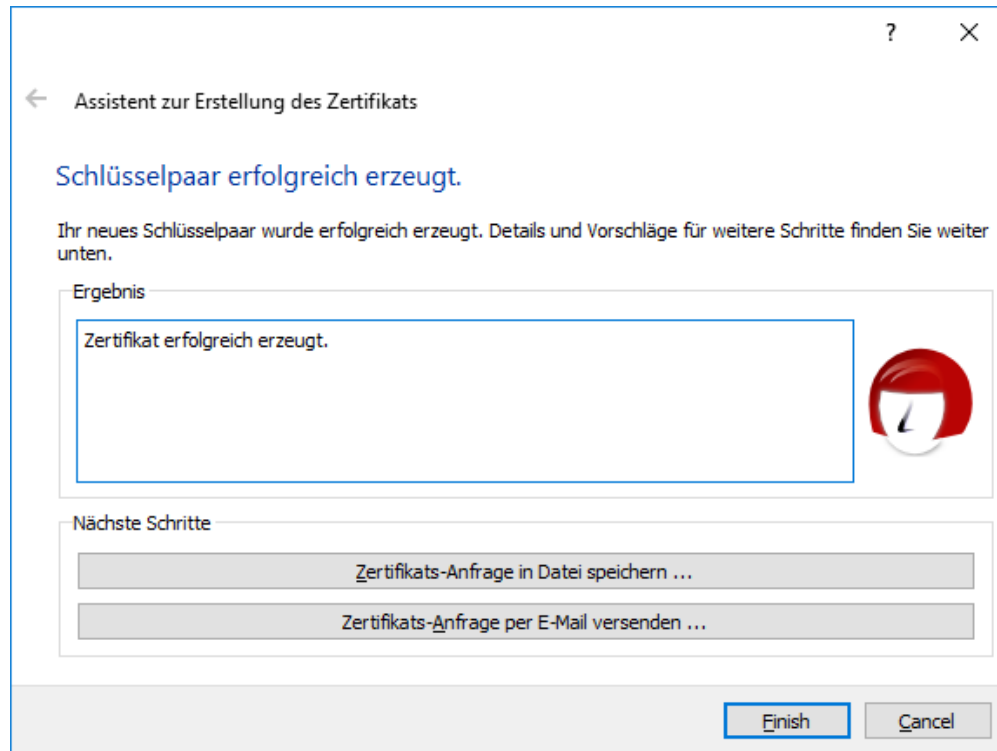
Es werden nun noch einmal alle wesentlichen Eingaben und Einstellungen zur **Kontrolle** aufgelistet. Falls Sie sich für die (vorbelegten) Experten-Einstellungen interessieren, können Sie diese über die Option *Alle Details* einsehen.



Wenn alles korrekt ist, klicken Sie auf [*Schlüssel erzeugen*].

Jetzt folgt der wichtigste Teil: die Eingabe Ihrer **Passphrase**! Das Vorgehen ist analog zu OpenPGP aus dem vorherigen Abschnitt 7.1.

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



Die nächsten Schritte werden durch die beiden folgenden Schaltflächen ausgelöst:

Anfrage in Datei speichern...

Geben Sie hier den Pfad an, unter dem Ihre X.509-Zertifikatsanfrage gesichert werden soll, und bestätigen Sie Ihre Eingabe. Kleopatra fügt beim Speichern automatisch die Dateiendung `.p10` hinzu. Diese Datei kann später an eine Beglaubigungsinstanz (kurz CA für Certificate Authority) gesendet werden. Etwas weiter unten weisen wir Sie auf cacert.org hin, eine nicht kommerzielle Beglaubigungsinstanz (CA), die kostenlos X.509-Zertifikate ausstellt.

Anfrage per E-Mail versenden...

Es wird eine neue E-Mail erstellt – mit der soeben erstellten Zertifikatsanfrage im Anhang. Geben Sie eine Empfänger-E-Mail-Adresse an – in der Regel die Ihrer zuständigen Beglaubigungsinstanz – und ergänzen Sie ggf. den vorbereiteten Text dieser E-Mail.

Beachten Sie: Nicht alle E-Mail-Programme unterstützen diese Funktion. Es geht aber natürlich auch manuell: Sollte sich kein neues E-Mail-Fenster öffnen, dann speichern Sie Ihre Anfrage zunächst in eine Datei (siehe oben) und versenden diese Datei per E-Mail an Ihre Beglaubigungsinstanz (Certificate Authority, CA).

Sobald die Anfrage von der CA bearbeitet wurde, erhalten Sie von Ihrem zuständigen CA-Systemadministrator das fertige und von der CA unterzeichnete X.509-Zertifikat. Dieses müssen Sie dann nur noch in Kleopatra importieren (vgl. Kapitel 17).

Beenden Sie anschließend den Kleopatra-Assistenten mit [*Fertigstellen*].

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

CACert ist eine nicht kommerzielle Beglaubigungsinstanz (CA), die kostenlos X.509-Zertifikate ausstellt. Damit wird eine Alternative zu den kommerziellen Root-CAs geboten, die zum Teil recht hohe Gebühren für ihre Zertifikate erheben.

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

Sofort anschließend können Sie ein oder mehrere Client-Zertifikat(e) auf cacert.org erstellen: Sie sollten dabei auf eine ausreichende Schlüssellänge (z.B. 2048 Bit) achten. Im dortigen Web-Assistenten legen Sie Ihre sichere Passphrase für Ihr Zertifikat fest.

Ihr Client-Zertifikat 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 Sie Ihr Zertifikat in Ihrem Browser. Bei Firefox können Sie danach z.B. über *Bearbeiten*→*Einstellungen*→*Erweitert*→*Zertifikate* Ihr installiertes Zertifikat unter dem ersten Reiter „Ihre Zertifikate“ mit dem Namen (CN) **CACert WoT User** finden.

Sie können nun ein persönliches X.509-Zertifikat ausstellen, das Ihren Namen im CN-Feld trägt. Dazu müssen Sie Ihren CACert-Account von anderen Mitgliedern des CACert-Web-of-Trust beglaubigen lassen. Wie Sie eine derartige Bestätigung in die Wege leiten, erfahren Sie auf den Internetseiten von CACert.

Speichern Sie abschließend eine Sicherungskopie Ihres persönlichen X.509-Zertifikats. Die Sicherungskopie erhält automatisch die Endung `.p12`.

Achtung: Diese `.p12` Datei enthält Ihren öffentlichen *und* Ihren geheimen Schlüssel. Achten Sie daher unbedingt darauf, dass diese Datei nicht in fremde Hände gelangt.

Wie Sie Ihr persönliches X.509-Zertifikat in Kleopatra importieren, erfahren Sie in Kapitel 17.

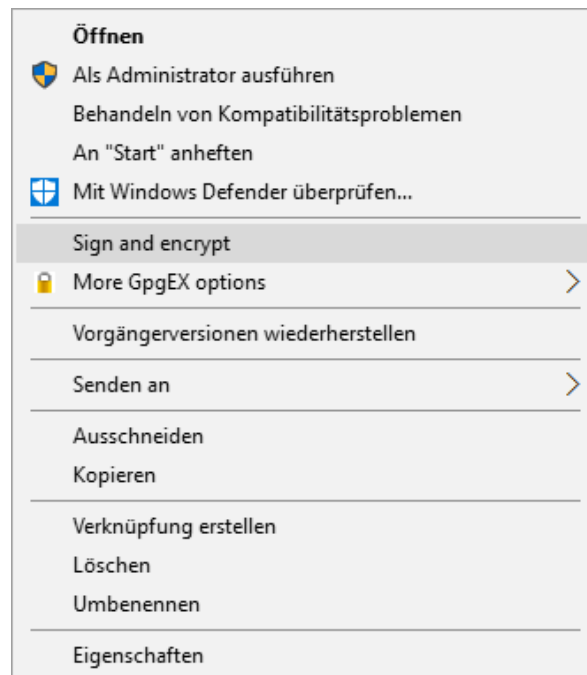
8. Schnellstart mit Übungen für OpenPGP

In den folgenden zwei Unterkapiteln werden Ihnen zwei Anleitungen für einen Schnelleinstieg in die wichtigsten Funktionen gegeben. Sie werden lernen, wie man Dateien und E-Mails ver- und entschlüsselt.

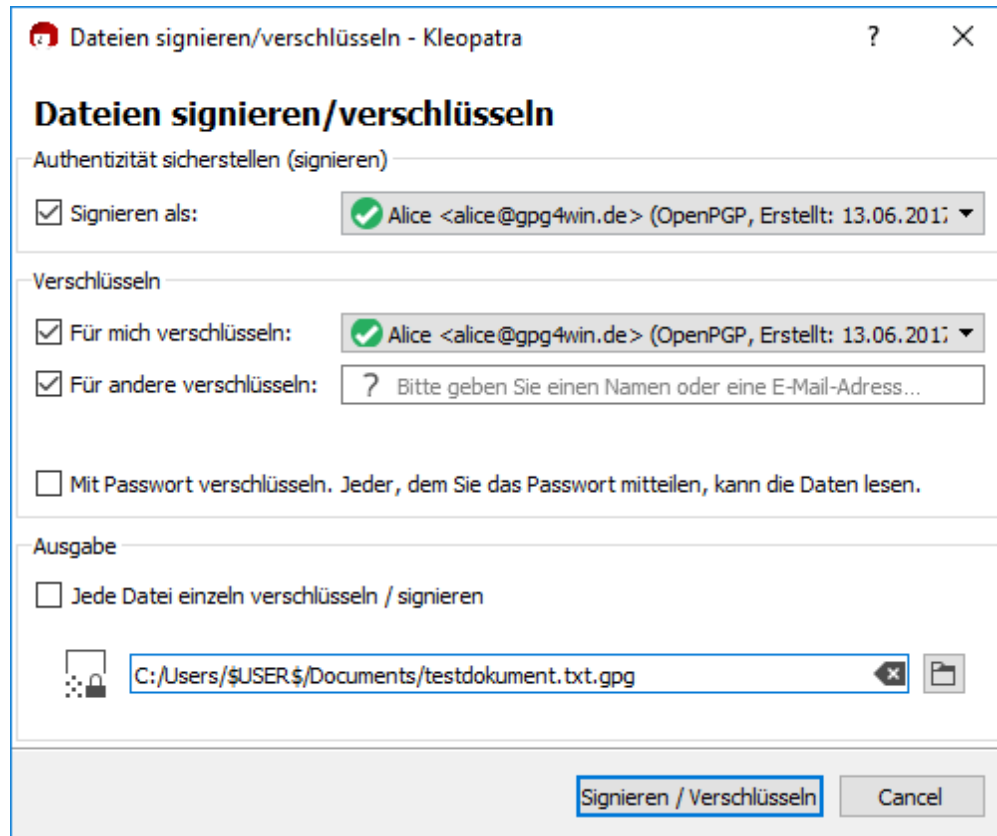
8.1. Dateiverschlüsselung

Dateien lassen sich, wie E-Mails, signieren und verschlüsseln. Das sollten Sie im folgenden Abschnitt mit GpgEX und Kleopatra einmal durchspielen.

Selektieren Sie eine (oder mehrere) Datei(en), öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie hier *Signieren und verschlüsseln* aus:



Sie erhalten diesen Dialog zum Signieren/Verschlüsseln einer Datei:



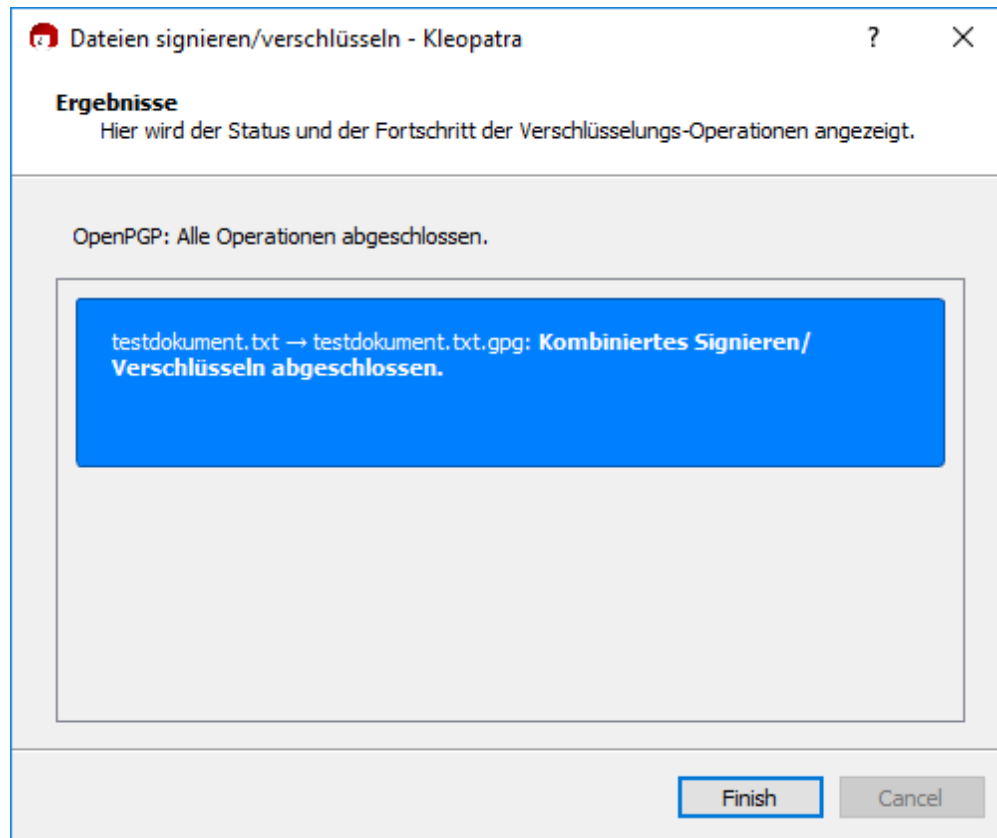
Für wen soll die Datei verschlüsselt werden? Wählen Sie im folgenden Dialog einen oder mehrere Empfänger-Schlüssel aus. Lassen Sie aber unbedingt die Option *Für mich verschlüsseln* aktiv.

Wie sie weitere Einstellungen ändern, können Sie unter Abschnitt 12.1 nachschlagen.

Klicken Sie abschließend auf [*Signieren / Verschlüsseln*].

Geben Sie nun Ihre geheime Passphrase ein.

Nach erfolgreicher Verschlüsselung sollte Ihr Ergebnisfenster etwa so aussehen:



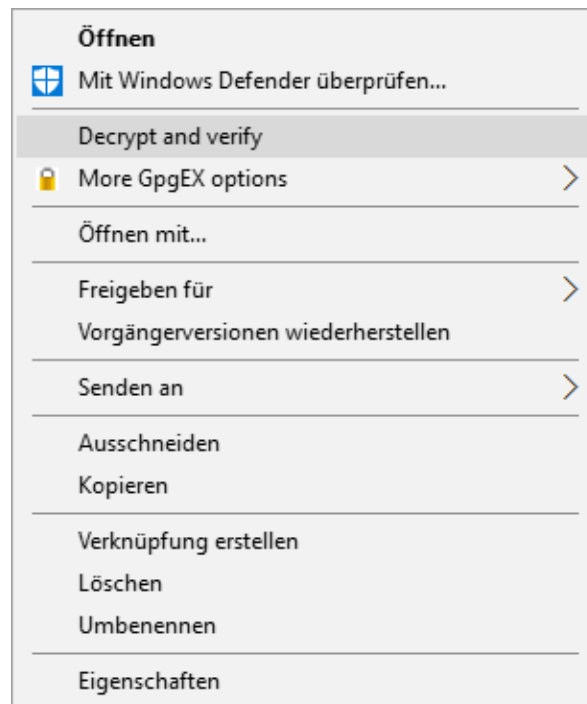
Das war's! Sie haben Ihre Datei erfolgreich verschlüsselt!

Datei entschlüsseln

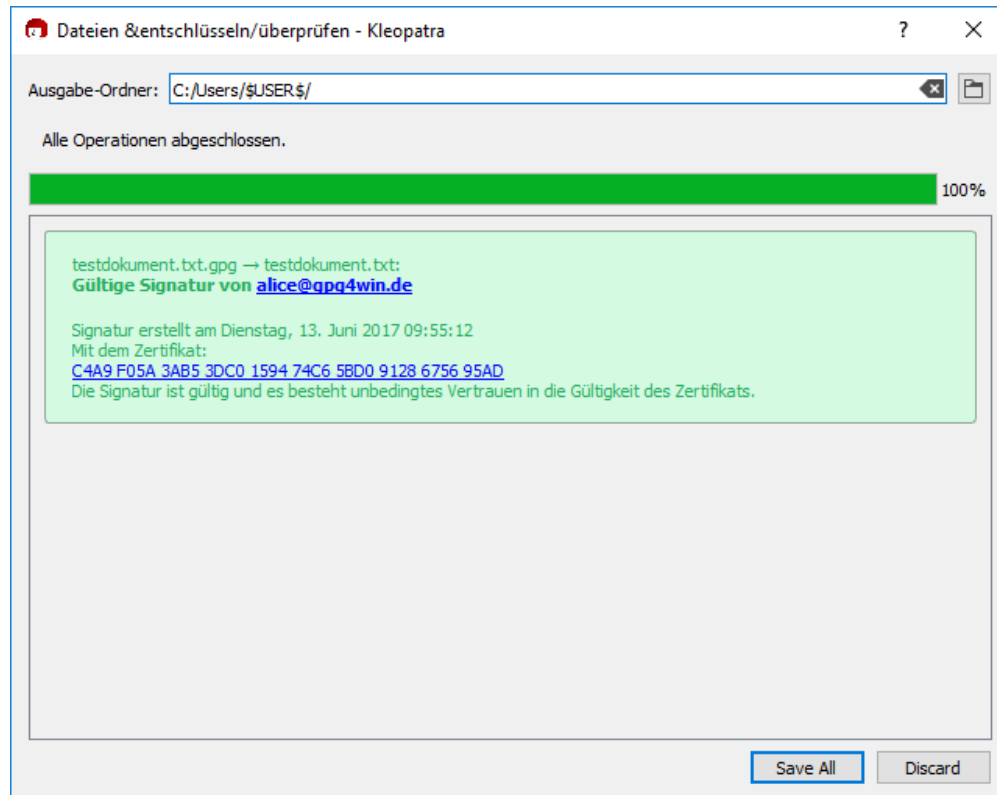
Nun kann die zuvor verschlüsselte Datei zum Testen einmal entschlüsselt werden.

Dazu sollten Sie vorher beim Verschlüsseln auch mit Ihrem eigenen Schlüssel verschlüsselt haben – andernfalls können Sie die Datei nicht mit Ihrem geheimen Schlüssel entschlüsseln.

Selektieren Sie die verschlüsselte Datei und wählen Sie im Kontextmenü des Windows-Explorers den Eintrag *Entschlüsseln und prüfen*:



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



Klicken Sie abschließend auf [*Entschlüsseln/Prüfen*].

Geben Sie anschließend Ihre Passphrase ein.

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

8.2. E-Mail-Verschlüsselung

Ihr Partner bei diesen Übungen wird **Edward** sein. Edward ist ein Testservice, der von der Free Software Foundation betrieben wird. Wir bedanken uns bei der Free Software Foundation für den Betrieb von Edward.

Der Vorgänger von Edward baut auf dem Projekt GnuPP auf und hörte auf dem Namen **Adele**. Dieser Name wird in diesem Abschnitt weiterhin verwendet.

Das Gpg4win-Team hat keinen Einfluss auf den Betrieb von Adele/Edward und kann nicht gewährleisten, dass der Testservice antwortet. Sollte es Probleme mit Adele geben, schauen Sie bitte unter <https://wiki.gnupg.org/EmailExercisesRobot> nach.

Mit Hilfe von Adele können Sie Ihr erzeugtes OpenPGP-Schlüsselpaar ausprobieren und testen.

Nachdem Sie Ihren Schlüssel erstellt haben, wollen Sie direkt loslegen. Sie können das Prozedere zunächst mit einem freundlichen E-Mail-Roboter üben. Adele soll Ihnen dabei behilflich sein. Die folgenden Übungen gelten nur für OpenPGP. Anmerkungen zum Veröffentlichen von öffentlichen X.509-Zertifikaten finden Sie auf Seite 79.

Adele ist ein sehr netter E-Mail-Roboter, mit dem Sie zwanglos korrespondieren können. Bitte beachten Sie, dass Adele eventuell nicht immer antwortet. Falls Sie nicht antwortet, üben Sie lieber mit einem Menschen. Weil man gewöhnlich mit einer klugen und netten jungen Frau lieber korrespondiert als mit einem Stück Software (was er in Wirklichkeit natürlich ist), können Sie sich Adele so vorstellen:



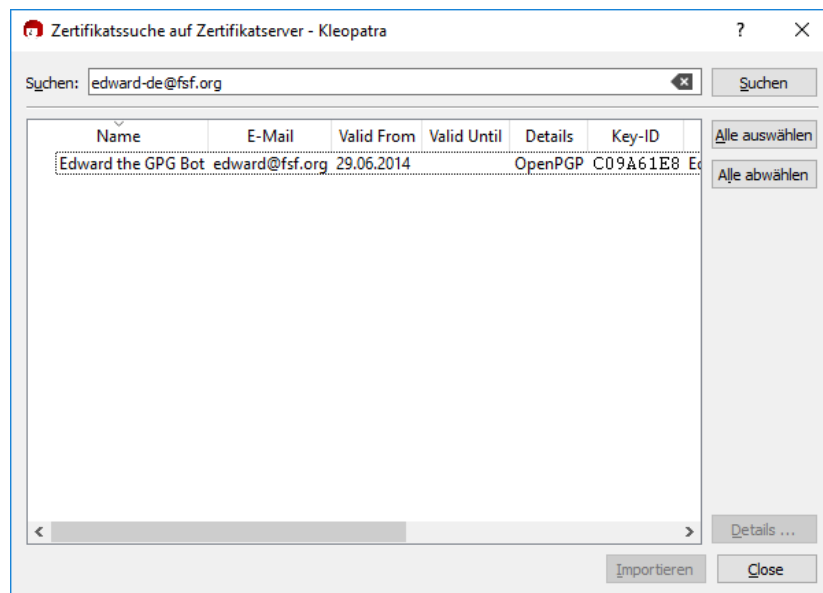
Um mit Adele zu kommunizieren, müssen Sie ihr Ihren öffentlichen Schlüssel per E-Mail schicken. Außerdem müssen Sie Adeles Schlüssel importieren, um Adele eine verschlüsselte E-Mail zu senden.

Um sichere E-Mails austauschen zu können, müssen beide Partner jeweils den öffentlichen Schlüssel des anderen besitzen und benutzen.

Um Ihren Schlüssel zu exportieren selektieren Sie in Kleopatra den öffentlichen Schlüssel (durch Klicken auf die entsprechende Zeile in der Liste der Schlüssel) und klicken Sie dann auf *Datei→Schlüssel exportieren...* im Menü. Wählen Sie einen geeigneten Dateipfad

Um Adeles Schlüssel zu erhalten, müssen Sie zunächst Adeles Schlüssel importieren, denn ohne Adeles öffentlichen Schlüssel, können Sie ihr keine verschlüsselten E-Mails senden.

Wählen Sie im Kleopatra-Fenster die Schaltfläche [*Auf Server Suchen*]. Geben Sie in die Suchfläche die E-Mail-Adresse von Adele ein (`edward-de@fsf.org`) und bestätigen Sie Ihre Suche. Unter den Suchergebnissen wählen Sie nun den Schlüssel mit den Schlüssel-ID `C09A61E8`. Abschließend wählen Sie unten im Fenster die Schaltfläche [*Importieren*].



Den Anschließendenden Dialog zur Schlüsselverifikation verlassen Sie mit *Nein*. Mehr zu diesem Thema finden Sie in Abschnitt 10.

Öffnen Sie eine neue E-Mail und füllen Sie etwas in die Betreffzeile, wie „Verschlüsselungstest“. Fügen Sie als Anhang den gerade exportierten Schlüssel hinzu und achten Sie darauf, dass das Verschlüsseln-Symbol aktiv ist. Geben Sie als Zieladresse `edward-de@fsf.org` an und senden Sie die E-Mail ab.

Nach einigen Minuten sollten Sie eine verschlüsselte Antwort von Adele erhalten. Diese Antwort von Adele entschlüsseln Sie mit Ihrem eigenen geheimen Schlüssel.

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 irgendwann ziemlich übel nehmen würde.

Herzlichen Glückwunsch! Sie haben erfolgreich verschlüsselt kommuniziert!

9. Öffentliche Schlüssel importieren

In Kapitel 8 wurde bereits kurz erläutert, wie man einen öffentlichen Schlüssel von einem Schlüsselservers importiert. Wenn Sie nun mit einer Person kommunizieren wollen, die ihren öffentlichen Schlüssel nicht auf einem Schlüsselservers hat, so müssen Sie diesen Schlüssel auf anderen Wegen importieren. In diesem Kapitel werden die beiden Möglichkeiten beschrieben, wie Sie einen Schlüssel importieren können. Falls Sie sich im speziellen für das Importieren eines geheimen Schlüssels interessieren, springen Sie direkt zu Kapitel 17.

Nach einem Schlüsselimport werden Sie immer gefragt ob und wie sie diesem Schlüssel vertrauen wollen. Diesen Dialog können Sie für Zunächst ignorieren und mit *Nein* verlassen. Auf Dieses Thema gehen wir näher im kommenden Abschnitt 10 ein.

9.1. Importieren aus Datei

Zum Importieren eines zuvor exportierten oder als E-Mail-Anhang zugesendeten öffentlichen Schlüssels, klicken Sie auf *Datei* → *Zertifikat importieren...* und wählen Sie die zu importierende Datei aus. Anschließend erhalten Sie einen Ergebnisdiallog über den erfolgten Schlüsselimport.

9.2. Importieren vom Schlüsselserver

Es gibt viele öffentliche Schlüsselserver, die alle untereinander synchronisiert sind. Diese sind zu einem Kollektiv unter der Adresse `keys.gnupg.net` erreichbar. Nach der Installation ist dieses bereits ein Schlüsselserver hinterlegt. Um einen anderen Server als Schlüsselserver hinzuzufügen, öffnen Sie das Schlüsselserver-Menü *Einstellungen* → *Kleopatra einrichten ...* und wählen Sie dort *Schlüsselserver*. Fügen Sie über die Schaltfläche *Neu* einen neuen Eintrag in die Liste hinzu. Dort erscheint nun ein vorkonfigurierter Eintrag für `keys.gnupg.net`. Diese Einstellung kann mit *OK* bestätigt werden.

Weitere Informationen zu diesem Thema finden Sie unter Kapitel 15.

Herzlichen Glückwunsch! - Sie haben erfolgreich einen Schlüsselserver eingerichtet und können nun über die Schaltfläche [*Auf Server suchen*] nach Namen oder E-Mail-Adressen von Kommunikationspartnern suchen.

10. Öffentliche Schlüssel prüfen

Woher wissen Sie eigentlich, dass der fremde (öffentliche) Schlüssel wirklich vom genannten Absender stammt? Und umgekehrt – warum sollte Ihr Korrespondenzpartner glauben, dass der öffentliche Schlüssel, den Sie ihm geschickt haben, auch wirklich von Ihnen stammt? Die Absenderangabe auf einer E-Mail besagt eigentlich gar nichts, genauso wie die Absenderangabe auf einem Briefumschlag.

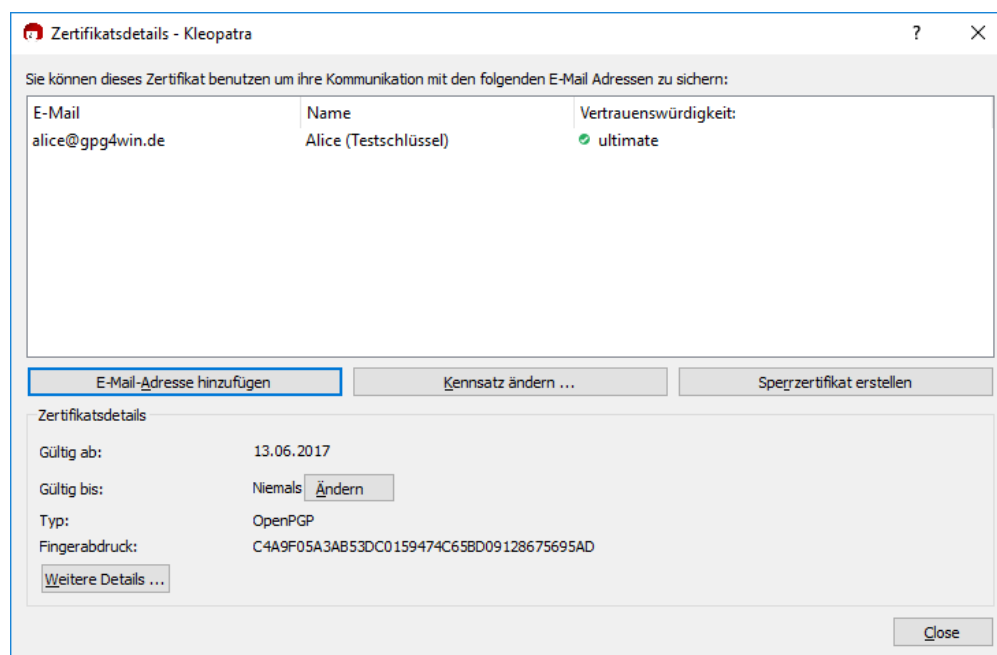
Wenn Ihre Bank z.B. eine E-Mail mit Ihrem Namen und der Anweisung erhält, Ihr sämtliches Guthaben auf ein Nummernkonto auf den Bahamas zu überweisen, wird sie sich hoffentlich weigern – E-Mail-Adresse hin oder her. Eine E-Mail-Adresse besagt überhaupt nichts über die Identität des Absenders.

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 Schlüssels.

Jeder Schlüssel trägt eine einmalige Kennzeichnung, die es zweifelsfrei identifiziert; besser noch als ein Fingerabdruck eines Menschen. Deshalb bezeichnet man diese Kennzeichnung ebenfalls als „Fingerabdruck“.

Wenn Sie sich die Details eines Schlüssels in Kleopatra anzeigen lassen, z.B. durch Doppelklick auf den Schlüssel, sehen Sie u.a. dessen 40-stelligen Fingerabdruck:



Wie gesagt – der Fingerabdruck identifiziert den Schlüssel und seinen Besitzer eindeutig.

Rufen Sie Ihren Korrespondenzpartner einfach an und lassen Sie sich von ihm den Fingerabdruck seines Schlüssels vorlesen. Wenn die Angaben mit dem Ihnen vorliegenden Schlüssel übereinstimmen, haben Sie eindeutig den richtigen Schlüssel.

Natürlich können Sie sich auch persönlich mit dem Eigentümer des Schlüssels treffen oder auf einem anderen Wege sicherstellen, dass Schlüssel und Eigentümer zusammen gehören. Häufig ist der Fingerabdruck auch auf Visitenkarten abgedruckt; wenn Sie also eine garantiert authentische Visitenkarte haben, so können Sie sich den Anruf ersparen.

OpenPGP-Schlüssel beglaubigen

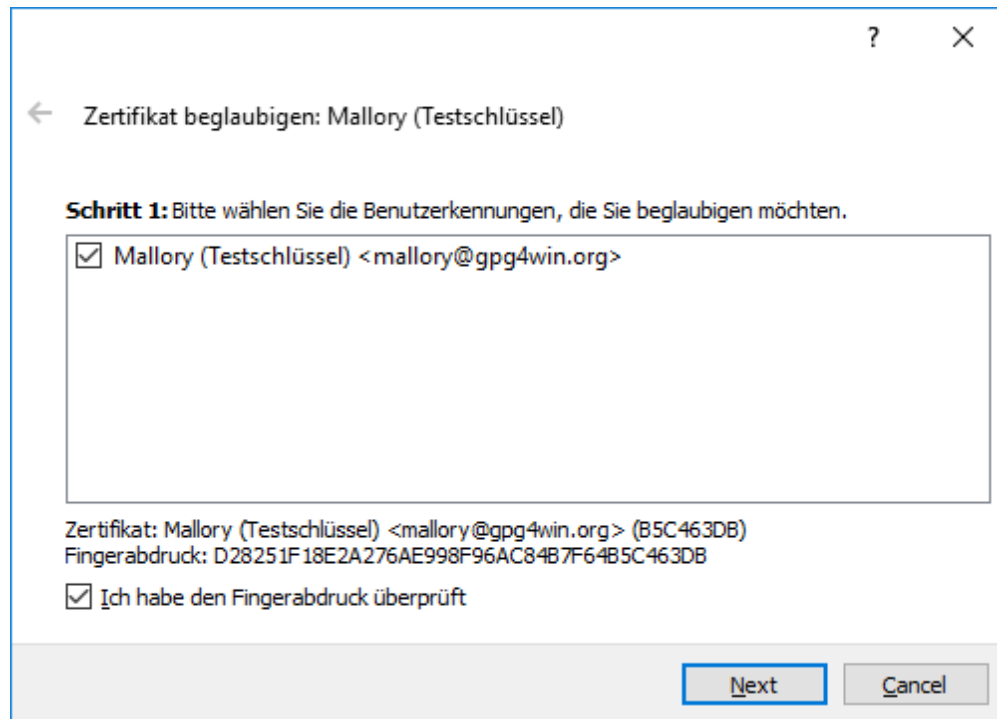
Nachdem Sie sich „per Fingerabdruck“ von der Echtheit des Schlüssels überzeugt haben, können Sie ihn beglaubigen – allerdings nur in OpenPGP. Bei X.509 können Benutzer keine Schlüssel beglaubigen – das bleibt den Beglaubigungsinstanzen (CAs) vorbehalten.

Durch das Beglaubigen eines Schlüssels teilen Sie anderen (Gpg4win-)Benutzern mit, dass Sie diesen Schlüssel für echt – also authentisch – halten: Sie übernehmen so etwas wie die „Patenschaft“ für diesen Schlüssel und erhöhen das allgemeine Vertrauen in seine Echtheit.

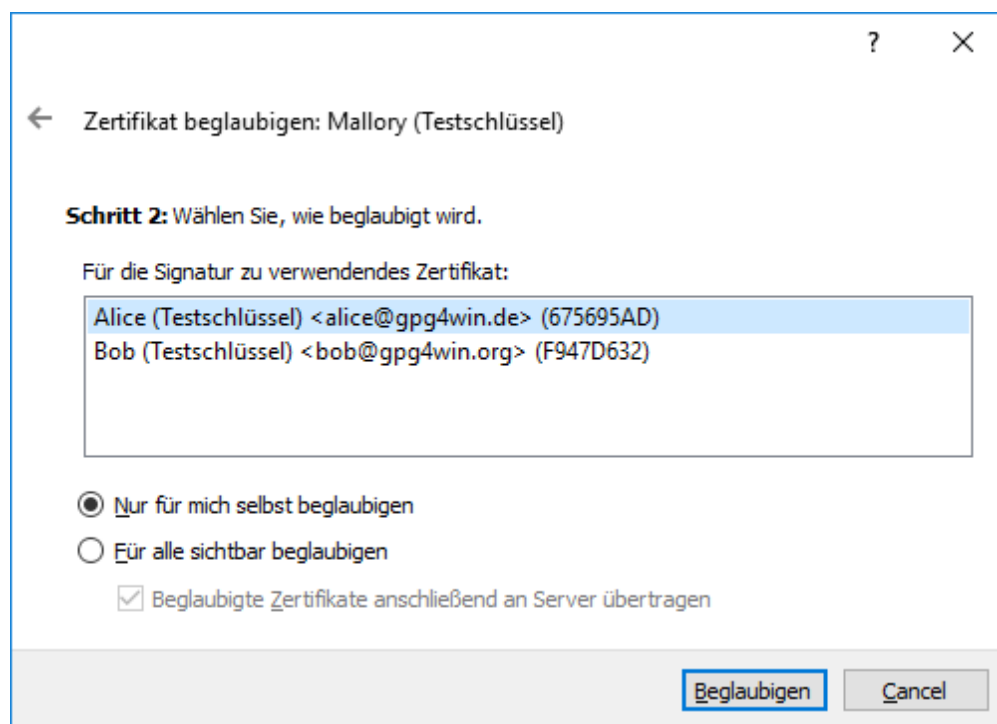
Wie funktioniert das Beglaubigen nun genau?

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

Im nachfolgenden Dialog bestätigen Sie nun noch einmal den zu beglaubigenden OpenPGP-Schlüssel und den Fingerabdruck mit [Weiter]:



Im nächsten Schritt wählen Sie Ihren eigenen OpenPGP-Schlüssel aus, mit dem Sie den im letzten Schritt ausgewählten Schlüssel beglaubigen wollen:

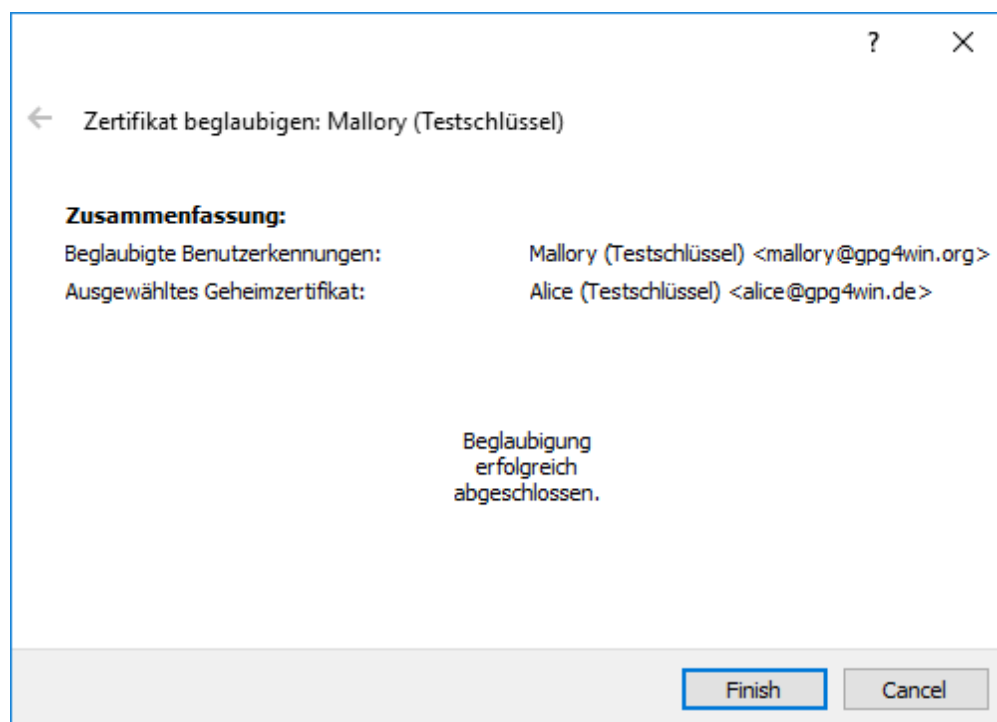


Entscheiden Sie hier, ob Sie [*Nur für mich selbst beglaubigen*] oder [*Für alle sichtbar beglaubigen*] wollen. Bei letzterer Variante haben Sie die Option, den beglaubigten Schlüssel anschließend auf einen OpenPGP-Schlüsselservers hochzuladen und damit der Welt einen mit Ihrer Beglaubigung versehenen, aktualisierten Schlüssel zur Verfügung zu stellen.

Bestätigen Sie Ihre Auswahl mit [*Beglaubigen*].

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

Nach erfolgreicher Beglaubigung erhalten Sie folgendes Fenster:



Das Netz des Vertrauens in OpenPGP

Durch das Beglaubigen von Schlüsseln entsteht – auch über den Kreis von Gpg4win-Benutzern und Ihre täglichen Korrespondenz hinaus – ein „Netz des Vertrauens“ („Web of Trust“, WoT), bei dem Sie nicht mehr zwangsläufig darauf angewiesen sind, ein OpenPGP-Schlüssel direkt auf Echtheit (Autentizität) zu prüfen.



Natürlich steigt das Vertrauen in einen Schlüssel, wenn mehrere Leute es beglaubigen. Ihren eigenen OpenPGP-Schlüssel wird im Laufe der Zeit die Beglaubigungen vieler anderer GnuPG-Benutzer tragen. Damit können immer mehr Menschen darauf vertrauen, dass dieser Schlüssel wirklich Ihnen und niemandem sonst gehört.

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

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

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

Beglaubigungsinstanzen bei X.509

Hier braucht man eine „übergeordnete“ Instanz, der alle Benutzer vertrauen können. Sie prüfen 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 Beglaubigungsinstanzen gibt es auch für OpenPGP-Schlüssel. 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 einen OpenPGP-Schlüssel erhält, der durch eine solche Beglaubigungsinstanz per Beglaubigung seine Echtheit bestätigt, sollte man sich darauf verlassen können.

Derartige Beglaubigungsinstanzen oder „Trust Center“ sind auch bei anderen Verschlüsselungsverfahren – 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 gibt, Benutzerzertifikate zu beglaubigen (vgl. Kapitel 5).



Am besten ist diese Infrastruktur mit einem Siegel vergleichbar: Die Plakette auf Ihrem Autonommerschild kann Ihnen nur eine dazu berechnigte Institution geben, die die Befugnis dazu wiederum von einer übergeordneten Stelle erhalten hat. Technisch ist eine Beglaubigung nichts anderes als eine Signatur eines Schlüssels durch den Beglaubigenden.

Die hierarchischen Beglaubigungs-Infrastrukturen entsprechen 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 Beglaubigungsstruktur (S/MIME). Demnach bietet Gpg4win eine Grundlage, um dem Signaturgesetz der Bundesrepublik Deutschland zu entsprechen.

Wenn Sie sich weiter für dieses Thema interessieren, dann können Sie sich z.B. bei folgenden Webadressen über dieses und viele andere IT-Sicherheits-Themen informieren:

- www.bsi.bund.de
- www.bsi-fuer-buerger.de
- www.gpg4win.de

Eine weitere, eher technische Informationsquelle zum Thema der Beglaubigungsinfrastrukturen bietet das GnuPG-Handbuch, das Sie ebenfalls im Internet finden unter:

www.gnupg.org/gph/de/manual

11. E-Mails signieren und verschlüsseln

Sie wissen: Normalerweise verschlüsseln Sie eine Nachricht mit Hilfe des öffentlichen Schlüssels Ihres Korrespondenzpartners, der dann mit seinem geheimen Schlüssel die E-Mail entschlüsselt.

Die umgekehrte Möglichkeit – Verschlüsselung mit dem geheimen Schlüssel – macht keinen Sinn, weil alle Welt den dazugehörigen öffentlichen Schlüssel kennt und die Nachricht damit entschlüsseln könnte.

Es gibt aber ein anderes Verfahren, um mit Ihrem geheimen Schlüssel eine Datei zu erzeugen: die Signatur.

Solch eine digitale Signatur bestätigt eindeutig die Urheberschaft – denn wenn jemand Ihren öffentlichen Schlüssel auf diese Datei (die Signatur) anwendet und diese Prüfung erfolgreich ist, so kann diese Datei nur von Ihrem privaten Schlüssel kodiert worden sein. Und zu dem dürfen ja nur Sie selbst Zugang haben.

Sie können beide Möglichkeiten kombinieren, also eine E-Mail signieren und verschlüsseln:

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

Damit hat die Botschaft sozusagen zwei Sicherheitsmerkmale:

1. Ihr Siegel auf der Nachricht: die Signatur mit Ihrem geheimen Schlüssel.
2. Einen soliden äußeren Umschlag: die Verschlüsselung mit dem öffentlichem Schlüssel des Korrespondenzpartners.

Ihr Korrespondenzpartner öffnet die äußere, starke Hülle mit seinem eigenen geheimen Schlüssel. Hiermit ist die Geheimhaltung gewährleistet, denn nur dieser Schlüssel kann den Text dekodieren. Das Siegel liest er mit Ihrem öffentlichem Schlüssel und hat den Beweis Ihrer Urheberschaft, denn wenn Ihr öffentlicher Schlüssel passt, kann das Siegel (die digitale Signatur) nur mit Ihrem geheimen Schlüssel kodiert worden sein.

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

Dieses Kapitel beschäftigt sich mit dem Signieren und Verschlüsseln von E-Mails mit dem Outlook-Plugin GpgOL.

11.1. E-Mails signieren und verschlüsseln mit GpgOL

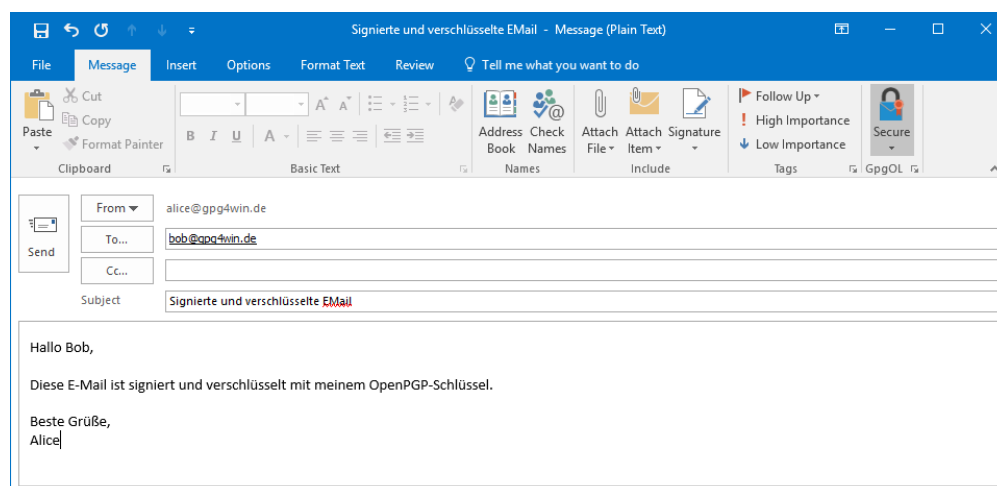
Das Signieren und Verschlüsseln einer E-Mail ist sehr einfach. Nachdem Sie eine neue E-Mail verfasst haben, gehen sie folgende Schritte durch:

- Nachricht signiert und verschlüsselt senden
- Schlüssel auswählen
- Signierung und Verschlüsselung abschließen

Auf den nächsten Seiten werden diese Schritte im Detail beschrieben.

Verfassen Sie zunächst in Outlook eine neue E-Mail und adressieren Sie diese an Ihren Korrespondenzpartner. Die Standardauswahl zum Signieren und Verschlüsseln ist bereits für Sie markiert.

Ihr E-Mail-Fenster sollte anschließend etwa so aussehen:

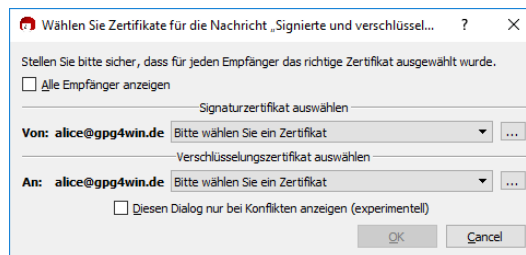


Klicken Sie nun auf [*Senden*].

Schlüsselauswahl

GpgOL erkennt automatisch, für welches Protokoll – OpenPGP oder S/MIME – Ihr eigener privater Schlüssel zum Signieren und Verschlüsseln vorliegt.

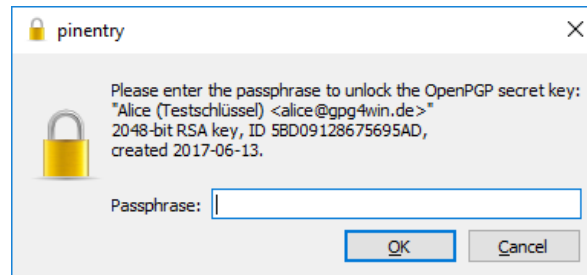
Sollten Sie gleichzeitig einen eigenen OpenPGP-Schlüssel *und* ein S/MIME-Zertifikat mit der gleichen E-Mail-Adresse besitzen, fragt Sie Kleopatra vor dem Signieren nach dem gewünschten Protokollverfahren, oder haben Sie vom gewählten Verfahren mehrere eigene Schlüssel (z.B. zwei OpenPGP-Schlüssel zu der gleichen E-Mail-Adresse), dann öffnet Kleopatra ein Fenster, in dem Ihre eigenen Schlüssel angezeigt werden, zu denen Ihnen jeweils ein geheimer Schlüssel vorliegt:



Bestätigen Sie Ihre Auswahl anschließend mit [OK].

Signierung und Verschlüsselung abschließen

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



Dies ist notwendig, denn Sie wissen:

Signieren können Sie nur mit Ihrem eigenen geheimen Schlüssel.

Logisch, denn nur Ihr geheimer Schlüssel bestätigt Ihre Identität. Der Korrespondenzpartner kann dann mit Ihrem öffentlichen Schlüssel, den er bereits hat oder sich besorgen kann, Ihre Identität prüfen und die E-Mail entschlüsseln. Denn nur Ihr geheimer Schlüssel passt zu Ihrem öffentlichem Schlüssel.

Bestätigen Sie Ihre Passphrase-Eingabe mit [*OK*]. Ihre Nachricht wird nun signiert, verschlüsselt und versendet.

Ihnen wird vielleicht schon aufgefallen sein, dass Sie nicht jedes mal Ihre Passphrase erneut eingeben müssen. Sie wird im Hintergrund für einige Zeit gespeichert.

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

Übrigens: Sie können eine Nachricht auch nur signieren oder nur verschlüsseln. Klicken Sie dazu auf das GpgOL Symbol in der Nachricht. Der Ablauf ist analog zu dem hier beschriebenen. Es wird aber davon abgeraten, eine Nachricht nur zu verschlüsseln, da hierbei nicht die Urheberschaft der Nachricht geschützt wird.

11.1.1. Signatur prüfen mit GpgOL

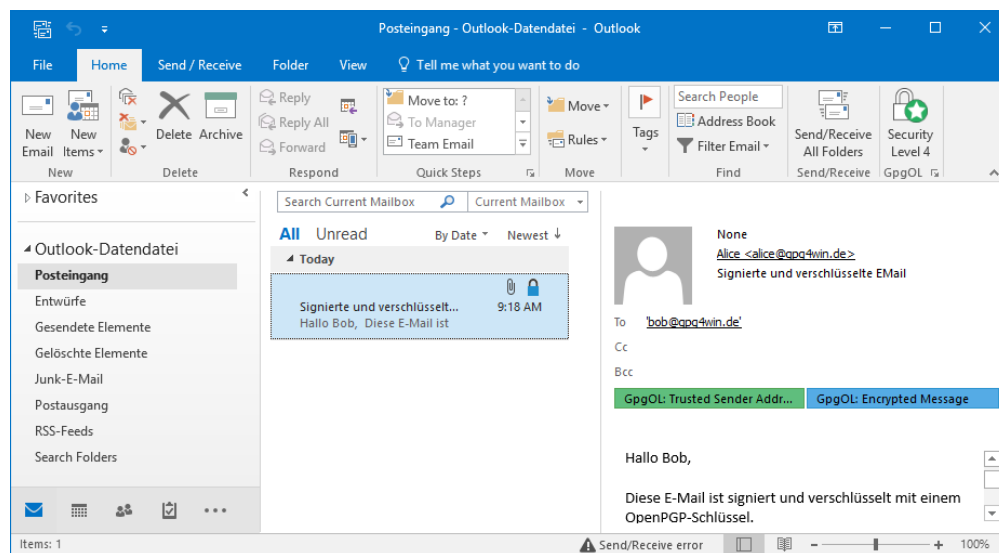
Angenommen, Sie erhalten eine signierte E-Mail Ihres Korrespondenzpartners.

Die Überprüfung dieser digitalen Signatur ist sehr einfach. Alles, was Sie dazu brauchen, ist der öffentliche Schlüssel Ihres Korrespondenzpartners, den Sie vor der Überprüfung in Ihre Schlüsselverwaltung importiert haben sollten (vgl. Kapitel 9.1).

Um eine signierte E-Mail zu prüfen, gehen Sie wie folgt vor:

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!

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

Sollte die Signaturprüfung fehlschlagen, 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. Allerdings muss eine gebrochene Signatur nicht zwangsläufig bedeuten, dass die E-Mail manipuliert wurde. Es ist ebenfalls nicht auszuschließen, dass die E-Mail durch eine fehlerhafte Übertragung verändert wurde. Nehmen Sie in jedem Fall eine gebrochene Signatur ernst und fordern Sie immer die E-Mail erneut beim Absender an!

11.2. E-Mails signieren

Sie haben in Kapitel 10 gelesen, wie Sie sich von der Echtheit eines öffentlichen Schlüssels überzeugen und es dann mit Ihrem eigenen geheimen Schlüssel signieren können.

Dieser Abschnitt beschäftigt sich damit, wie Sie nicht nur einen Schlüssel, sondern auch eine komplette **E-Mail signieren** können. Das bedeutet, dass Sie die E-Mail mit einer digitalen Signatur versehen – einer Art elektronischem Siegel.

So „versiegelt“ ist der Text dann zwar noch für jeden lesbar, aber der 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 öffentlichen Schlüssel Sie nicht haben (aus welchem Grund auch immer), können Sie so die Nachricht wenigstens mit Ihrem eigenen privaten Schlüssel „versiegeln“.

Sie haben sicher bemerkt, dass diese digitale Signatur nicht mit der E-Mail-„Signatur“ identisch ist, die man manchmal unter eine E-Mail setzt und die z.B. Telefonnummer, Adresse und Webseite nennt. Während diese E-Mail-Signaturen einfach nur als eine Art Visitenkarte fungieren, schützt die digitale Signatur Ihre E-Mail vor Manipulationen und bestätigt den Absender eindeutig.

Übrigens ist die digitale Signatur auch nicht mit der qualifizierten elektronischen 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.



11.3. E-Mails verschlüsselt archivieren

Ihre wichtigen – und daher möglicherweise verschlüsselten – E-Mails sollten Sie auch so archivieren: verschlüsselt.

Natürlich können Sie einfach eine Klartext-Fassung 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 brauchen Sie aber den geheimen Schlüssel des Empfängers – und den haben Sie nicht und werden ihn nie haben ...

Also was tun?

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

Die Nachricht wird einmal für Ihren eigentlichen Korrespondenzpartner verschlüsselt und ein zweites Mal auch für Sie selbst (mit Hilfe Ihres öffentlichen Schlüssels). So können Sie die E-Mail später einfach mit Ihrem eigenen geheimen Schlüssel wieder lesbar machen.

Jede verschlüsselte Nachricht wird von GpgOL automatisch auch an Ihren eigenen öffentlichen Schlüssel verschlüsselt. Dazu nutzt GpgOL Ihre Absender-E-Mail-Adresse. Sollten Sie mehrere Schlüssel zu einer Adresse besitzen, so müssen Sie sich beim Verschlüsselungsvorgang entscheiden, mit welchem Schlüssel verschlüsselt werden soll.

Kurz zusammengefasst

Sie haben gelernt, wie Sie eine E-Mail mit Ihrem geheimen Schlüssel **signieren** – und mit dem öffentlichen Schlüssel Ihres Korrespondenzpartners **verschlüsseln**.

Damit beherrschen Sie nun die beiden wichtigsten Techniken für einen sicheren E-Mail-Versand: signieren und verschlüsseln. Sie sollten beide Techniken stets kombinieren. Sie können aber bei jeder neuen E-Mail entscheiden, wie Sie Ihre Nachricht versenden wollen – je nachdem, wie wichtig und schutzbedürftig der Inhalt Ihrer E-Mail ist.

Zusätzlich sollte jede verschlüsselte E-Mail auch zusätzlich an Sie selbst verschlüsselt sein. Dafür sorgt GpgOL automatisch.

12. 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 Hilfe Ihres geheimen Schlüssels, um sicherzugehen, dass die Datei unverändert bleibt.
- Sie **verschlüsseln** eine Datei mit Hilfe eines öffentlichen Schlüssels, um die Datei vor unbefugten Personen geheim zu halten.

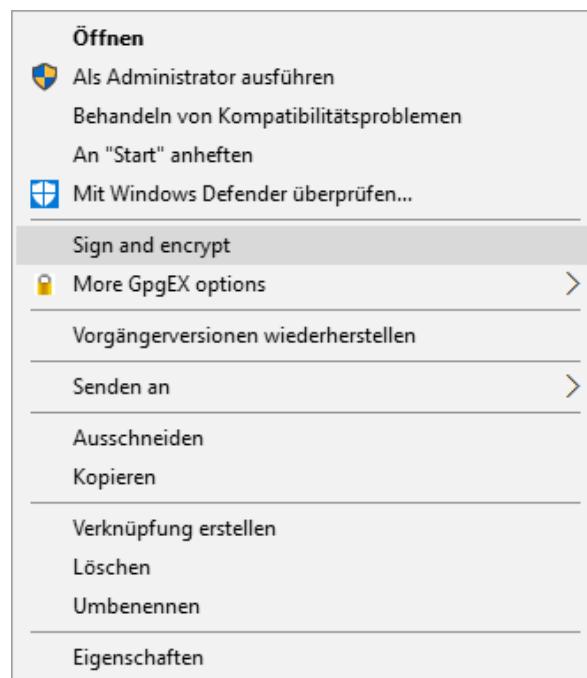
Mit der Gpg4win-Programmkomponente **GpgEX** können Sie Dateien ganz einfach aus dem Windows-Explorer heraus signieren oder verschlüsseln – egal, ob mit OpenPGP oder S/MIME. Dieses Kapitel erläutert Ihnen, wie das genau funktioniert.

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 zu kümmern.

12.1. Dateien signieren, verschlüsseln und prüfen

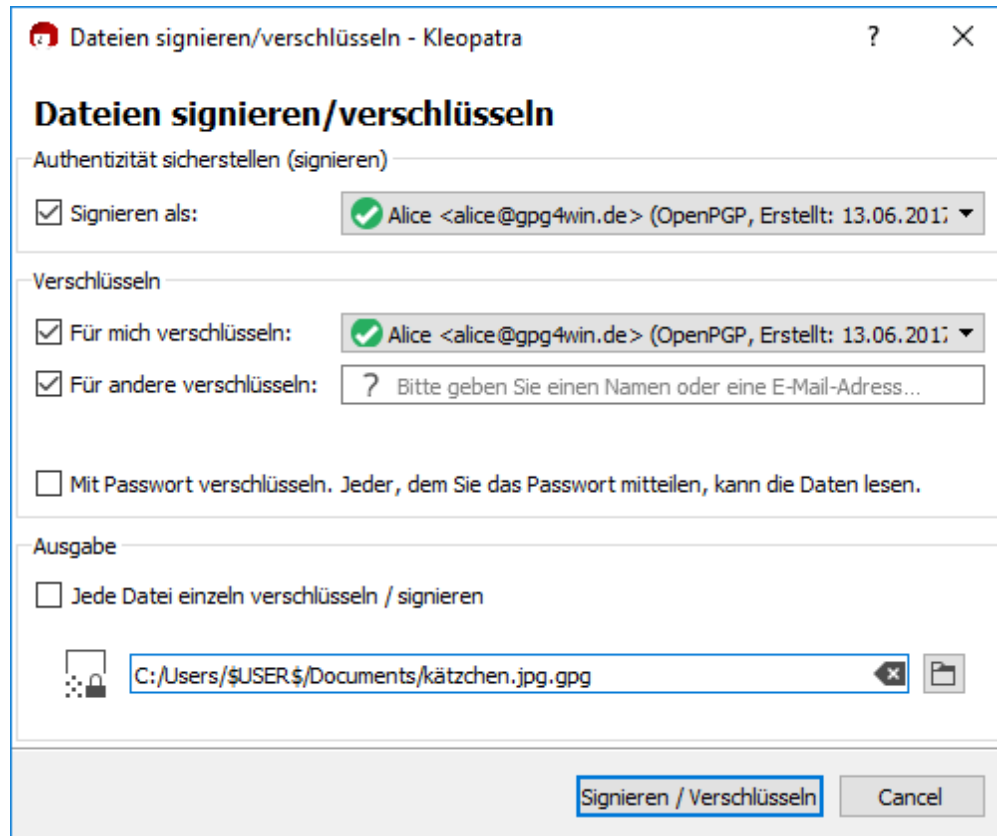
Beim Signieren einer Datei kommt es vorrangig nicht auf die Geheimhaltung, sondern auf die Unverändertheit (Integrität) der Datei an.

Die Signierung können Sie bequem mit **GpgEX** aus dem Kontextmenü des Windows-Explorers ausführen. Selektieren Sie eine (oder mehrere) Datei(en) oder Ordner und öffnen Sie mit der rechten Maustaste das Kontextmenü:



Dort wählen Sie *Signieren* und *Für mich verschlüsseln* aus. In beiden Menüpunkten sollte ihr eigener Schlüssel ausgewählt sein. Unter dem Punkt *Für andere verschlüsseln* können Sie auch andere Schlüssel angeben.

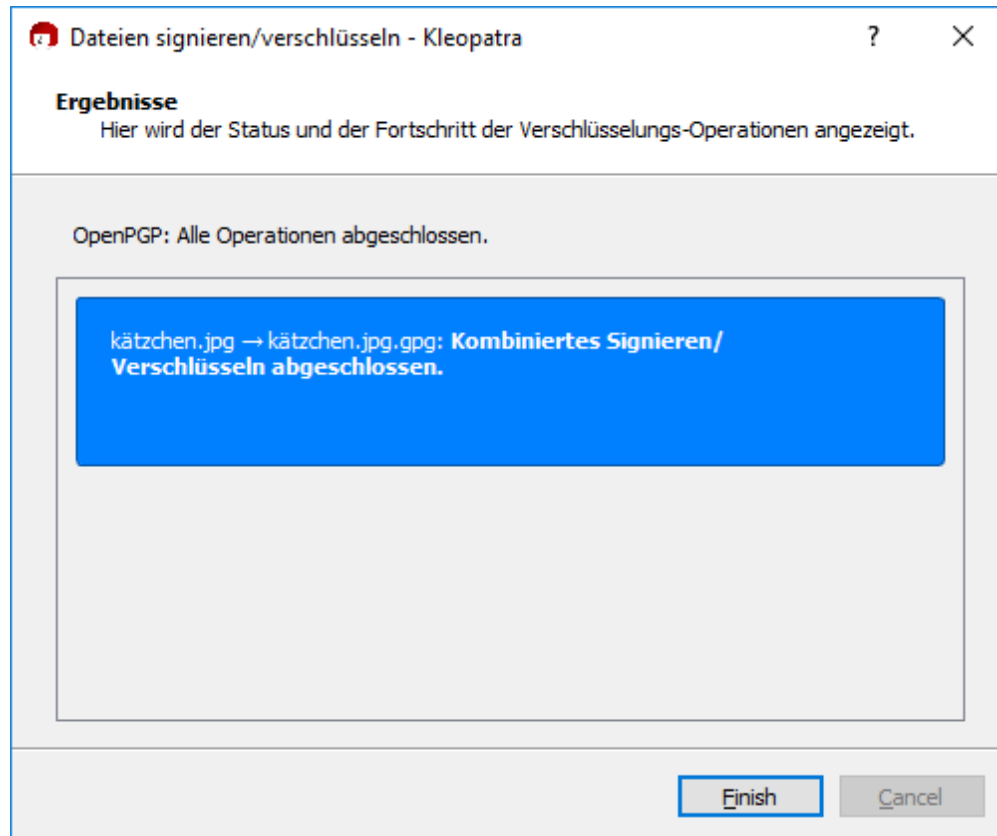
Selektieren Sie im erscheinenden Fenster die Option *Signieren*:



Klicken Sie anschließend auf [Weiter].

Geben Sie nun Ihre Passphrase in den Pinentry-Dialog ein.

Nach erfolgreicher Signierung und Verschlüsselung erhalten Sie folgendes Fenster:



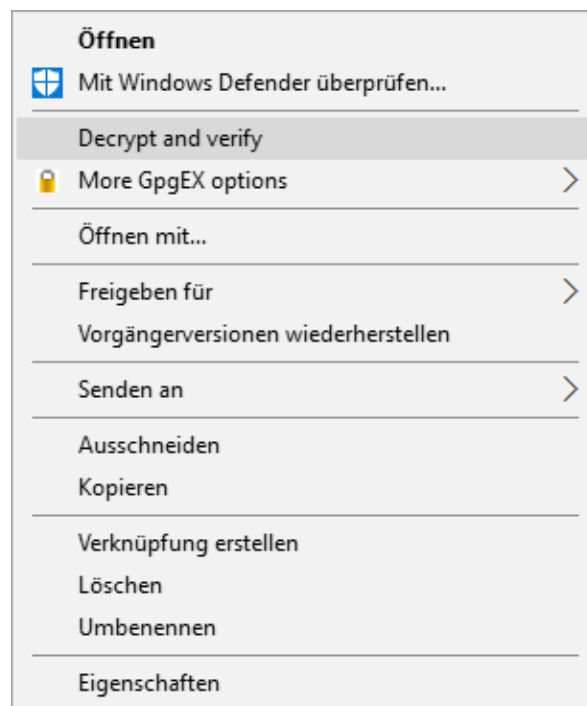
Sie haben damit Ihre Datei erfolgreich signiert und verschlüsselt.

Beim Signieren einer Datei wird stets eine „abgetrennte“ (separate) 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 prüfen, sind beide Dateien notwendig.

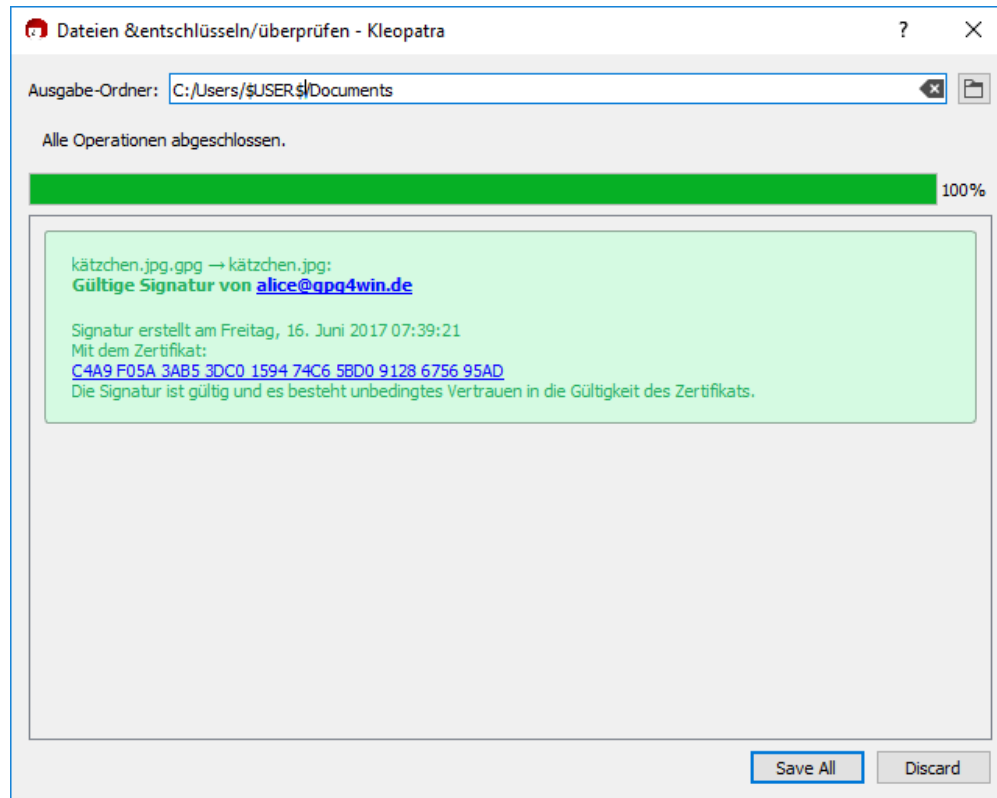
Signatur prüfen

Prüfen Sie nun, ob die eben signierte Datei integer – d.h. korrekt – ist!

Zum Überprüfen der Unverändertheit (Integrität) und der Authentizität müssen die Signatur-Datei – also die mit der Endung `.sig`, `.asc`, `.p7s` oder `.pem` – und die signierte Originaldatei (Originaldatei) in demselben Dateiordner liegen. Selektieren Sie die Signatur-Datei und wählen Sie aus dem Kontextmenü des Windows-Explorers den Eintrag *Entschlüsseln und prüfen*:



Daraufhin erhalten Sie folgendes Fenster:



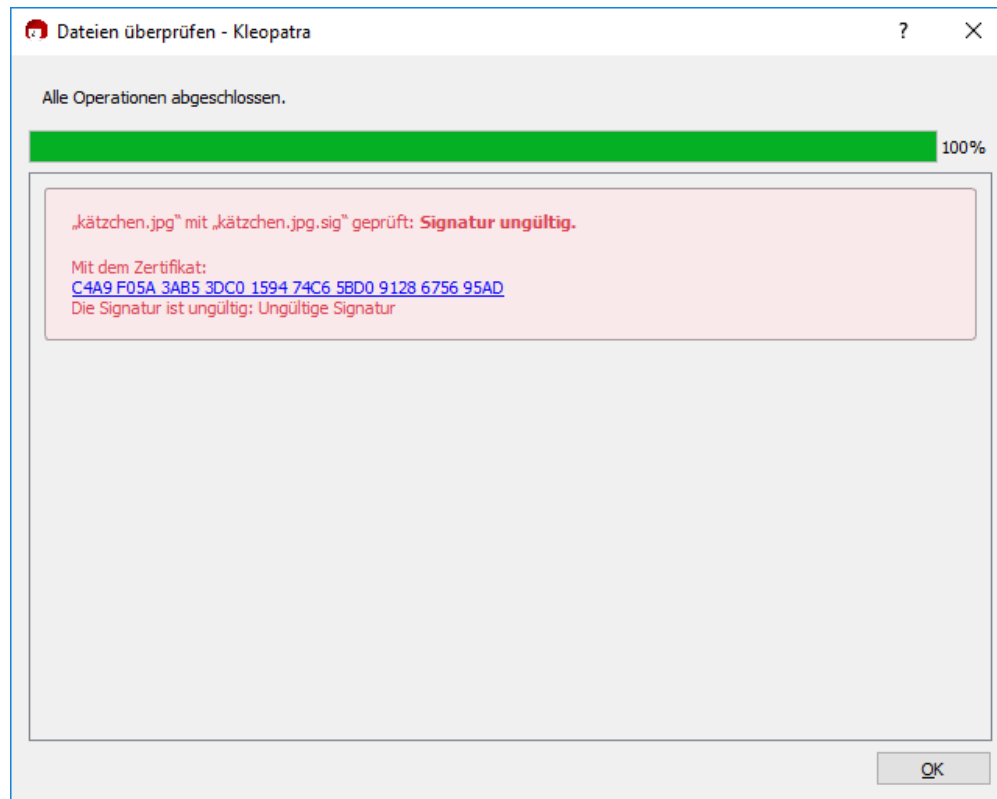
Kleopatra zeigt unter *Ausgabe-Ordner* den vollständigen Pfad zur ausgewählten Signatur-Datei an. Dieser kann angepasst werden.

Das Ergebnis zeigt, dass die Signatur korrekt ist – also die Datei integer ist und somit **nicht** verändert wurde.

Automatisch ist auch für den *Ausgabe-Ordner* der gleiche Pfad ausgewählt. Dieser wird aber erst relevant, wenn Sie mehr als eine Datei gleichzeitig verarbeiten.

Die Signaturprüfung wird schon angezeigt und über [*Speichern*] können Sie die entschlüsselte und geprüfte Datei sichern.

Selbst wenn nur ein Zeichen in der Originaldatei hinzugefügt, gelöscht oder geändert wurde, wird die Signatur als gebrochen angezeigt (Kleopatra stellt das Ergebnis als rote Warnung dar):



Kurz zusammengefasst

Sie haben gelernt, wie Sie mit GpgEX:

- Dateien signieren
- signierte Dateien prüfen
- Dateien verschlüsseln
- verschlüsselte Dateien entschlüsseln

Gleichzeitig signieren und verschlüsseln

Diese Option ist Ihnen sicher schon in den entsprechenden Dialogen aufgefallen. Wählen Sie sie aus, dann kombiniert GpgEX beide Krypto-Operationen in einem Schritt.

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

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

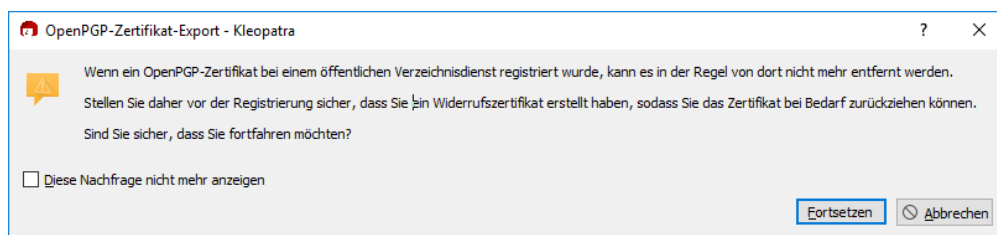
13. Öffentliche Schlüssel veröffentlichen

Um Ihren Schlüssel anderen Korrespondenzpartnern zur Verfügung zu stellen, müssen Sie Ihren öffentlichen Schlüssel verbreiten. Sie haben in Kapitel 8.2 bereits gelesen, wie man einen öffentlichen Schlüssel exportiert und ihn versendet, in diesem Abschnitt gehen wir nun auf die Veröffentlichung Ihres öffentlichen Schlüssels auf einem Schlüsselserver ein.

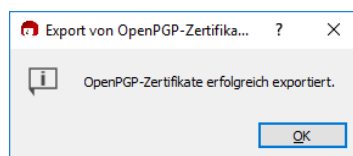
Mehr zu diesem Thema finden Sie unter Abschnitt 15.

13.1. Veröffentlichen auf OpenPGP-Schlüsselservern

Wählen Sie in der Schlüsselübersicht in Kleopatra Ihren Schlüssel aus. Über *Datei* → *Auf Server veröffentlichen..* öffnet sich folgender Dialog



Lesen Sie die Meldung aufmerksam und bestätigen Sie den Dialog. Wenn der Schlüssel erfolgreich auf den Schlüsselserver exportiert wurde. Erhalten Sie folgende Nachricht:



Ihr Schlüssel ist nun erfolgreich auf dem Schlüsselserver veröffentlicht und Korrespondenzpartner können ihn einfach über die Schlüsselsuche finden.

13.2. Veröffentlichen von X.509-Zertifikaten

Für das Veröffentlichen von X.509-Zertifikaten ist der Anbieter zuständig. Wenn Sie Ihre Zertifikatsanfrage zum Anbieter geschickt haben und ein gültiges Zertifikat erhalten haben, ist dies im Normalfall über den Schlüsselserver des Anbieters verfügbar. Dies ist jedoch von Anbieter zu Anbieter unterschiedlich. Wie man einen Schlüsselsever für X.509 Zertifikate hinzufügt, wird in Abschnitt 15 besprochen.

Teil II.

Für Fortgeschrittene

14. Schlüssel im Detail

In Kapitel 7 haben Sie sich schon den Detaildialog Ihres erzeugten Schlüsselpaares angesehen. Viele Angaben zu Ihrem Schlüsselpaar sind dort aufgelistet. Nachfolgend die Wichtigsten, mit kurzen Hinweisen auf die Unterschiede zwischen OpenPGP-Schlüsseln und X.509-Zertifikaten:

Die Benutzerkennung besteht aus dem Namen und der E-Mail-Adresse, die Sie während der Schlüsselpaarerzeugung eingegeben haben, also z.B.: Alice <alice@gpg4win.de>

Für OpenPGP-Schlüssel können Sie mit Kleopatra über den Menüpunkt *Zertifikate* → *Benutzerkennung hinzufügen...* Ihr Schlüsselpaar um weitere Benutzerkennungen erweitern. Das ist dann sinnvoll, wenn Sie z.B. für eine weitere E-Mail-Adresse den Schlüssel nutzen möchten.

Beachten Sie: Hinzufügen neuer Benutzerkennungen ist in Kleopatra nur für OpenPGP-Schlüssel möglich, nicht aber für X.509-Zertifikate.

Der Fingerabdruck wird verwendet, um mehrere Schlüssel voneinander zu unterscheiden. Mit dieser Kennung können Sie nach (öffentlichen) Schlüsseln suchen, die z.B. auf einem weltweit verfügbaren OpenPGP-Schlüsselserver (engl. „key server“) oder auf einem X.509-Zertifikats-server liegen. Was Schlüsselserver sind, erfahren Sie im folgenden Kapitel.

Die Schlüssel-ID (auch Schlüsselkennung genannt) besteht aus den letzten acht Stellen des Fingerabdrucks. Die wesentlich geringere Länge macht die Schlüsselkennung einfacher handhabbar, erhöht aber das Risiko von Mehrdeutigkeiten (unterschiedliche Schlüssel mit derselben Kennung).

Die Gültigkeit von Schlüsseln bezeichnet die Dauer ihrer Gültigkeit und ggf. ihr Verfallsdatum.

Für OpenPGP-Schlüssel ist die Gültigkeit normalerweise auf *Unbegrenzt* gesetzt. Sie können dies mit Kleopatra ändern, indem Sie auf die Schaltfläche [*Ablaufdatum ändern*] in den Details klicken – oder das Menü *Zertifikate* → *Ablaufdatum ändern* auswählen – und ein neues Datum eintragen. Damit können Sie Schlüssel für eine begrenzte Zeit gültig erklären, z.B. um sie an externe Mitarbeiter auszugeben.

Die Gültigkeitsdauer von X.509-Zertifikaten wird bei der Ausstellung von der Beglaubigungsinstanz (CA) festgelegt und kann nicht vom Nutzer geändert werden.

Das Vertrauen in den Schlüsselinhaber (nur OpenPGP) beziffert Ihre eigene, subjektive Zuversicht, dass der Besitzer des OpenPGP-Schlüssels echt (authentisch) ist und auch andere OpenPGP-Schlüssel korrekt beglaubigen wird. Sie können das Vertrauen über die Schaltfläche [*Vertrauen in den Schlüsselinhaber ändern*] in den Details oder über das Menü *Zertifikate* → *Vertrauensstatus ändern* einstellen.

Der Vertrauensstatus ist nur für OpenPGP-Schlüssel relevant. Für X.509-Zertifikate gibt es diese Methode der Vertrauensstellung nicht.

Die Beglaubigungen (nur OpenPGP) Ihres OpenPGP-Schlüssels beinhalten die Benutzerkennungen derjenigen Schlüsselinhaber, die sich von der Echtheit Ihres Schlüssels überzeugt und es

dann auch beglaubigt haben. Das Vertrauen in die Echtheit Ihres Schlüssels steigt mit der Anzahl an Beglaubigungen, die Sie von anderen Nutzern erhalten.

Beglaubigungen sind nur für OpenPGP-Schlüssel relevant. Für X.509-Zertifikate gibt es diese Methode der Vertrauensstellung nicht.

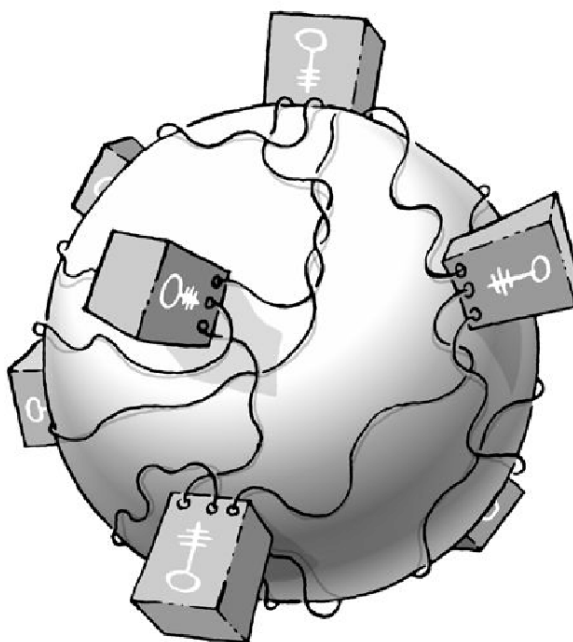
Diese Details müssen Sie für die tagtägliche Benutzung von Gpg4win nicht unbedingt kennen, aber sie werden relevant, wenn Sie neue Schlüssel erhalten oder ändern wollen.

15. Die Schlüsselservers

Die Nutzung eines Schlüsselservers zum Verbreiten Ihres öffentlichen (OpenPGP- oder X.509-) Schlüssel wurde bereits im Abschnitt 13.1 einführend erläutert. Dieses Kapitel beschäftigt sich mit den Details von Schlüsselserversn und zeigt Ihnen, wie Sie diese mit Kleopatra nutzen können.

Schlüsselservers können von allen Programmen benutzt werden, die die Standards OpenPGP bzw. X.509 unterstützen. Kleopatra unterstützt beide Arten, also sowohl OpenPGP-Schlüsselservers als auch X.509-Zertifikatsservers.

OpenPGP-Schlüsselservers (im Englischen auch „key server“ genannt) sind dezentral organisiert und synchronisieren sich weltweit miteinander. Aktuelle Statistiken über ihre Zahl oder die Anzahl der dort liegenden OpenPGP-Schlüssel gibt es nicht. Dieses verteilte Netz von OpenPGP-Schlüsselservers sorgt für eine bessere Verfügbarkeit und verhindert, dass einzelne Systemadministratoren Schlüssel löschen, um so die sichere Kommunikation unmöglich zu machen („Denial of Service“-Angriff).



X.509-Zertifikatsservers werden in der Regel von den Beglaubigungsinstanzen (CAs) über LDAP bereitgestellt und manchmal auch als Verzeichnisdienste für X.509-Zertifikate bezeichnet.



15.1. Schlüsselserver einrichten

Öffnen Sie den Konfigurationsdialog von Kleopatra:

Einstellungen → *Kleopatra einrichten...*

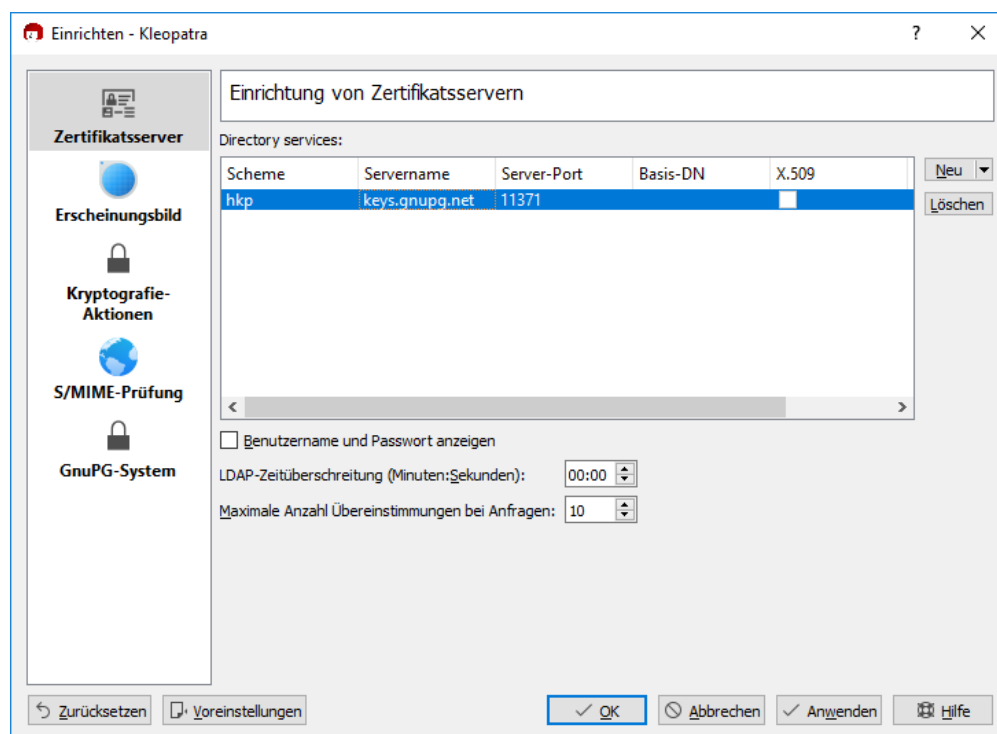
Legen Sie unter der Gruppe *Schlüsselserver* einen neuen Schlüsselserver an, indem Sie auf die Schaltfläche *Neu* klicken. Wählen Sie zwischen *OpenPGP* oder *X.509*.

Bei *OpenPGP* wird in die Liste ein voreingestellter OpenPGP-Schlüsselserver mit der Serveradresse `hkp://keys.gnupg.net` (Port: 11371, Protokoll: hkp) hinzugefügt. Sie können diesen ohne Änderung direkt verwenden – oder Sie nutzen eine der vorgeschlagenen OpenPGP-Serveradressen von der nächsten Seite.

Bei *X.509* erhalten Sie folgende Vorbelegungen für einen X.509-Zertifikatsserver: (Protokoll: ldap, Servername: server, Server-Port: 389). Vervollständigen Sie die Angaben zu Servername und Basis-DN Ihres X.509-Zertifikatsservers und prüfen Sie den Server-Port.

Sollte Ihr Schlüsselserver Benutzername und Passwort fordern, so aktivieren Sie die Option *Benutzerauthentisierung notwendig* und tragen Ihre gewünschten Angaben ein.

Der folgende Screenshot zeigt einen konfigurierten OpenPGP-Schlüsselserver:



Bestätigen Sie abschließend Ihre Konfiguration mit [*OK*]. Ihr Schlüsselserver ist nun erfolgreich eingerichtet.

Um sicherzugehen, dass Sie den Schlüsselserver korrekt konfiguriert haben, ist es hilfreich, z.B. eine Schlüsselserver auf dem Server zu starten (Anleitung siehe Abschnitt 15.3).

Proxy-Einstellung: Falls Sie einen Proxy in Ihrem Netzwerk nutzen, müssen Sie die Schlüsselserver-Adresse in der Datei:

`%APPDATA%\gnupg\gpg.conf`

ergänzen. Fügen Sie dazu in der Datei eine weitere Zeile ein, mit dem Inhalt:

`keyserver-options http-proxy=<proxy-address>`

Erläuterungen zur systemweiten Konfiguration von X.509-Zertifikatsservern finden Sie im Abschnitt 21.5.

OpenPGP-Schlüsselserver-Adressen

Es wird empfohlen, nur moderne OpenPGP-Schlüsselserver zu verwenden, da nur diese mit den neueren Merkmalen von OpenPGP umgehen können.

Hier eine Auswahl von gut funktionierenden Schlüsselserver:

- `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://keyserver.pramberger.at`
- `http://gpg-keyserver.de`

Sollten Sie Probleme mit einer Firewall haben, so versuchen Sie es am besten mit Schlüsselserver, deren URL mit `http://` beginnen.

Die Schlüsselserver unter den Adressen

- `hkp://keys.gnupg.net` (Vorauswahl von Kleopatra, siehe Bildschirmfoto auf vorheriger Seite)
- `hkp://subkeys.gpg.net`

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

Achtung: Nicht `ldap://keyserver.gpg.com` als Schlüsselserver benutzen, weil dieser sich nicht mit den anderen Servern synchronisiert (Stand: Mai 2010).

15.2. X.509 Schlüsselserver einrichten

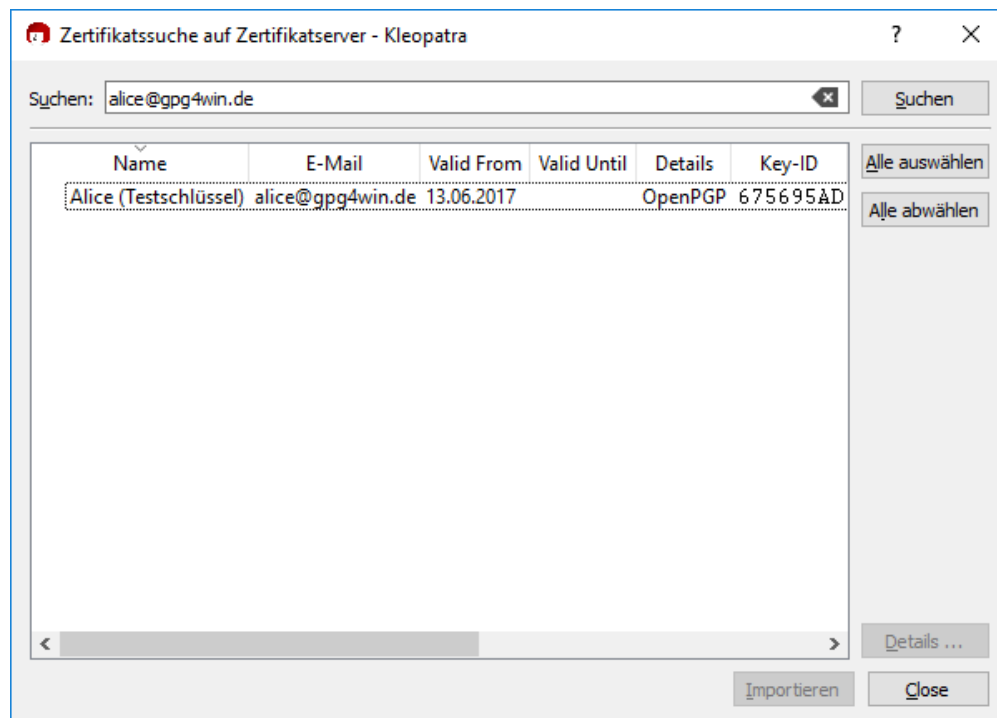
Wenn Sie einen S/MIME Schlüssel importieren wollen, müssen Sie sich zunächst über die Adresse des Schlüsselserver und der Basis-DN (Basis Domain-Name) informieren. Diese erhalten Sie von Ihrem System-Administrator oder Kommunikations-Partner. Sobald Sie diese Informationen haben, öffnen Sie unter *Einstellungen* → *Kleopatra einrichten ...* das *Schlüsselserver*-Menü. Dort fügen Sie mit der Schaltfläche *Neu* einen neuen Eintrag der Liste hinzu. Sie können die vorausgewählten Informationen durch Ihre ersetzen. Wenn Sie z.B. Die Informationen erhalten haben, dass der Schlüsselserver die URL `ldap://ca.gnupg.org:389/` und die Basis-DN `o="GnuPG", C=DE` besitzt, dann wählen Sie als *Protokoll* „ldap“ aus, tragen unter *Server* „ca.gnupg.org“ ein, unter *Port* tragen Sie „389“ ein, bei *Basis-DN* tragen Sie „o="GnuPG", C=DE“ ein und setzen am Ende einen Haken unter der *X.509-Option*. Bestätigen Sie abschließend Ihren Eintrag mit *OK*.

15.3. Schlüssel auf Schlüsselservern suchen und importieren

Nachdem Sie mindestens einen Schlüsselserver eingerichtet haben, können Sie nun dort nach Schlüsseln suchen und diese anschließend importieren.

Klicken Sie dazu in Kleopatra auf *Datei*→*Auf Server suchen...*

Sie erhalten einen Suchdialog, in dessen Eingabefeld Sie den Namen des Schlüsselbesitzers – oder eindeutiger und daher besser geeignet – seine E-Mail-Adresse seines Schlüssels eingeben können.



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

Wenn Sie nun eines der gefundenen Schlüssel in Ihre lokale Schlüsselsammlung einfügen möchten, selektieren Sie das Schlüssel 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 nun das ausgewählte Schlüssel in der Zertifikatsverwaltung von Kleopatra.

16. Dateianhänge verschlüsseln

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

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

GpgOL übernimmt die Verschlüsselung und Signierung von Anhängen automatisch.

Bei weniger komfortabel in einem E-Mail-Programm integriertem Verschlüsselungswerkzeugen 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. Dies läuft also auf ein ganz gewöhnliches Verschlüsseln von Dateien hinaus, das in Kapitel 12 beschrieben ist.

17. Im- und Export eines geheimen Schlüssels

In den Kapiteln 13 und 9.1 wurde der Im- und Export von Schlüsseln erläutert. Sie haben Ihren eigenen Schlüssel exportiert, um ihn zu veröffentlichen, und den Schlüssel Ihres Korrespondenzpartners importiert und so „an Ihrem Schlüsselbund befestigt“ (d.h. in Ihre Schlüsselverwaltung aufgenommen).

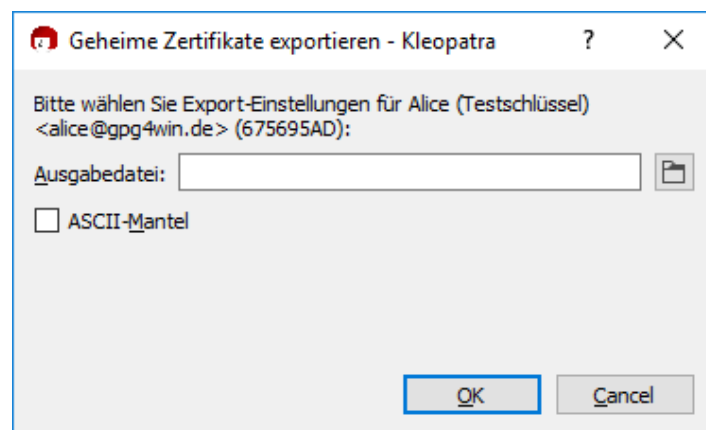
Dabei ging es stets um **öffentliche** Schlüssel. Es gibt aber auch hin und wieder die Notwendigkeit, einen **geheimen** Schlüssel zu im- oder exportieren. Wenn Sie z.B. ein bereits vorhandenes (OpenPGP oder S/MIME) Schlüsselpaar mit Gpg4win weiterbenutzen wollen, müssen Sie es importieren. Oder wenn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

17.1. 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.

Eine solche Sicherungskopie haben Sie evtl. schon einmal am Ende Ihrer OpenPGP-Schlüsselpaar-erzeugung angelegt. Da Ihr OpenPGP-Schlüssel aber inzwischen weitere Beglaubigungen haben kann, sollten Sie ihn ggf. erneut sichern.

Öffnen Sie Kleopatra, selektieren Sie Ihren eigenen Schlüssel 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 `.p12` (S/MIME) ausgewählt. Bei diesen Dateien handelt es sich um Binärdateien, die Ihr Schlüsselpaar (inkl. geheimem Schlüssel) verschlüsselt enthalten.

Bei Aktivierung der Option *ASCII-geschützt* (*ASCII armor*) erhalten Sie die Dateiendung `.asc` (OpenPGP) bzw. `.pem` (S/MIME). Diese Dateitypen können mit jedem Texteditor geöffnet werden – Sie sehen dort allerdings nur den Buchstaben- und Ziffernsalat, den Sie schon kennen.

Ist diese Option nicht ausgewählt, so wird eine verschlüsselte Datei mit der Endung `.gpg` (OpenPGP) oder `.p12` (S/MIME) angelegt. Diese Dateien sind Binärdateien, sie können also nicht mit einem Texteditor angesehen werden.

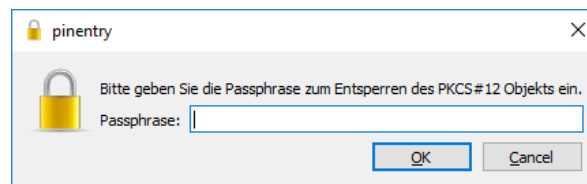
Beide Schlüsselteile – der öffentliche und der geheime – werden von Kleopatra in **einem** einzigen geheimen Schlüsselpaar abgespeichert.

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

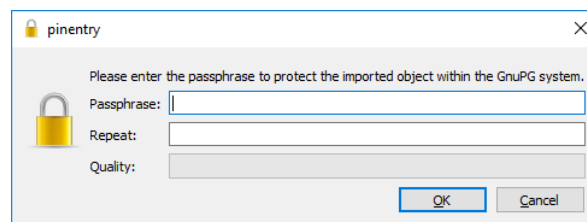
17.2. Import

Zum Importieren Ihres zuvor exportierten geheimen Schlüssels in Kleopatra gehen Sie so vor, wie Sie es vom Import fremder öffentlicher Schlüssel gewohnt sind (vgl. Kapitel ??):

Klicken Sie auf *Datei*→*Schlüssel 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 geheimen Schlüssels gefragt:



Setzen Sie nun eine Passphrase, gegebenenfalls auch eine neue, mit der nach dem Importvorgang Ihr geheimer Schlüssel geschützt werden soll:



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

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

Sichern Sie die Sicherungskopie Ihres geheimen Schlüssels – möglichst auf einem physikalisch gesicherten (z.B. in einem Tresor) externen Medium. Löschen Sie sie danach von Ihrer Festplatte und denken Sie auch daran, die gelöschte Datei aus Ihrem „Papierkorb“ zu entfernen. Andernfalls stellt diese Datei ein großes Sicherheitsrisiko für Ihre geheime E-Mail-Verschlüsselung dar.



Es kann in einigen Fällen vorkommen, dass Sie ein mit PGP („Pretty Good Privacy“) exportierten Schlüssel nicht importieren können: Sie geben zwar die richtige Passphrase ein, diese wird aber nicht akzeptiert. Der Grund ist, dass bestimmte Versionen von PGP intern einen Algorithmus (IDEA) verwenden, den GnuPG aus rechtlichen Gründen nicht unterstützen kann.

Um das Problem zu beheben, ändern Sie in PGP einfach die Passphrase und exportieren/importieren Sie das OpenPGP-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, dass Sie sowohl die **Datei sicher löschen** als auch in PGP und in Gpg4win danach wieder eine echte **Passphrase setzen**.

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

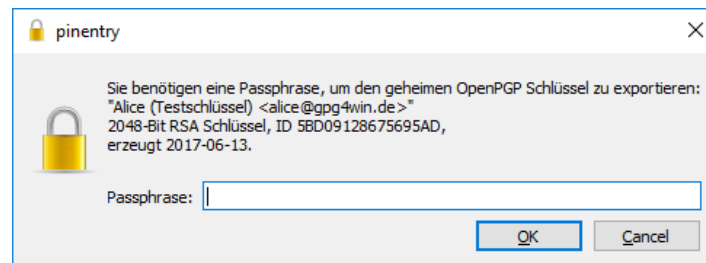
17.3. Paperkey

Mit Paperkey haben Sie die Möglichkeit Ihren privaten Schlüssel nicht nur auf digitale Medien zu sichern, sondern auf analoge, so können Sie ihn z.B. in einem Bankschließfach hinterlegen und von diesem analogen Medium auch wieder herstellen.

Wichtig: Dies ist nur für OpenPGP-Schlüssel möglich.

17.3.1. Export mit Paperkey

Wählen Sie dazu das zu exportierende Schlüsselpaar in der Übersicht aus. Über *Datei* → *Geheimen Schlüssel drucken...* öffnen Sie den Dialog zum drucken Ihres Schlüssels. Geben Sie zunächst die Passphrase Ihres Schlüssels ein.



Anschließend öffnet sich ein Druck-Dialog. Über diesen wählen Sie Ihren Drucker aus und drucken Ihren privaten Schlüssel.

Heften Sie dieses Dokument an einen sicheren Ort zur Verwahrung. Es sollte nicht in fremde Hände gelangen!

17.3.2. Import mit Paperkey

Um den zuvor exportierten Schlüssel wieder zu importieren, müssen Sie zunächst sicherstellen, dass Ihr öffentlicher Schlüssel bereits in Kleopatra vorhanden ist.

Anschließend öffnen Sie ein Textdokument und tippen den Paperkey ab.

Machen Sie einen Rechtsklick auf Ihren öffentlichen Schlüssel und klicken Sie auf *Details*. Wählen Sie im nun auftauchenden Schlüsseldialog den Knopf [*Weitere Details...*].

Abschließend tätigen Sie einen Rechtsklick auf einen der angezeigten Einträge. Nun wählen sie die Option zum importieren aus und navigieren im Dateidialog zu ihrem abgetippten Paperkey.

18. Konfiguration von Smartcards

Smartcards oder auch Chipkarten sind kleine Plastikkarten mit Chips auf ihnen, die einen kleinen Mikroprozessor enthalten. Sie kennen wahrscheinlich bereits einige Smartcards, wie zum Beispiel Simkarten in Handys oder Krankenkassenkarten. Zur Verwendung mit OpenPGP oder X.509 gibt es spezielle Karten, die mit GnuPG genutzt werden können.

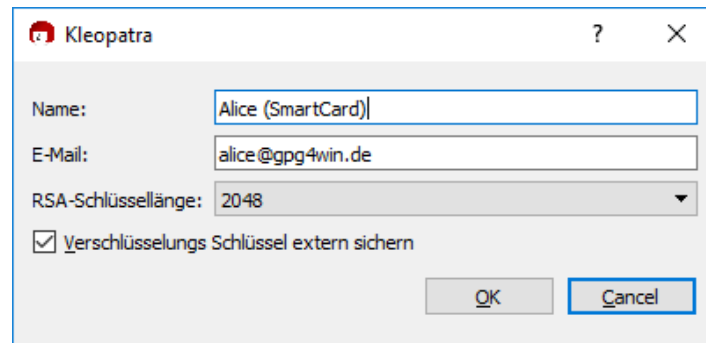
Ähnlich wie Krankenkassenkarten oder Simkarten haben die Smartcards, die mit GnuPG verwendet werden können, bestimmte Eigenschaften. Die Smartcards fungieren dabei als Speicher für den privaten Schlüssel und führen alle kryptografischen Operationen auf der Karte selbst durch. Dies kann für einige Szenarien sehr spannend sein, denn so muss der private Schlüssel selbst nicht mehr auf dem Computer, mit dem Sie arbeiten, nicht mehr vorhanden sein.

18.1. Nutzung von Smartcards mit OpenPGP

Zunächst muss die Frage beantwortet werden, ob eine Sicherheitskopie des Schlüssels behalten werden soll oder ob der Schlüssel ausschließlich auf der Smartcard existieren soll. Beide Möglichkeiten haben Ihre Vor- und Nachteile. Wenn der Schlüssel ausschließlich auf der Karte existieren soll, kann er nachträglich nicht mehr von der Karte kopiert werden. Dies bringt zusätzliche Sicherheit, sobald die Karte jedoch defekt ist oder abhanden kommt, haben Sie keinen Zugriff mehr auf den Schlüssel. Für die meisten Fälle bietet es sich an, den Schlüssel auf einem Rechner zu erstellen, mit Hilfe von Paperkey17.3 zu exportieren und den erstellten Schlüssel auf die Smartcard zu übertragen. Im Folgenden werden wir auf beide Fälle eingehen.

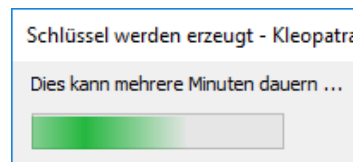
18.1.1. Erstellen des OpenPGP-Schlüssels auf der SmartCard

Schließen Sie das Smartcard-Lesegerät an Ihren Computer an. In Kleopatra finden Sie unter dem Punkt *Extras* → *SmartCards verwalten* die Einstellungen für Ihre Smartcard. Stecken Sie die zu benutzende SmartCard in das Lesegerät und drücken Sie [F5] und warten Sie, dass die SmartCard erkannt wird. Klicken Sie in SmartCard-Übersicht auf die Schaltfläche [*Neue Schlüssel erzeugen*]. Anschließend geben Sie die Informationen für die Schlüsselerzeugung ein.



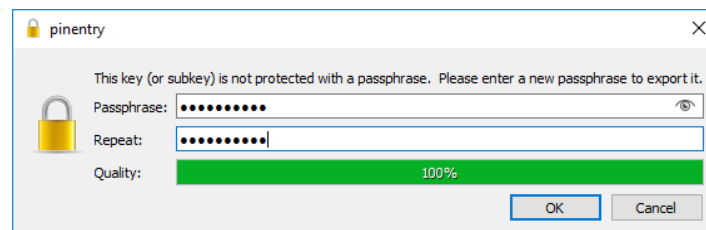
Achten Sie dabei auf das Feld *Verschlüsselungs Schlüssel extern sichern*. Dieses Feld sollte aktiv bleiben, damit Sie abschließend eine Sicherheitskopie Ihres Schlüssels machen können.

Anschließend werden die Schlüssel auf der Karte erzeugt.

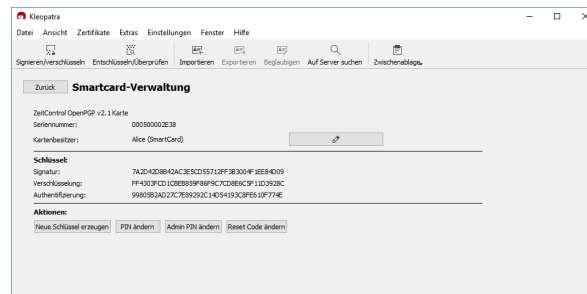


Je nach verwendetem SmartCard-Leser kann es während der Erzeugung zu einer Abfrage des Administrations-PINs der Karte kommen. Dieser liegt der Kartenanleitung bei.

Während der Schlüsselerzeugung werden Sie gebeten eine Passphrase einzugeben, hierbei gelten die gleichen Richtlinien wie auch bei der Erstellung des herkömmlichen Schlüssels.



Abschließend können Sie Ihren erstellen Schlüssel sichern. Beachten Sie dabei: Wenn der Schlüssel nicht gesichert wird und sie den Zugang zu Ihrer SmartCard verlieren, gibt es keine andere Möglichkeit mehr an Ihr Schlüsselpaar zu gelangen!



Herzlichen Glückwunsch! Sie haben erfolgreich einen Schlüssel auf Ihrer SmartCard erstellen.

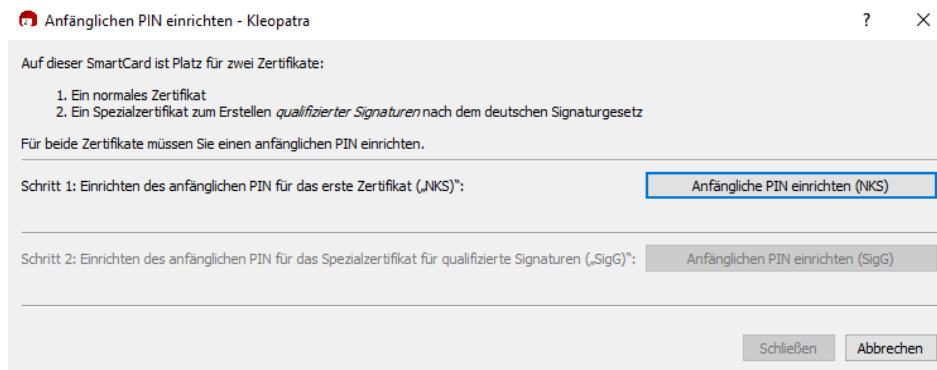
Bei zukünftigen kryptografischen Operationen werden Sie nun immer gebeten Ihre Smartcard in das Lesegerät einzuführen und ggf. eine PIN oder Passphrase darüber einzugeben.

18.2. Nutzung von NetKey-Cards mit X.509

X.509 Smartcards kommen im Unterschied zu OpenPGP Karten üblicherweise mit vorkonfigurierten Zertifikaten, welche von der Zertifizierungsstelle eingerichtet wurden. Kleopatra unterstützt die NetKey Karten der Telekom.

Bei der ersten Verwendung einer X.509 Smartcard muss eine PIN für die jeweiligen Zertifikate gesetzt werden. Zudem müssen die Zertifikate der Karte im Gpg4win-System registriert werden damit diese für Kryptographie- Aktionen zur Auswahl stehen.

Legt man eine, unbekannte, unterstützte X.509 Smartcard in einen Kartenleser ein blinkt das Systemtray Icon von Kleopatra. Nach einem Linksklick öffnet sich der Einrichtungsdialog um eine anfängliche PIN zu setzen:

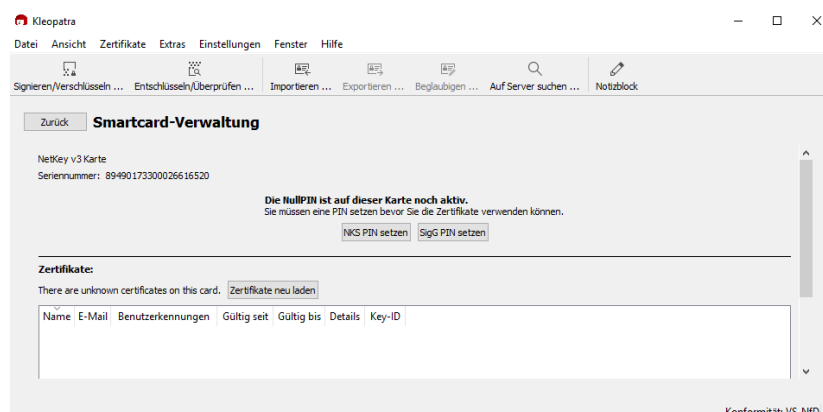


Dabei ist ein Spezialfall bei NetKey Karten das diese zwei Zertifikate unterstützen. Ein gewöhnliches und ein weiteres für qualifizierte Signaturen.¹

Nachdem die anfänglichen PINs eingerichtet sind können die Zertifikate registriert werden. Dazu kann man erneut auf das blinkende Symbol klicken. Es folgen einige Diagnoseausgaben. Diese können ignoriert werden. Anschließend sind die Zertifikate im System registriert.

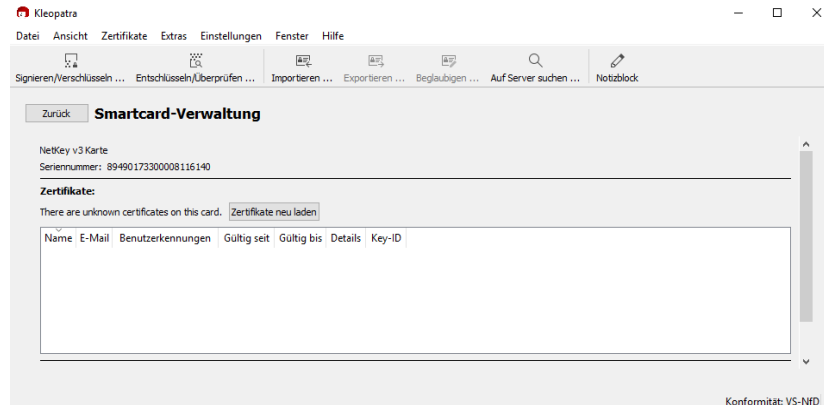
Alternativ zum Tray Icon können beide Aktionen auch über die SmartCard-Verwaltung angestoßen werden. Das Menü erreichen Sie über *Extras* → *SmartCards verwalten*.

Ansicht einer nicht initialisierten NetKey Karte:

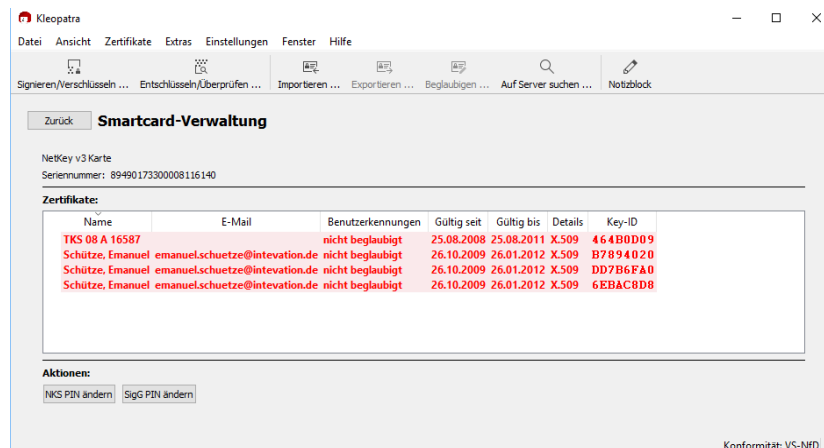


¹[https://de.wikipedia.org/wiki/Signaturgesetz_\(Deutschland\)](https://de.wikipedia.org/wiki/Signaturgesetz_(Deutschland))

Mit "Zertifikate neu Laden" werden die Zertifikate der SmartCard registriert:



Ist alles eingerichtet werden die auf der Karte vorhandenen Zertifikate in der SmartCard-Verwaltung angezeigt.



19. Systemweite Konfiguration und Vorbelegung für S/MIME

Im Rahmen einer zentralen Softwareverteilung oder in Umgebungen, in denen viele Anwender auf einem gemeinsamen Rechner arbeiten, ist es sinnvoll, einige systemweite Vorgaben und Vorbelegungen für Gpg4win einzurichten.



Das betrifft vor allem S/MIME, denn bei vorgegebenen Vertrauensketten ist es sinnvoll, dass die Anwender die Informationen dazu miteinander teilen.

Einige typische systemweite Einstellungen sind:

Vertrauenswürdige Wurzelzertifikate: Um zu vermeiden, dass jeder Anwender selbst die notwendigen Wurzelzertifikate suchen und installieren sowie deren Vertrauenswürdigkeit prüfen und beglaubigen muss (vgl. Abschnitt 21.7), ist eine systemweite Vorbelegung der wichtigsten Wurzelzertifikate sinnvoll.

Dazu sollten die Wurzelzertifikate abgelegt – wie in Abschnitt 21.3 beschrieben – und die vertrauenswürdigen Wurzelzertifikate definiert werden – wie in Abschnitt 21.6 beschrieben.

Direkt verfügbare CA-Zertifikate: Um den Anwendern zusätzlich die Mühe zu ersparen, die Zertifikate der Beglaubigungsinstanzen (Certificate Authorities, CAs) zu suchen und zu importieren, ist auch hier eine systemweite Vorbelegung der wichtigsten CA-Zertifikate sinnvoll. Eine Beschreibung hierzu finden Sie im Abschnitt 21.4.

Proxy für Zertifikatsserver- und Sperrlisten-Suche: Für die Gültigkeitsinformationen bieten die X.509-Protokolle verschiedene Möglichkeiten an. Von den meisten Zertifizierungsstellen werden Sperrlisten (auch CRLs genannt, nach RFC5280) und OSCP (Online Certificate Status Protocol, nach RFC2560) unterstützt. OSCP bringt zeitnähere Informationen, hat aber den Nachteil, dass Netzverkehr bis zum OSCP-Dienst erfolgt und daran auch gut erkannt werden kann, mit welchen Partnern gerade Nachrichten ausgetauscht werden. GnuPG kann mit beiden Möglichkeiten umgehen, es ist die Komponente „DirMngr“, welche als systemweiter Dienst läuft.

Es können interne Netzwerke keine direkten Verbindungen der einzelnen Rechner nach außen zulassen (zentrale Firewall), sondern einen Stellvertreterdienst (einen sogenannten „Proxy“) vorsehen. Der DirMngr kann ebenfalls mit HTTP- und LDAP-Proxies umgehen.

S/MIME-Zertifikate enthalten meist die Angabe, wo Ihre Sperrliste extern abgeholt werden kann. Oft kommt dabei HTTP vor, aber auch Verzeichnisdienste über LDAP. Anders als bei OpenPGP kann sich der Klient nicht aussuchen, wo er die Sperrliste abholen kann, er muss den verfügbaren Angaben folgen. Da manche Zertifikate ausschließlich Sperrlisten per LDAP zur Verfügung stellen, ist es erforderlich sowohl HTTP- als auch LDAP-Abfragen nach außen zuzulassen. Sofern möglich, kann ein Stellvertreterdienst auf Inhaltsebene sicherstellen, dass X.509-Sperrlisten ausschließlich mit korrekten Informationen übermittelt werden.



Ist in Ihrem Netzwerk für die bei OpenPGP bzw. S/MIME wichtigen HTTP- und HKP- oder LDAP-Abfragen ein Proxy nötig, so führen Sie folgende Schritte durch:

1. Stellen Sie X.509-Zertifikatsserver-Suche auf einen Proxy ein, wie in Abschnitt 21.5 beschrieben.
2. Stellen Sie Sperrlisten-Suche auf einen Proxy ein, wie ebenfalls in Abschnitt 21.5 beschrieben.
3. Starten Sie den DirMngr neu (siehe Abschnitt 20.7).

20. Bekannte Probleme und Abhilfen

20.1. GpgOL-Menüs und -Dialoge nicht mehr in Outlook zu finden

Es kann vorkommen, dass die von GpgOL zu Outlook hinzugefügten Menüs und Dialoge nicht mehr zu finden sind.

Das kann dann passieren, wenn ein technisches Problem auftrat und Outlook aus diesem Grund die GpgOL-Komponente deaktiviert hat.

Reaktivieren Sie GpgOL über das Outlook-Menü:

Outlook2007: ?→*Deaktivierte Elemente*

Outlook2003: ?→*Info*→*Deaktivierte Elemente*

Um GpgOL manuell zu (de-)aktivieren, nutzen Sie den Add-In-Manager von Outlook:

- **Outlook2003:** *Extras*→*Optionen*→*Weitere*→*Erweiterte Optionen...* → *Add-In-Manager...*
- **Outlook2007:** *Extras*→*Vertrauensstellungscenter*→*Add-Ins* – dann unter *Verwalten* die *Exchange-Clienterweiterungen* auswählen und auf [*Gehe zu...*] klicken.

20.2. GpgOL-Schaltflächen sind in Outlook2003 nicht in der Symbolleiste

Wenn bereits viele Schaltflächen in der Symbolleiste des Nachrichtenfensters vorhanden sind, so stellt Outlook2003 die Signieren-/Verschlüsseln-Icons von GpgOL nicht unbedingt sofort sichtbar dar.

Sie können diese Schaltflächen 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 Schaltflächen. Ein Klick auf einen dieser Einträge verschiebt ihn in den sichtbaren Teil der Symbolleiste.

20.3. GpgOL-Schaltflächen sind in Outlook2007 unter „Add-Ins“

Mit Outlook2007 wurde die sogenannte „Ribbon“-Oberfläche eingeführt. Diese Multifunktionsleiste im Outlook-Nachrichtenfenster besitzt verschiedene Registerkarten. Die GpgOL-Schaltflächen (für Verschlüsseln, Signieren etc.) sind unter der Registerkarte „Add-Ins“ eingeordnet; so wie alle Schaltflächen von Erweiterungen durch Outlook dort angelegt werden. Eine Integration der GpgOL-Schaltflächen z.B. unter „Nachrichten“ ist nicht möglich.

Sie können Ihre *Symbolleiste für den Schnellzugriff* anpassen und dort die Symbolleistenbefehle der Add-In-Registerkarte aufnehmen.

20.4. Fehler beim Start von GpgOL

Haben Sie Gpg4win (und damit die Programmkomponente GpgOL) erst auf einem Laufwerk installiert, anschließend wieder deinstalliert und auf einem anderen Laufwerk erneut installiert? Dann kann es sein, dass Outlook weiterhin den GpgOL-Pfad auf dem ersten (alten) Laufwerk sucht.

Dabei wird beim Start von Outlook die Programmerweiterung GpgOL nicht mehr gestartet und folgende Fehlermeldung erscheint:

Die Erweiterung '<alter-Pfad-zu-gpgol.dll>' konnte nicht installiert oder geladen werden. Das Problem kann u.U. durch das Benutzen von 'Erkennen und Reparieren' in der Hilfe behoben werden.

Lösen können Sie dieses Problem, in dem Sie den Outlook-internen (zwischengespeicherten) Programmerweiterungs-Pfad zurücksetzen. Löschen Sie dazu bitte folgende Datei:

```
%APPDATA%\Lokale Einstellungen\Anwendungsdaten\Microsoft\
Outlook\extend.dat
```

Dabei sollte Outlook nicht laufen. Anschließend starten Sie Outlook erneut. Outlook mit GpgOL sollten nun problemlos funktionieren.

20.5. Installation von Gpg4win auf einem virtuellen Laufwerk

Beachten Sie bitte, dass eine Installation von Gpg4win auf einem (mit dem Befehl `subst` simulierten) **virtuellen Laufwerk** nicht möglich ist. Diese virtuellen Laufwerke sind nur lokal für den aktuellen Benutzer nutzbar. Systemdienste (wie der DirMngr) sehen diese Laufwerke nicht. Der Installationspfad ist damit ungültig – die Installation stoppt mit einem Fehler in der Art `error:StartService: ec=3`. Installieren Sie bitte Gpg4win auf einem systemweit verfügbaren Laufwerk.

20.6. GpgOL überprüft keine InlinePGP-E-Mails von „CryptoEx“

Um signierte bzw. verschlüsselte InlinePGP-E-Mails zu prüfen bzw. zu entschlüsseln, die von der Outlook-Programmerweiterung „CryptoEx“ versendet wurden, muss in den GpgOL-Optionen die S/MIME-Unterstützung eingeschaltet sein.

Versichern Sie sich, dass die folgende Option in Outlook unter *Extras*→*Optionen*→*GpgOL* aktiv ist: *S/MIME Unterstützung einschalten*.

20.7. Keine S/MIME-Operationen möglich (Systemdienst „DirMngr“ läuft nicht)

Der „Directory Manager“ (DirMngr) ist ein durch Gpg4win installierter Dienst, der die Zugriffe auf Zertifikatsserver verwaltet. Eine Aufgabe des DirMngr ist das Laden von Sperrlisten (CRLs) für S/MIME-Zertifikate.



Es kann vorkommen, dass die S/MIME-Operationen (Signaturerstellung und -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 Sperrlisten prüft – geschieht das nicht, darf die jeweilige Operation nicht ausgeführt werden, da möglicherweise ein kompromittiertes Zertifikat genutzt wird.

Abhilfe schafft 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.

20.8. Keine S/MIME-Operationen möglich (CRLs nicht verfügbar)

Es kann vorkommen, dass die S/MIME-Operationen (Signaturerstellung und -prüfung, Ver- oder Entschlüsselung) nicht durchgeführt werden können, weil CRLs nicht verfügbar sind. In der Voreinstellung von Gpg4win ist es zwingend notwendig, dass Sperrlisten geprüft werden – geschieht das nicht, darf die jeweilige Operation nicht ausgeführt werden, da möglicherweise ein kompromittiertes Zertifikat genutzt wird.



Abhilfe schafft das Einrichten eines Stellvertreterdienstes („Proxies“) für das Abholen der Sperrlisten (vgl. Abschnitt 21.5).

Im Notfall (oder zum Testen) lassen sich die CRL-Prüfungen auch abschalten. Öffnen Sie dafür das Kleopatra-Menü *Einstellungen*→*Kleopatra einrichten* und anschließend die Gruppe *S/MIME-Prüfung*. Aktivieren Sie hier die Option *Nie Sperrlisten zu Rate ziehen*.

Achtung: Machen Sie sich bewusst, dass in diesem Fall ein wesentlich höheres Risiko besteht, ein kompromittiertes Zertifikat zu nutzen. Das Abschalten der Sperrlisten-Prüfung ist niemals einer Alternative zur Einrichtung eines Proxies.

20.9. Keine S/MIME-Operationen möglich (Wurzelzertifikat nicht vertrauenswürdig)

Für eine vollständige Prüfung von X.509-Zertifikatsketten muss dem jeweiligen Wurzelzertifikat vertraut werden. Andernfalls kann keine S/MIME-Operationen (Signaturerstellung und -prüfung, Ver- oder Entschlüsselung) durchgeführt werden.



Um einem Wurzelzertifikat das Vertrauen auszusprechen, haben Sie zwei Möglichkeiten:

- Den Fingerabdruck des entsprechenden Wurzelzertifikats in eine *systemweite* Konfigurationsdatei schreiben. Damit ist die Wurzel für alle Nutzer vertrauenswürdig. Sie müssen hierfür Windows-Administratorrechte besitzen. Eine genaue Erläuterung finden Sie im Abschnitt 21.6.
- Das Wurzelzertifikat durch den Benutzer setzen (keine systemweite Anpassung nötig). Dazu müssen Sie einmalig die Option *Erlauben, Wurzelzertifikate als vertrauenswürdig zu markieren* in Kleopatras Einstellung aktivieren. Anschließend werden Sie nach jedem Importieren neuer Wurzelzertifikate gefragt, ob Sie diesem vertrauen wollen. Genauer dazu im Abschnitt 21.7.

21. Dateien und Einstellungen von Gpg4win

21.1. Persönliche Einstellungen der Anwender

Die persönlichen Einstellungen für jeden Anwender befinden sich im Dateiordner:

`%APPDATA%\gnupg`

Oft entspricht das dem Dateiordner:

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

Beachten Sie, dass es sich um einen versteckten Dateiordner handelt. Um ihn sichtbar zu schalten, müssen Sie im Explorer über das Menü *Extras*→*Ordneroptionen* im Reiter *Ansicht* die Option *Alle Dateien und Ordner anzeigen* unter der Gruppe *Versteckte Dateien und Ordner* aktivieren.

In diesem Dateiordner befinden sich sämtliche persönlichen GnuPG-Daten, also die privaten Schlüssel, Schlüsselpaare, Vertrauensstellungen und Konfigurationen. Bei einer Deinstallation von Gpg4win wird dieser Ordner *nicht* gelöscht. Denken Sie daran, regelmäßig Sicherheitskopien dieses Ordners anzulegen.

21.2. Zwischengespeicherte Sperrlisten

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 (CRLs) von den Ausgabestellen der Zertifikate (CAs) abgeholt und für die Dauer ihrer Gültigkeit zwischengespeichert.



Abgelegt werden diese Sperrlisten unter:

`C:\Dokumente und Einstellungen\LocalService\Lokale
Einstellungen\Anwendungsdaten\GNU\cache\dirmgr\crls.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-Explorers.

In diesem Dateiordner sollten keine Änderungen vorgenommen werden.

21.3. Vertrauenswürdige Wurzelzertifikate von DirMngr

Für eine vollständige Prüfung von X.509-Zertifikaten muss den Wurzelzertifikaten vertraut werden, mit deren Hilfe die Sperrlisten signiert wurden.



Die Wurzelzertifikate, denen der DirMngr systemweit bei den Prüfungen vertrauen soll, werden im folgenden Dateiordner abgelegt:

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

Wichtig: Die entsprechenden Wurzelzertifikate müssen als Dateien im Dateiformat DER mit der Dateinamens-Erweiterung `.crt` oder `.der` im o.g. Dateiordner vorliegen.

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

Beachten Sie auch Abschnitt 21.6, um den Wurzelzertifikaten vollständig (systemweit) zu vertrauen.

21.4. Weitere Zertifikate von DirMngr

Da vor einer Krypto-Operation die X.509-Zertifikatskette geprüft werden soll, muss somit auch das jeweilige Zertifikat der Beglaubigungsinstanz („Certificate Authority“, CA) geprüft werden.



Für eine direkte Verfügbarkeit können CA-Zertifikate in diesem (systemweiten) Dateiordner abgelegt werden:

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

Zertifikate, die nicht hier oder bei den Anwendern vorliegen, müssen automatisch von X.509-Zertifikatsservern geladen werden.

Diese CA-Zertifikate können aber auch immer manuell vom Anwender importiert werden.

Es ist sinnvoll, im Rahmen von systemweiten Vorgaben hier die wichtigsten CA-Zertifikate abzulegen.

21.5. Systemweite Konfiguration zur Verwendung externer X.509-Zertifikatsserver

GnuPG kann so konfiguriert werden, dass bei Bedarf fehlende X.509-Zertifikate oder Sperrlisten auf externen X.509-Zertifikatsservern gesucht werden (vgl. auch Kapitel 19).



Für die **X.509-Zertifikatssuche** verwendet der Systemdienst DirMngr eine Liste von Zertifikatsservern, die in der Datei

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

angegeben werden können. Diese Zertifikatsserver werden für alle Nutzer (systemweit) verwendet. Jeder Nutzer kann darüber hinaus noch weitere, benutzerspezifische Zertifikatsserver für die Zertifikatssuche einrichten – z.B. direkt über Kleopatra (vgl. Kapitel 15.1).

Die genaue Syntax für die Zertifikatsserver-Einträge in der o.g. Konfigurationsdatei lautet:

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

Sind im internen Netz die Zugänge zu externen X.509-Zertifikatsservern mittels Firewall gesperrt, so kann man in der `ldapservers.conf` einen Proxy-Dienst für die entsprechende Durchleitung der Zertifikatssuche konfigurieren, wie folgende Zeile im Beispiel illustriert:

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

Für die Suche von **Sperrlisten** (CRLs) gibt es im gleichen Verzeichnis eine Konfigurationsdatei von DirMngr:

```
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
GNU\etc\dirmngr\dirmngr.conf
```

Beachten Sie, dass nur Administratoren diese Datei schreiben dürfen.

Folgende Proxy-Optionen können Sie nach Bedarf in dieser Konfigurationsdatei ergänzen (jede Option in einer Zeile):

- `http-proxy HOST[:PORT]` Diese Option verwendet `HOST` und `PORT` für den Zugang zum Zertifikatsserver. Die Umgebungsvariable `http_proxy` wird bei Verwendung dieser Option überschrieben.

Ein Beispiel:

```
http-proxy http://proxy.mydomain.example:8080
```

- `ldap-proxy HOST[:PORT]` Diese Option verwendet `HOST` und `PORT` für den Zugang zum Zertifikatsserver. Ist keine Portnummer angegeben, wird der Standard LDAP-Port 389 benutzt. Diese Option überschreibt die im Zertifikat enthaltene LDAP-URL bzw. nutzt `HOST` und `PORT`, wenn keine LDAP-URL angegeben ist.
- `only-ldap-proxy`

Diese Option sorgt dafür, dass DirMngr niemals irgendetwas anderes nutzt als den unter `ldap-proxy` konfigurierten Proxy. Denn normalerweise versucht DirMngr andere konfigurierte Zertifikatsserver zu verwenden, wenn die Verbindung über `ldap-proxy` fehl schlägt.

21.6. Systemweite vertrauenswürdige Wurzelzertifikate

Die systemweit als vertrauenswürdig vorbelegten Wurzelzertifikate werden definiert in der Datei
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten
\GNU\etc\gnupg\trustlist.txt



Um ein Wurzelzertifikat als vertrauenswürdig zu markieren, muss der entsprechende Fingerabdruck des Zertifikats, gefolgt von einem Leerzeichen und einem großen S, in die o.g. Datei eingetragen werden. Ein Zertifikat wird explizit als nicht vertrauenswürdig markiert, wenn die Zeile mit dem Präfix „!“ beginnt. Sie können hier auch mehrere Wurzelzertifikate eintragen. Zu beachten ist dann, dass jeder Fingerabdruck in einer neuen Zeile steht. Eine Zeile, die mit einem # beginnt wird als Kommentar behandelt und ignoriert.

Wichtig: Abschließend (am Ende der Datei) muss eine Leerzeile erfolgen.

Ein Beispiel:

```
# CN=Wurzel ZS 3,O=Intevation GmbH,C=DE
A6935DD34EF3087973C706FC311AA2CCF733765B S

# CN=PCA-1-Verwaltung-02/O=PKI-1-Verwaltung/C=DE
DC:BD:69:25:48:BD:BB:7E:31:6E:BB:80:D3:00:80:35:D4:F8:A6:CD S

# CN=Root-CA/O=Schlapphuet/L=Pullach/C=DE
!14:56:98:D3:FE:9C:CA:5A:31:6E:BC:81:D3:11:4E:00:90:A3:44:C2 S
```

Es kann in einigen Fällen sinnvoll sein, die Kriterien bei der Überprüfung der Wurzelzertifikate zu verringern. Sie können dazu hinter S eine weitere Flagge `relax` setzen: <FINGERABDRUCK> S relax

Wichtig: Die Verwendung von `relax` setzt die Sicherheit herab, muss daher individuell entschieden werden und sollte nur bei Problemen verwendet werden.

Genauere Details finden Sie in der aktuellen GnuPG-Dokumentation (Punkt „trustlist.txt“):
<http://www.gnupg.org/documentation/manuals/gnupg/Agent-Configuration.html>

Die genaue Syntax für die Einträge in die trustlist.txt lautet also:

```
[!]<FINGERABDRUCK> S [relax]
```

wobei ! und relax optional sind.

Anstelle der Flagge S sind noch die Werte P und * vorgesehen, die für zukünftigen Gebrauch reserviert sind.

Wichtig: Damit Wurzelzertifikate in Kleopatra vollständig als vertrauenswürdig markiert werden (Zertifikat wird blau hinterlegt), müssen die Wurzelzertifikate zusätzlich für den DirMngr abgelegt werden, wie unter Abschnitt 21.3 beschrieben.

21.7. Vertrauenswürdigkeit der Wurzelzertifikate durch Benutzer markieren

Wurzelzertifikate können auch jeweils von den einzelnen Benutzern als vertrauenswürdig markiert werden – eine systemweite Konfiguration (siehe Abschnitt 21.3 und 21.6) ist dann nicht erforderlich.



Öffnen Sie das Kleopatra-Menü *Einstellungen*→*Kleopatra einrichten* und anschließend die Gruppe *S/MIME-Prüfung*. Aktivieren Sie hier die Option *Erlauben, Wurzelzertifikate als vertrauenswürdig zu markieren*. Dadurch werden Sie beim Gebrauch eines bisher nicht als vertrauenswürdig eingestuften Wurzelzertifikats gefragt, ob Sie es nun als vertrauenswürdig einstufen wollen. Beachten Sie, dass der gpg-agent ggf. einmalig neu gestartet werden muss, bevor die Änderung wirksam wird (z.B. durch ausloggen und wieder einloggen).

Die von Ihnen als vertrauenswürdig (oder wahlweise explizit als nicht vertrauenswürdig) gekennzeichneten Wurzelzertifikate werden automatisch in folgender Datei gespeichert:

```
C:\Dokumente und Einstellungen\\
Anwendungsdaten\gnupg\trustlist.txt
```

Für die trustlist.txt gilt die gleiche Syntax wie im Abschnitt 21.6 beschrieben.

22. Probleme in den Gpg4win-Programmen aufspüren (Logdateien)

Es kann vorkommen, dass eine der Gpg4win-Programmkomponenten nicht wie erwartet zu funktionieren scheint.

Nicht selten ist dabei eine Besonderheit der Arbeitsumgebung verantwortlich, sodass die Softwareentwickler von Gpg4win das beobachtete Problem gar nicht selbst nachvollziehen können.

Um die Softwareentwickler bei der Problemsuche zu unterstützen oder auch, damit der Anwender selbst einmal in die technischen Detail-Abläufe hineinschnuppern kann, bieten die Gpg4win-Programme Unterstützung an.

In der Regel muss diese Unterstützung aber erst einmal eingeschaltet werden. Eine der wichtigsten Hilfsmittel sind Logdateien: Dort werden detaillierte Diagnose-Informationen zu den internen technischen Vorgängen festgehalten. Ein Softwareentwickler kann ein Problem und die mögliche Lösung oft leicht anhand dieser Logdatei erkennen, auch wenn das Problem auf den ersten Blick unverständlich wirken mag.

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

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

Logdateien – unter o.g. URL als „Debug-Informationen“ bezeichnet – bieten oft wertvolle Hinweise und sollten daher einem Fehlerbericht beigelegt werden.

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

22.1. Logdateien von Kleopatra einschalten

Die Logdaten von Kleopatra bestehen aus vielen Dateien, daher besteht der erste Schritt darin, zunächst einen Dateiordner für die Logdateien 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 Logdaten von Kleopatra erstellen möchte, separat vorgenommen werden und es muss darauf geachtet werden, dass unterschiedliche `kleologdir`-Dateiordner verwendet werden.

Der Pfad zu diesem Ordner muss nun in der neuen Umgebungsvariablen `KLEOPATRA_LOGDIR` vermerkt werden:

Öffnen Sie dazu die Systemsteuerung, wählen Sie 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 der angegebene Dateiordner existieren muss. Sie können ihn auch nachträglich erstellen.

Um die Logfunktion wirksam werden zu lassen, muss Kleopatra beendet und neu gestartet werden und der Dateiordner der Logdaten existieren sowie für Kleopatra beschreibbar sein.

Während Kleopatra verwendet wird, zeichnet es Ablauf-Informationen in der Datei `kleo-log` (Haupt-Logdatei) auf sowie möglicherweise viele Dateien mit einem Namen nach dem Schema:

`pipe-input-<ZEITSTEMPEL>-<ZUFALLSZEICHEN>`

Möglicherweise reichen diese Informationen einem Softwareentwickler 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_LOGOPTIONS`

Wert der Variable: `all`

Möglicherweise werden die Logdateien sehr schnell sehr groß. Sie sollten diese Logdaten-Aufzeichnung nur einschalten, um ein bestimmtes Fehlverhalten zu provozieren und dabei aufzuzeichnen.

Anschließend schalten Sie die Aufzeichnung wieder aus, indem Sie die Umgebungsvariable löschen oder ihren Namen leicht variieren (für späteres leichtes Reaktivieren). Vergessen Sie nicht, die Logdateien zu löschen oder zu verschieben, gerade wenn sie sehr umfangreich geworden sind oder es sich um sehr viele Dateien handelt. Bevor Sie eine neue Aufzeichnung beginnen, ist es ebenfalls sinnvoll, die Logdateien zu entfernen.



22.2. Logdatei von GpgOL einschalten

Um die Logdatei von GpgOL einzuschalten, müssen Sie einen „Registry-Editor“ starten. Geben Sie dazu das Kommando `regedit` unter *Start→Ausführen* oder in einer Eingabeaufforderung ein.

Wählen Sie nun aus der Baumstruktur auf der linken Seite den folgenden GpgOL-Schlüssel 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 der GpgOL-Logdatei führen Sie einen Doppelklick auf den Eintrag `enableDebug` aus und setzen Sie dessen Wert auf 1.

Als Wert für `logFile` geben Sie nun einen Namen für die Datei an, in die die Logdatei geschrieben werden soll, z.B.: `C:\TEMP\gpgol.log`

Starten Sie Outlook neu, um die Aufzeichnung zu starten.

Bedenken Sie, dass diese Datei sehr umfangreich werden kann. Stellen Sie `enableDebug` auf 0, sobald Sie die GpgOL-Logdatenaufzeichnung nicht mehr benötigen.

Vergessen Sie auch hier nicht, die Logdatei zu löschen oder zu verschieben, gerade wenn sie umfangreich geworden ist. Bevor Sie eine neue Aufzeichnung beginnen, ist es ebenfalls sinnvoll die Logdatei zu entfernen.

Fortgeschrittene technische Informationen zu GpgOL – wie z.B. weitere mögliche Werte für `enableDebug` – finden Sie im technischen (englischsprachigen) Handbuch von GpgOL. Es befindet sich in Ihrem Gpg4win-Installationsverzeichnis, in der Regel:

`C:\Programme\GNU\GnuPG\share\doc\gpgol\gpgol.pdf`



22.3. Logdatei von DirMngr einschalten

Bei DirMngr handelt es sich um einen systemweiten Dienst und daher ist das Einschalten der Logdatei nur mit Administratorrechten möglich.

Um die Logdatei einzuschalten, öffnen Sie zunächst folgende Konfigurationsdatei:

```
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
GNU\etc\dirmngr\dirmngr.conf
```

Fügen Sie die folgenden zwei Zeilen in die Konfigurationsdatei ein (den Pfad zur Logdatei können Sie natürlich anpassen):

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

Starten Sie anschließend den Dienst DirMngr unter *Systemsteuerung*→*Verwaltung*→*Dienste* neu, sodass die geänderte Konfigurationsdatei neu eingelesen wird und die vorgenommenen Einstellungen wirksam werden.

Kommentieren Sie Ihre Anpassung in o.g. Konfigurationsdatei aus (also # `debug-all`), sobald Sie die DirMngr-Logdtenaufzeichnung nicht mehr benötigen.

Vergessen Sie auch hier nicht, die Logdatei zu löschen oder zu verschieben, gerade wenn sie umfangreich geworden ist. Bevor Sie eine neue Aufzeichnung beginnen, ist es ebenfalls sinnvoll die Logdatei zu entfernen.



22.4. Logdatei von GnuPG einschalten

Für folgende GnuPG-Komponenten können Sie jeweils einzeln das Anlegen einer Logdatei 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 die jeweilige Logdatei im GnuPG Backend – erreichbar über das Kleopatra-Menü *Einstellungen* → *Kleopatra einrichten...* → *GnuPG-System*. Für jedes 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 Debug-Stufe *4 - Guru* ist die höchste Stufe und erzeugt dementsprechend große Dateien. Schalten Sie daher die Logdateien wieder aus (Debug-Stufe *0 - 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\gpg-agent.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.

Vergessen Sie auch hier nicht, die Logdatei zu löschen oder zu verschieben, gerade wenn sie umfangreich geworden ist. Bevor Sie eine neue Aufzeichnung beginnen, ist es ebenfalls sinnvoll die Logdatei zu entfernen.



22.5. Logdatei von GpgME einschalten

Die Logdatei-Einstellungen für GpgME („GnuPG Made Easy“) müssen – ebenso wie bei Kleopatra – für jeden Anwender separat vorgenommen werden.

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

Fügen Sie dort folgende neue **Benutzervariable** ein:

Name der Variable: GPGME_DEBUG

Wert der Variable: <DEBUGLEVEL;PFAD>, also z.B.: 5;c:\TEMP\gpgme.log

Beachten Sie, dass der angegebene Dateiordner existieren muss. Sie können ihn auch nachträglich erstellen.

Als Diagnosestufe wird hier der Wert 5 empfohlen. In den meisten Fällen liefert diese Stufe ausreichend Informationen. Falls nicht, können fortgeschrittene Nutzer diesen Wert schrittweise erhöhen.

Zum Ausschalten der Logdatenaufzeichnung setzen Sie die Diagnosestufe auf den Wert 0 oder entfernen Sie die Benutzervariable.

Vergessen Sie auch hier nicht, die Logdatei zu löschen oder zu verschieben, gerade wenn sie umfangreich geworden ist. Bevor Sie eine neue Aufzeichnung beginnen, ist es ebenfalls sinnvoll die Logdatei zu entfernen.

23. Warum Gpg4win nicht zu knacken ist ...

..., jedenfalls nicht mit heute bekannten Methoden, und falls die Software frei von Fehlern ist.

In der Realität liefern allerdings genau diese Fehler in den Programmen, bei ihrer Benutzung oder im Betriebssystem die Möglichkeiten, um doch noch an die geheimen Informationen zu gelangen. Freie Software bietet allerdings die denkbar besten Voraussetzungen, um diese Fehler zu vermeiden.

In jedem Beispiel dieses Kompendiums haben Sie gesehen, dass zwischen dem geheimen und dem öffentlichen Schlüssel eine Verbindung besteht. Nur wenn beide zueinander passen, können geheime Botschaften entschlüsselt werden.

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 Nichtmathematiker verstehen, da nur die Grundrechenarten (Addition, Subtraktion, Multiplikation, Division) benötigt werden, um eine spezielle Art der Addition und Multiplikation zu definieren. Es gehört sowohl zur Sicherheitsphilosophie der Kryptografie wie auch zum Prinzip der Freien Software, dass es keine geheim gehaltenen Methoden und Algorithmen gibt. Letztendlich versteht man auch erst dadurch wirklich, warum GnuPG (die eigentliche Maschinerie hinter Gpg4win) sicher ist.

Hier beginnt also sozusagen die Kür nach dem Pflichtteil.

24. GnuPG und das Geheimnis der großen Zahlen

Kryptografie 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 von 1.024 Bit und mehr gelungen, wie sie in GnuPG verwendet werden. Es ist zwar theoretisch möglich, aber praktisch nicht durchführbar. Denn selbst bei vielen Jahren Rechenzeit und Abertausenden von vernetzten Rechnern würde nicht 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 liefert, welches hinter RSA steckt – allerdings wohl nicht sehr bald.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) veröffentlicht von Zeit zu Zeit Prognosen und Einschätzungen, welche Schlüssellängen noch wie viele Jahre für absolute Geheimhaltung benutzt werden sollen. GnuPG erfüllt mit seinen Standardeinstellungen diese Mindestanforderungen. Wie im vorigen Kapitel schon angerissen, ist die Mathematik der mit Abstand sicherste Teil der praktisch angewandten Kryptografie.

¹Es wird hier RSA als Beispiel verwendet, da RSA als Voreinstellung von GnuPG verwendet wird und einfacher zu verstehen ist als der Elgamal-Algorithmus.



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 Nichtmathematiker verstehen, da nur die Grundrechenarten benötigt werden. 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.

Zwei Begriffsklärungen vorneweg:

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

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

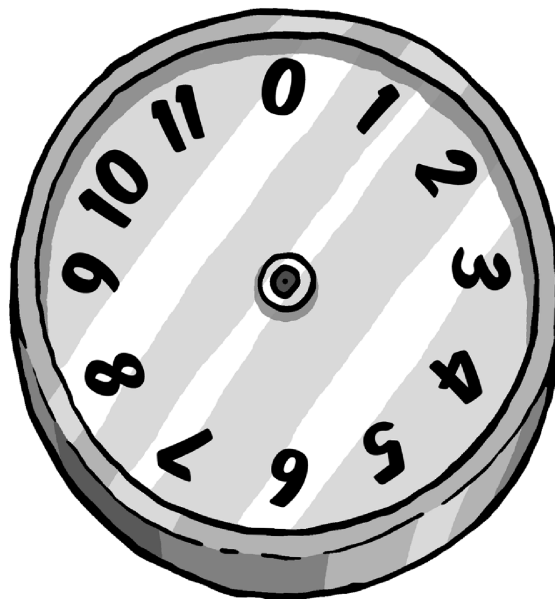
Die Verschlüsselung mit GnuPG basiert auf dem sogenannten RSA-Algorithmus². RSA steht für die Nachnamen von Ron Rivest, Ami Shamir und Leonard Adleman, die diesen Algorithmus im Jahr 1978 entdeckt haben. Dieser Algorithmus verwendet einen Typ der Arithmetik, die Rechnen mit Restklassen oder „Modulo-Arithmetik“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

24.1. 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 Sie beispielsweise mit dem Teiler oder der Modulzahl 5 rechnen, sagt man auch, „ich rechne 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 (die Modulzahl ist also 12) – eine Uhr mit einem normalen Zifferblatt, allerdings mit einer 0 anstelle der 12. Sie können damit Modulo-Arithmetik betreiben, indem Sie einfach den gedachten Zeiger bewegen.

Um beispielsweise $3 + 2$ zu rechnen, beginnen Sie bei der Ziffer 2 und drehen den Zeiger um 3 Striche weiter (oder Sie 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 Sie Zahlen also nach den normalen Regeln der Alltagsarithmetik, verwenden dabei jedoch immer nur den Rest nach der Teilung. Um anzuzeigen, dass Sie 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 z.B. „4 modulo 5“, schreibt aber kurz „4 mod 5“.

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

$$4 \bmod 5 + 3 \bmod 5 = 7 \bmod 5 = 2 \bmod 5$$

Anders ausgedrückt, ist in der Modulo-5-Arithmetik das Ergebnis aus 4 plus 3 gleich 2.

Ein weiteres Beispiel in Modulo-5-Arithmetik:

$$8 \bmod 5 + 6 \bmod 5 = 14 \bmod 5 = 4 \bmod 5$$

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

$$8 \bmod 5 + 6 \bmod 5 = 3 \bmod 5 + 1 \bmod 5 = 4 \bmod 5$$

Denn 3 ist dasselbe wie 8, und 1 dasselbe wie 6, da Sie sich ja nur für den jeweiligen Rest nach der Teilung durch 5 interessieren.

An den letzten Beispielen wird deutlich, dass bei der Modulo-Arithmetik jederzeit ein ganzzahliges Vielfaches der Modulzahl (hier 5) addiert werden kann, das Rechenergebnis aber stets dasselbe bleibt.

Dieses Schema funktioniert auch beim Multiplizieren.

Ein Beispiel:

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

Ebenso können Sie schreiben:

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

da Sie 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 Sie nur den Rest nach Teilung durch 5 betrachten.

Wiederum können Sie feststellen, dass es egal ist, wenn Sie das Vielfache von 5 einfach weglassen.

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

Noch drei Beispiele mit anderen Modulzahlen:

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

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

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

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

24.2. 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 typisch zwischen 0 und 255 liegen.

Sie können also eine Nachricht auch in eine Zahlenfolge umwandeln. Nach welcher Methode (oder welchem Algorithmus) dies geschieht, wird im nächsten Abschnitt beschrieben. Darin wird die Methode vorgestellt, 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 (also eine 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 Erzeugung eines Schlüsselpaares 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 Rechner – kennen diese beiden Primzahlen und Sie müssen für deren 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 mit dem Buchstaben n bezeichnet. Dies ist die Modulzahl, mit der Sie später immer rechnen werden.

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

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

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

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 geheim gehalten werden – es ist Ihr geheimer Schlüssel. Die beiden Primzahlen (p und q) 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.

24.3. RSA-Verschlüsselung mit kleinen Zahlen

Sie 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 vielen Ziffern bestehen.

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

Und zwar erzeugen Sie zunächst den öffentlichen Schlüssel.

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

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

Die dritte Zahl ist der geheime Schlüssel. Diese Zahl wird wie folgt errechnet:

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

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

$$13 \bmod 60 * ? \bmod 60 = 1 \bmod 60$$

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

$$13 \bmod 60 * 37 \bmod 60 = 481 \bmod 60 = 1 \bmod 60$$

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

Sie verschlüsseln mit dem öffentlichen Schlüssel eine Nachricht

Nun zerlegen Sie 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 Ihr öffentlicher Schlüssel.

Nehmen Sie ein Beispiel mit der Zahl 2: Sie wird in die Zahl 30 umgewandelt, weil $2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 2^{13} = 8192 = 30 \bmod 77$ 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 24.1.:

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

Entschlüsseln Sie eine Nachricht mit dem privaten Schlüssel

Um das Beispiel mit der 2 von oben umzukehren, also die Nachricht zu dekodieren, multiplizieren Sie 30 (die umgewandelte 2) 37 mal mit sich selbst (30^{37}). Das Ergebnis wird modulo der Modulzahl 77 gerechnet. Sie erinnern sich: 37 ist der geheime Schlüssel.

Diese wiederholte Multiplikation ergibt eine Zahl, die $2 \bmod 77$ ist.

Das andere Beispiel: Die Zahl $47 \bmod 77$ wird zur Zahl $75 \bmod 77$ dekodiert.

Tabelle 24.2 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 24.2.: Zahlentransformation modulo 77, unter Verwendung des geheimen Schlüssels 37

Um eine Zahl mit Tabelle 24.2 zu transformieren, gehen Sie nach der gleichen Methode vor wie bei Tabelle 24.1. 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 man davon ausgeht, dass Sie bei der Umwandlung von 25 mit Hilfe von Tabelle 24.1 als Ergebnis 60 erhalten, dann sollten Sie auch bei der Transformation von 60 mit Hilfe von Tabelle 24.2 zum Ergebnis 25 gelangen. Dabei haben Sie den öffentlichen Schlüssel, hier die 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 Sie sich der Modulo-77-Arithmetik bedient.

Zusammenfassung

Sie haben ...

- durch den Rechner zwei zufällige Primzahlen erzeugen lassen;
- daraus das Produkt und den öffentlichen und den geheimen Schlüssel gebildet;
- mit dem öffentlichen Schlüssel eine Nachricht verschlüsselt;
- mit dem geheimen Schlüssel eine Nachricht 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.

Sie haben gesehen, dass die Rechnerei sogar in diesem einfachen Beispiel recht aufwändig 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 24.1 – 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. Ein aufmerksamer Leser wird auch bemerkt haben, dass etliche Zahlen, z.B. 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 24.3.:

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 extrem aufwändig, aufgrund des öffentlichen Schlüssels den geheimen Schlüssel zu ermitteln. Selbst die schnellsten Rechnern der Welt würden sehr lange benötigen, 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 man sie oben in den Tabellen 24.1 und 24.2 findet – stetig ab, je größer die Primzahlen werden. Von Primzahlen in der Größenordnung, die Sie 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 zwei großen Primzahlen bilden. Kein Rechner der Welt dagegen kann aus diesem Produkt wieder die ursprünglichen Primzahlen herausrechnen – jedenfalls nicht in vertretbarer Zeit.

24.4. Die Darstellung mit verschiedenen Basiszahlen

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

Dazu machen Sie sich zunächst mit den Zahlenpotenzen vertraut.

Zwei hoch eins, dargestellt als $2^1 = 2$;

zwei hoch drei, dargestellt als $2^3 = 2 * 2 * 2 = 8$;

zwei hoch zehn, dargestellt als $2^{10} = 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 1024$.

Jede Zahl hoch 0 ist gleich 1, z.B. $2^0 = 1$ 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 z.B. 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, ...

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 Sie ja auch Ihre normale Zahlen mit den Ziffern 0 bis 9 dar.

„578“, z.B., bedeutet $5 * 10^2 + 7 * 10^1 + 8 * 10^0 = 500 + 70 + 8$, und dies entspricht 578.

Hier haben Sie die „5“ stellvertretend für fünfhundert, „7“ für siebzig und „8“ für acht. In diesem Fall ist die Basis 10, eine für Sie 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 man Zahlen normalerweise zur Basis 10 darstellt, machen Sie sich 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 Sie Zahlen nicht zur Basis 10 darstellen, so müssen Sie 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, ...

Ihr dreistelliger Tacho zur Basis 8 stellt z.B. folgende Zahl dar:

356

Die 6 auf dem rechten Rädchen zählt einzelne Kilometer, also $6 * 8^0 = 6$ Kilometer.

Die 5 auf dem Rädchen in der Mitte für $5 * 8^1$, also 40 Kilometer.

Die 3 links steht für je 64 Kilometer pro Umdrehung, also hier $3 * 8^2$ Kilometer.

So rechnet man also mit Zahlen zur Basis 8. Ein Beispiel: 72_8 bedeutet $7 * 8^1 + 2 * 8^0 = 58$.

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, sodass 453_8 eigentlich $4 * 64 + 5 * 8 + 3$ bedeutet, was 299_{10} 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). Sie verfügen also über keine Ziffer, die 10 oder eine größere Zahl darstellt. Um 10 darzustellen, brauchen Sie zwei Ziffern, mit denen Sie dann die „10“ schreiben können.

Sie haben also nur die Ziffern 0 bis 9.

So ähnlich ist es, wenn Sie mit der Basiszahl 8 rechnen: Dann haben Sie nur die Ziffern 0 bis 7 zur Verfügung. Wollen Sie zu dieser Basis eine höhere Zahl als sieben darstellen, müssen Sie wieder zwei Ziffern verwenden – z.B. 9_{10} ist 11_8 , 73_{10} ist 111_8 .

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

$$1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 1 * 16 + 0 * 8 + 1 * 4 + 0 * 2 + 1 = 21$$

In der Informatik verwendet man auch Gruppen von acht Binärziffern, genannt „Byte“. Ein Byte kann Werte zwischen 0 – dargestellt als Byte 00000000_2 – und 255 – dargestellt als Byte 11111111_2 – annehmen. Ein Byte stellt also Zahlen zur Basis $2^8 = 256$ dar.

Zwei weitere Beispiele:

$$10101010_2 = 170$$

$$00000101_2 = 5$$

Da ein Rechner die Buchstaben, Ziffern und Satzzeichen als Bytes speichert, schauen Sie sich an, welche Rolle dabei die Darstellung zur Basis 256 spielt.

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

Diese Zahlenwerte sind für Rechner standardisiert und werden ASCII-Code genannt.

Sie können also die Silbe „un“ darstellen durch die Zahl:

$$117 * 2^{8*1} + 110 * 2^{8*0} = 117 * 256 + 110 = 30062$$

Entsprechend würde man die Buchstabenfolge „und“ mit der Zahl

$$117 * 2^{8*2} + 110 * 2^{8*1} + 100 * 2^{8*0} = 117 * 65536 + 110 * 256 + 100 = 7695972$$

darstellen, denn das „d“ wird durch 100 repräsentiert.

Sie haben hier also Zahlen und Symbole, die auf der Computertastatur als normale Zahlen zur Basis 10 stehen, intern durch Zahlen zur Basis $2^8 = 256$ repräsentiert.

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

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

Die folgende Prozedur umfasst mehrere Schritte, die hier zunächst zusammengefasst und anschließend in Einzelschritten dargestellt werden:

1. Die Nachricht *aba, cad, aca* wandeln Sie – wie beschrieben – in Zahlen um.
2. Diese Darstellung, beispielhaft zur Basis $2^2 = 4$ (statt $2^8 = 256$), wandeln Sie in eine Darstellung zur Basis 10 um, damit Sie zur Verschlüsselung die Tabelle 24.1 benutzen können, in der 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 Sie die zur Basis 10 kodierte Nachricht auf die Basis 4 zurück und wandeln sie dann wieder in eine Buchstabenfolge.
4. So entsteht aus der Nachricht *aba, cad, aca* die verschlüsselte Nachricht *dbb, ddd, dac*.

Und nun ausführlich:

1. Die Nachricht *aba, cad, aca* wandeln Sie in Zahlen um.

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

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

Verschlüsseln Sie nun die Nachricht *aba, cad, aca*. Sie 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 Sie bereits aus dem früheren Kapitel: Sie haben damit die Tabellen 24.1 und 24.2 konstruiert.

2. Diese Darstellung zur Basis 4 wandeln Sie in eine Darstellung zur Basis 10 um. Damit können Sie zur Verschlüsselung die Tabelle 24.1 benutzen, in denen die Zahlen ja auch auf 10er-Basis dargestellt werden.

Weil Sie vier Buchstaben für die Nachricht verwenden, rechnen Sie zur Basis 4. Für die Rechnung modulo 77 müssen Sie 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 Sie stattdessen die Nachricht in vier Zeichen lange Stücke zerlegen, würde 3333_4 den Wert 76_{10} übersteigen und es würden unerwünschte Doppeldeutigkeiten entstehen.

Folglich würde die Nachricht in dreiziffrigen Stücken nun

$$aba, cad, aca$$

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

Da Sie 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 Sie z.B. 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 Sie zur Verschlüsselung die Tabelle 24.1 von Seite 124 benutzen, die ja zur Basis 10 berechnet wurde. Diese Tabelle benutzen wir, weil Sie mit dem schon bekannten Schlüsselpaar arbeiten wollen. Dabei entsteht eine kodierte Nachricht zur Basis 10.

Zum Verschlüsseln der Nachricht nehmen Sie jetzt die o.g. Tabelle 24.1 zur Hilfe. Die Nachricht wird nun zu der Zahlenfolge 53, 63, 50 (zur Basis 10).

4. 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 24.2 und erhält die Sequenz 4, 35, 8. Und das entspricht als Zahlenfolge genau der ursprünglichen Nachricht.

Anhand der Tabellen 24.1 und 24.2 können Sie ebenso gut Nachrichten unter Verwendung des geheimen Schlüssels (d.h., erst Tabelle 24.2 benutzen) verschlüsseln, dann mit dem öffentlichen Schlüssel (d.h., Tabelle 24.1 als zweites benutzen) dekodieren und damit Ihre ursprüngliche Zahl wieder herstellen. Das bedeutet, dass der Inhaber des geheimen Schlüssels damit Nachrichten unter Verwendung des RSA-Algorithmus verschlüsseln kann, die daher 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 nicht mehr erleben dürfte, diese dann noch lesen zu können...

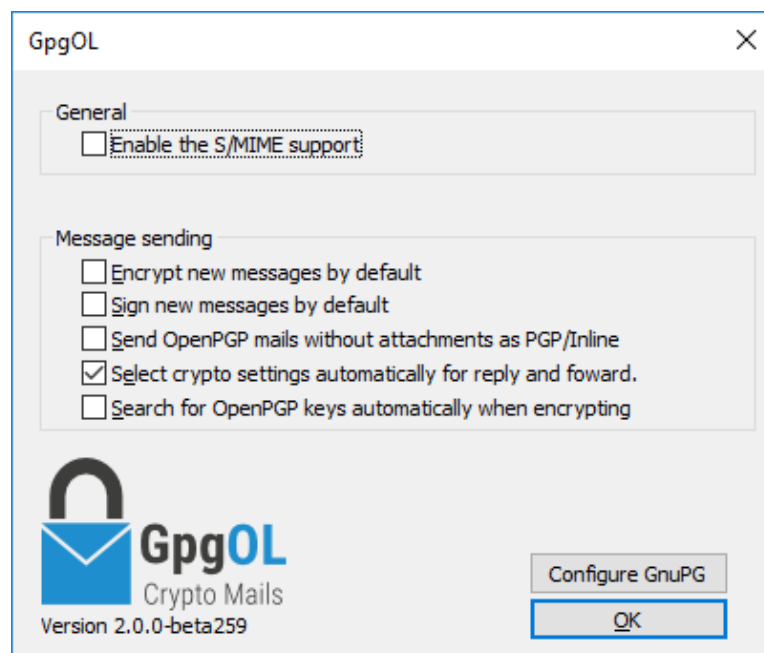
Teil III.

Anhang

A. Hinweise zur Outlook- Programmerweiterung GpgOL

GpgOL ist eine Programmerweiterung für Microsoft Outlook, es integriert dort die Bedienung von GnuPG.

GpgOL wird durch den Gpg4win-Installationsassistenten installiert. Beim nächsten Start von Outlook finden Sie in Ihrer Adressleiste einen Eintrag für GpgOL. Wenn Sie auf den Pfeil neben dem Schriftzug klicken kommen Sie in das Einstellungs Menü:



Die Karteikarte *GpgOL* unterteilt sich in zwei Bereiche:

1. Allgemein

- *S/MIME Unterstützung einschalten*

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 der Gpg4win-Komponente GnuPG. Konkret heißt das, dass alle Einstellungen, das Schlüsselmanagement und die Benutzerdialoge unterschiedlich sind. Es ist zu beachten, dass Outlook von sich aus OpenPGP-Unterstützung anbietet.

Wenn Sie S/MIME in Outlook mit Gpg4win nutzen möchten, aktivieren Sie die GpgOL-Option *S/MIME Unterstützung einschalten*. Sollten Sie das von Outlook unterstützte S/MIME nutzen wollen, lassen Sie diese GpgOL-S/MIME-Option deaktiviert.

2. Nachrichten Versenden

- *Neue Nachrichten per Voreinstellungen verschlüsseln*
- *Neue Nachrichten per Voreinstellungen signieren*

Diese beiden Optionen in diesem Bereich 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.

- *Sende OpenPGP Nachrichten ohne Anhänge als PGP/Inline*

Es gibt zwei verschiedenen Wege eine E-Mail verschlüsselt oder signiert zu senden. Zum einen PGP/MIME und zum anderen PGP/Inline. Diese unterscheiden sich darin, wie Ihre Nachricht signiert wird. Mit PGP/Inline wird der reine Textblock signiert und verschlüsselt und mit PGP/MIME wird der komplette E-Mail Inhalt signiert und verschlüsselt. Deswegen steht die PGP/Inline Methode auch nur bei Nachrichten ohne Anhängen zur Verfügung. Wir raten davon ab diese Methode dauerhaft zu aktivieren.

- *Wähle Verschlüsselungseinstellungen automatisch bei Antwort oder Weiterleitung*

Wenn Sie eine verschlüsselte oder signierte Nachricht erhalten, wird eine sinnvolle kryptografische Voreinstellung für die Weiterleitung oder Antwort vorgenommen. Diese Option ist standardmäßig aktiviert.

- *Suche automatisch nach OpenPGP Schlüsseln bei Verschlüsselung*

Wenn Sie eine Nachricht an einen bislang unbekannten Empfänger senden und die Option zum verschlüsseln aktiviert haben, so wird automatisch auf dem Schlüsselservers (siehe Kapitel 15) nach einem passenden Schlüssel für den Empfänger gesucht.

Alle Optionen sind nach einer Neuinstallation bereits sinnvoll vorgelegt.

B. GnuPG mit anderen E-Mail-Programmen 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. Große Unterschiede gibt es jedoch im Bedienkomfort: Je besser GnuPG in ein E-Mail-Programm integriert ist, desto einfacher die Verwendung.

Die einfachste Methode, z.B. 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 komplexe PGP/MIME-E-Mails werden Sie über eine Zwischenspeicherung als Datei gehen müssen. Beide Methoden werden im ersten Teil dieses Kompendiums beschrieben.

Eine Integration von GnuPG wird derzeit für folgende E-Mail-Programme unter Windows angeboten:

Thunderbird mit **Enigmail**¹.

Outlook ab Version 2003 mit GpgOL. GpgOL ist Bestandteil des Gpg4win-Pakets.

Claws Mail: Dieses E-Mail-Programm ist verwandt mit dem Gpg4win-Paket und kann optional installiert werden. Eine solche Installation konfiguriert bereits die Programmerweiterung für die Verwendung von PGP/MIME und S/MIME. Diese Erweiterung verwendet jedoch nicht Kleopatra und bietet daher derzeit nicht denselben Komfort, wie die Outlook-Erweiterung GpgOL.

KMail/Kontact: Eine komfortable und erprobte Integration von GnuPG bieten KMail und Kontact. Sie sind für nahezu jedes GNU/Linux-System und neuerdings auch für Windows und MacOS X verfügbar.

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

C. Automatische Installation von Gpg4win

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

In einigen Fällen, wie z.B. für Software-Verteilungssysteme, ist es notwendig, dass die Installation von Gpg4win ohne die Interaktion über Dialoge funktioniert. Um aber trotzdem vorab alle Installationseinstellungen bestimmen zu können, unterstützt Gpg4win das Setzen des Installationspfads und weiterer Optionen auf der Kommandozeile wie auch in einer Steuerungsdatei.

Der Installationspfad kann mit der Option `/D=<PFAD>` angegeben werden, welche als letzte Option auf der Kommandozeile übergeben werden muss. Der Dateiname (hier: `gpg4win.exe`) kann je nach Version variieren. Die Groß-/Kleinschreibung bei der Eingabe in der Kommandozeile ist hierbei wichtig. Eventuell sind noch Zugriffsrechte (z.B. lesen und schreiben) auf den Installationsordner zu setzen. Ein Beispiel:

```
gpg4win.exe /D=D:\Programme\Gpg4win
```

Mit der Option `/S` läuft die Installation „still“ (also ohne Dialog) ab. Ohne Angabe von weiteren Parametern werden alle Voreinstellungen übernommen.

Gpg4win 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 Pfadangaben für die zu installierenden Konfigurationsdateien. Relative Pfade, also abhängig vom aktuellen Arbeitsverzeichnis, dürfen hier nicht angegeben werden. Absolute Pfade enthalten den vollständigen Pfad inklusive der Laufwerksangabe. In der Regel sind die Einstellungen dann anzugeben, wenn nicht die Voreinstellung verwendet werden soll. Ausnahmen davon sind im Beispiel auf der nächsten Seite dokumentiert.

Hier ist ein Beispiel für den Inhalt einer Steuerungsdatei, das **alle** erlaubten Schlüsselworte zeigt:

```
[gpg4win]
; Installationseinstellungen. Weg- oder leer lassen für
; Voreinstellung
inst_gpgol = true
inst_gpgex = true
inst_kleopatra = true
inst_gpa = true
inst_compendium = true

; Die Stellen, an denen Verknüpfungen erzeugt werden sollen.
inst_start_menu = true
inst_desktop = true
```

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 Programmen

Dieser Abschnitt erläutert Ihnen, 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, vollzieht sich unter Verwendung der im alten Programm vorhandenen Möglichkeiten. Suchen Sie sowohl nach einem Menüpunkt, um Ihre privaten (geheimen) Schlüssel zu sichern, als auch nach einem Menüpunkt, um alle vorhandenen öffentlichen Schlüssel zu sichern. Sichern Sie diese dann in einer oder mehreren Dateien.

Sobald Sie Gpg4win installiert haben, prüfen Sie, ob Ihre alten Schlüssel bereits vorhanden sind. Sie können dies mit den Zertifikatsmanagern Kleopatra oder GPA machen. Sind die Schlüssel schon vorhanden, so entsprach das alte Verschlüsselungssystem 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 17.

Falls Ihr altes Kryptographiesystem den Zertifikatsmanager GPA verwendet, so können Sie die dort vorhandene Backupmöglichkeit benutzen. Diese sollte sehr ähnlich zu dieser Funktion 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.

Migration von Gpg4win-1.1.x nach Gpg4win-2.x

Es wird dringend empfohlen, zunächst Gpg4win-1.1.x zu deinstallieren, bevor anschließend Gpg4win-2.x installiert wird.

Technischer Hintergrund

Das Problem bei einer Migration *ohne* Deinstallation von Gpg4win-1.1.x ist folgende Sequenz:

1. Installation von Gpg4win in der Version X, inkl. Komponente K.
2. Installation von Gpg4win in der Version X+1, aber Komponente K wird diesmal deselektiert.
Effekt: Die alte Komponente von K bleibt installiert in der Version X.

3. Deinstallation von Gpg4win in der Version X+1.
Effekt: Die Komponente K der Version X bleibt verwaist zurück.
Dies ist eine Beschränkung von Gpg4win seit der ersten Version.

Anmerkung 1: Beim Sprung von 1.1.x auf 2.x tritt dieser Fall *immer* ein, da bestimmte Komponenten K nicht mehr existieren (z.B. GpgEE), also auf jeden Fall (automatisch) als deselektiert zu betrachten sind.

Anmerkung 2: Im Falle von MSI übernimmt Windows die Aufgabe, nicht mehr verwendete Komponenten zu entfernen. Das bedeutet, dass der MSI-Installationsassistent in dem obigen Szenario korrekt handelt (alte Komponente K in Version X ist nach Schritt 2 nicht mehr auf dem Betriebssystem vorhanden).

E. Deinstallation von Gpg4win

Soll Gpg4win deinstalliert werden, dann sollten Sie zunächst alle nicht notwendigen Anwendungen beenden und alle Schlüssel sichern. Falls Sie auf Ihrem Rechner mit eingeschränkten Rechten arbeiten sollten, ist es für die Deinstallation außerdem notwendig, mit **Administratorrechten** angemeldet zu sein. Wurde die Installation bereits über Ihr Benutzerkonto durchgeführt, so verfügt es über Administratorrechte.

Wichtig:

Bevor Sie die Deinstallation durchführen, sollten Sie unbedingt Ihre mit GpgOL bearbeiteten E-Mails in Outlook von den GpgOL-Informationen „bereinigen“. Denn: Gpg4win/GpgOL setzt für jede Krypto-E-Mail in Outlook eine bestimmte Markierung. Sie müssen vor der Deinstallation diese Markierung zurücksetzen, damit andere Kryptografiesoftware Ihre E-Mails später korrekt lesen und z.B. entschlüsseln kann.

GpgOL stellt Ihnen für diese **Re-Migration** direkt in Outlook folgende Funktion bereit:

Wählen Sie einen Outlook-E-Mail-Ordner aus, dessen E-Mails Sie zurücksetzen möchten, und klicken Sie im Menü von Outlook auf *Extras → GpgOL Eigenschaften aus diesem Ordner entfernen*.

Sie werden darauf hingewiesen, dass GpgOL (für die anschließende Deinstallation) ausgeschaltet wird. Bestätigen Sie die Frage, ob Sie die E-Mails der jeweiligen Ordner von den Markierungen durch GpgOL reinigen wollen, mit *Ja*.

Führen Sie dieses Kommando nun für alle Outlook-Ordner durch.

Nachdem Sie alle Ordner zurückgesetzt haben, beginnen Sie mit der Deinstallation von Gpg4win.

Es gibt drei 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 alle Gpg4win-Programmkomponenten von Ihrem Betriebssystem.

- Die zweite Möglichkeit zur Deinstallation von Gpg4win bietet Ihnen die ausführbare Datei `gpg4win-uninstall.exe`. Sie wird mit Gpg4win mitgeliefert und liegt im Installationsordner (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.
- Diese ausführbare Datei ist auch im Startmenü unter Gpg4win vorhanden.

In allen drei Fällen werden alle Dateien von Gpg4win aus dem Installationsordner sowie die Verknüpfungen in Startmenü, Desktop und Schnellstartleiste entfernt.

Nicht gelöscht werden die benutzerspezifischen und systemweiten Anwendungs-Dateiordner mit den Konfigurationseinstellungen:

- Benutzerspezifische GnuPG-Anwendungsdaten
in %APPDATA%\gnupg, das entspricht in der Regel dem Dateiordner:
C:\Dokumente und Einstellungen\<Benutzername>\Anwendungsdaten\gnupg\

In diesem gnupg-Dateiordner befinden sich sämtliche persönlichen GnuPG-Daten, also die persönlichen Schlüssel, Vertrauensstellungen und Programmkonfigurationen.
- Systemweite GnuPG-Anwendungsdaten
in %COMMON_APPDATA%\GNU, das entspricht in der Regel dem Dateiordner:
C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\GNU\

Deinstallation von Gpg4win-1.1.3

Nach der Deinstallation von Gpg4win-1.1.3 bleiben folgende Dateiordner bzw. Registryschlüssel zurück:

- Dateinamen:
%APPDATA%\gnupg (Wird von einer Gpg4win2-Installation *weiter* verwendet.)
Wichtig: Hier sind Ihre persönlichen privaten und öffentlichen Schlüssel sowie GnuPG-Einstellungen enthalten.
- Registryschlüssel:
HKLM\Software\GNU\GnuPG (Wird von einer Gpg4win2-Installation *nicht* mehr verwendet.)
HKCU\Software\GNU\GPG4Win (Wird von einer Gpg4win2-Installation *nicht* mehr verwendet.)
HKCU\Software\GNU\GpgOL (Wird von einer Gpg4win2-Installation *weiter* verwendet.)
HKCU\Software\GPGee (Wird von einer Gpg4win2-Installation *nicht* mehr verwendet.)

F. Historie

- „GnuPP für Einsteiger“, 1. Auflage März 2002 und „GnuPP für Durchblicker“, 1. 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
Herausgeber: 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
Überarbeitung: Werner Koch, g10 Code GmbH
Herausgeber: die Gpg4win-Initiative
 - Dank der Erlaubnis des BMWi vom 14. November 2007 wurde der unveränderbare Abschnitt „Impressum“ entfernt und an die aktuelle Version angepasst.
 - Das „Gpg4win-Kompodium“ fasst „Gpg4win für Einsteiger“ und „Gpg4win für Durchblicker“ zusammen und ist in den Jahren 2009/2010 umfassend für Gpg4win2 aktualisiert und ergänzt worden.
Grundlegende Überarbeitung:
Werner Koch, g10 Code GmbH
Florian v. Samson, Bundesamt für Sicherheit in der Informationstechnik (BSI)
Emanuel Schütze, Intevation GmbH
Dr. Jan-Oliver Wagner, Intevation GmbH
 - Im Rahmen von des BSI Auftrages „Gpg4all“ im Jahre 2017 umfassend überarbeitet, ergänzt und aktualisiert worden.
Grundlegende Überarbeitung:
Jochen Saalfeld, Intevation GmbH
Emanuel Schütze, Intevation GmbH
- Das Programmpaket Gpg4win und das Gpg4win-Kompodium sind verfügbar unter:
<http://www.gpg4win.de>

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.

Index

A

ASCII armor 90
Asymmetrische Verschlüsselung 22, 24
Authentisierung 28
Authentizität 28
Automatische Installation 139

B

Beglaubigung 63
Beglaubigungsinstanzen 63
Briefgeheimnis 13
Brieftresor 19, 22, 28
Briefumschlag 11
Bundesamt für Sicherheit in der Informations-
technik 10
Bundesministerium für Wirtschaft und Techno-
logie 10

C

CA-Zertifikat 99, 106
CAcert 48
Certificate Authority (CA) 29, 47, 63
Claws Mail 138
CRLs *siehe* Sperrlisten
CryptoEx 102

D

Datei
 entschlüsseln 52
 Signatur prüfen 76
 signieren 73
 verschlüsseln 49
Dateianhänge verschlüsseln 88
Deinstallation 143
Denial of Service 83
Diagnosestufe 115

Directory Manager *siehe* DirMngr
DirMngr 99, 103, 105 f., 113

E

E-Mail
 signieren 69
 verschlüsselt archivieren 70
Echelon-System 12
Enigmail 138

F

Fernmeldegeheimnis 13
Fingerabdruck 42, 58
Freie Software 9

G

GNU FDL 146 – 152
GNU Free Documentation License *siehe* GNU
 FDL
GNU Privacy Assistant *siehe* GPA
GnuPG 9
GnuPG für Outlook *siehe* GpgOL
GnuPP 54, 145
GPA 9
GPG Explorer eXtension *siehe* GpgEX
Gpg4win 9
GpgEX 9, 72
GpgOL 9
GpgOL-Optionen 137

H

Hierarchisches Vertrauenskonzept 28
HTTP 99, 107

I

Installation 30

Integrität.....*siehe* Unverändertheit

K

Kleopatra 9, 37
KMail 138
Kontakt 138
Kryptografie 9

L

LDAP 83, 99, 107
Logdatei 110
 von DirMngr 113
 von GnuPG 114
 von GpgME 115
 von GpgOL 112
 von Kleopatra 111

M

Migration von Gpg4win 142
Modulo-Arithmetik 118

N

Netz des Vertrauens.....*siehe* Web of Trust
'Non-Public-Key'-Method*siehe* Symmetrische
 Verschlüsselung

O

OpenPGP 10, 28
 Zertifikat erstellen 38
OSCP 99
Outlook 101, 138
 Programmerweiterung 136

P

Passphrase 25
PGP 10
Pinentry 67
Postgeheimnis 13
Primzahlen 122 – 132
Problembhebungen 101
Proxy 85, 99, 107
'Public-Key'-Methode 14, 18

R

Re-Migration von GpgOL 143

Restklassen 119
RSA-Algorithmus 117 – 134

S

S/MIME 10, 28
Schlüssel
 -ID 42, 81
 öffentlicher 23
 erzeugen 36
 geheimer 23
 privater 23
Schlüsselbund 24, 89
Schlüsselkennung 42, 81
Schlüsselpaar 17, 36
Sicherheitsphilosophie 116
Signatur
 digitale 69
 prüfen mit GpgOL 54, 68
 qualifizierte elektronische 69
Signaturgesetz 63, 69
Sperrlisten 99, 103, 105, 107
Symmetrische Verschlüsselung 16, 19

T

Thunderbird 138
Trojaner 25
trustlist.txt 108 f.

U

Unverändertheit 73

V

Verfallsdatum 81
Vertrauenskette 28
Vertrauenswürdige Wurzelzertifikate . 99, 106,
 109
Viren 25

W

Würmer 25
Web of Trust 29, 62
Windows-Explorer 9
Wurzelzertifikat 99
Wurzelzertifikate 104, 106, 108 f.

X

X.509	10, 36
Zertifikat erstellen	45

Z

Zertifikat	
beglaubigen	59
Benutzerkennung	81
erstellen	36
exportieren	54, 90
Gültigkeit	81
importieren	86, 91
verbreiten	79
Zertifikatsaussteller	29
Zertifikatsdetails	81
Zertifikatskette	104
Zertifikatsserver	83
einrichten	84
OpenPGP	83
Suche nach Zertifikaten	86
X.509	83
Zertifikatsverwaltung	9, 37