F R I E D R I C H – A L E X A N D E R – U N I V E R S I T Ä T

E R L A N G E N – N Ü R N B E R G

Seminar *Konzepte von Betriebssystem-Komponenten*

Web of Trust,

Pretty Good Privacy, Gnu Privacy Guard

18. Juli 2010

Tobias Sammet

tobias.sammet@informatik.stud.uni-erlangen.de

Inhaltsverzeichnis

1 Überblick

In dieser Ausarbeitung werden zwei Programme vorgestellt, die vor allem im Bereich der E-Mail-Verschlüsselung Verwendung finden: *Pretty Good Privacy* (kurz: PGP) und *Gnu Privacy Guard* (kurz: GnuPG). Das Augenmerk liegt zunächst jedoch auf dem Vertrauensmodell, welches beiden Programmen zugrunde liegt, dem so genannten *Web of Trust*. Es erlaubt jedem Teilnehmer selbst zu entscheiden, welchen Personen er Vertrauen schenken möchte. Der Schwerpunkt in diesem Kapitel liegt hierbei auf der Funktionsweise dieses Modells.
Im nächsten Kapitel werden die wichtigsten Funktionen von PGP und GnuPG kurz vorgestellt: Verschlüsseln, Entschlüsseln, Signieren und Verifizieren. Durch eine kurze Gegenüberstellung beider Programme im Anschluss werden die wichtigsten Gemeinsamkeiten und Unterschiede herausgearbeitet.
Das letzte Kapitel umfasst schließlich einen kurzen Blick in die Gegenwart und die Zukunft.

2 Motivation

Als Einstieg in das Thema der Verschlüsselung müssen zunächst ein paar Grundbegriffe erklärt werden: Grundsätzlich unterscheidet man zwischen zwei verschiedenen Verschlüsselungstechniken: die symmetrische Verschlüsselung, welche denselben Schlüssel für Verschlüsselung und Entschlüsselung verwendet, und die asymmetrische Verschlüsselung, welche einen öffentlichen Schlüssel (*Public Key*) für die Verschlüsselung und einen privaten Schlüssel (*Private Key*) für die Entschlüsselung benutzt. Diese beiden Schlüssel sind jedoch unterschiedlich. Der öffentliche Schlüssel wird in der Regel auf Schlüsselservern (*Key Server*) hinterlegt, welche man am besten mit Telefonbüchern vergleichen kann, allerdings sind hier nicht zu jedem Name eine Telefonnummer, sondern zu jeder E-Mail-Adresse der entsprechende öffentliche Schlüssel aufgelistet. Somit hat jeder Zugriff auf die öffentlichen Schlüssel und dadurch die Möglichkeit, damit Nachrichten zu verschlüsseln, jedoch kann nur der Besitzer des entsprechenden privaten Schlüssels diese Nachrichten wieder entschlüsseln. Zusammen bilden privater und öffentlicher Schlüssel ein Schlüsselpaar.

Nun soll ein kleines Szenario aus dem Bereich der E-Mail-Verschlüsselung in dieses Thema einführen, welches auch graphisch in Abbildung 1 dargestellt wird.
Der Angreifer Eve generiert für die E-Mail-Adresse von Alice ein Schlüsselpaar, bestehend aus öffentlichem und privatem Schlüssel. Auf die E-Mail-Adresse selbst hat er jedoch keinen Zugriff, da er nicht im Besitz des dazugehörigen Passwortes ist. Nichtsdestotrotz verbreitet er den öffentlichen Schlüssel auf einigen Schlüsselservern.
Bob hat von der Möglichkeit der E-Mail-Verschlüsselung gehört und möchte dies gerne selbst testen. Seine Suche nach einer ihm bekannten E-Mail-Adresse auf einem der Schlüsselserver ist erfolgreich: Er erhält als Ergebnis den entsprechenden öffentlichen Schlüssel zu der E-Mail-Adresse von Alice. Damit verschlüsselt Bob seine Nachricht und verschickt sie im Anschluss an Alice. Allerdings weiß er nicht, dass sie diese E-Mail niemals entschlüsseln kann, da sie nicht im Besitz des passenden privaten Schlüssels ist, denn dieser befindet sich bei dem Angreifer. Eve fängt die E-Mail unterwegs ab (so genannter *Man-in-the-Middle*-Angriff), entschlüsselt diese mit dem passenden privaten Schlüssel und nutzt die geheimen Informationen für sich, während das Ganze für Alice hingegen nur nach zufällig aneinander gereihten Buchstaben und Zahlen aussieht.

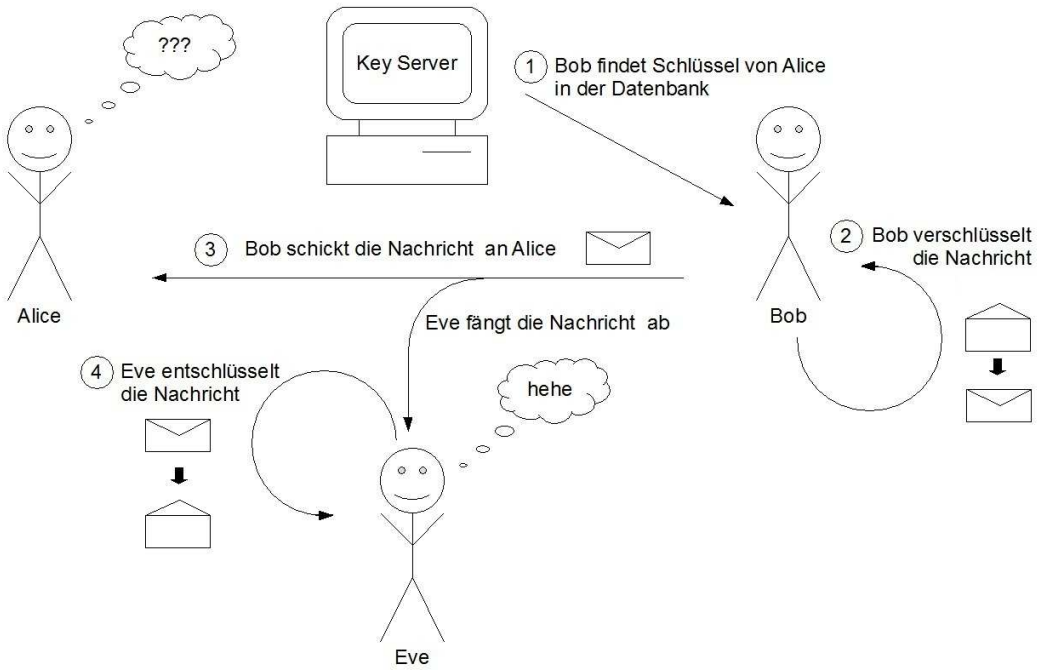

Abbildung 1: *Man-in-the-Middle*-Angriff

3 Das *Web of Trust*

Um unter anderem einem solchen eben vorgestellten Szenario vorzubeugen, wurde ein dezentrales Vertrauensmodell entwickelt: das so genannte *Web of Trust* [Sch05]. Hierbei gibt es für jeden öffentlichen Schlüssel ein digitales Zertifikat, das mit einem amtlichen Dokument vergleichbar ist. Es enthält unter anderem den öffentlichen Schlüssel, persönliche Daten des Inhabers (z.B. Name und E-Mail-Adresse) und Verwaltungsdaten (z.B. Eigenschaften des Schlüssels, auf die später noch eingegangen wird) [Tob02]. Dieses Zertifikat kann im *Web of Trust* von jedem Teilnehmer eingesehen, außerdem überprüft und mit seinem eigenen privaten Schlüssel signiert werden. Dieser Vorgang der Signierung wiederum ist vergleichbar mit dem Setzen der eigenen Unterschrift unter ein Dokument [Sch05].
Eben diese Überprüfung der Authentizität des Zertifikats eines öffentlichen Schlüssels samt anschließender Signierung wird im Folgenden genauer betrachtet. Außerdem bietet das *Web of Trust* die Möglichkeit, dem Schlüsselbesitzer persönliches Vertrauen auszusprechen, was im Anschluss daran erklärt wird.

3.1 Funktionsweise

3.1.1 Überprüfung der Authentizität

Ein jeder öffentliche Schlüssel besitzt Eigenschaften, wie beispielsweise die Schlüssel-ID, den digitalen Fingerabdruck, die Stärke der Verschlüsselung, den verwendeten Algorithmus oder das Erzeugungsdatum, wie in Abbildung 2 zu erkennen ist. Dabei handelt es sich bei dem digitalen Fingerabdruck, welcher – ähnlich wie bei uns Menschen – eindeutig identifizierbar ist, um die Prüfsumme (auch *Hash* genannt) des öffentlichen Schlüssels und der Signatur [Tob02]. Nun gilt es jedoch festzustellen, ob der Schlüsselinhaber auch wirklich diejenige Person ist, für die sie sich ausgibt.

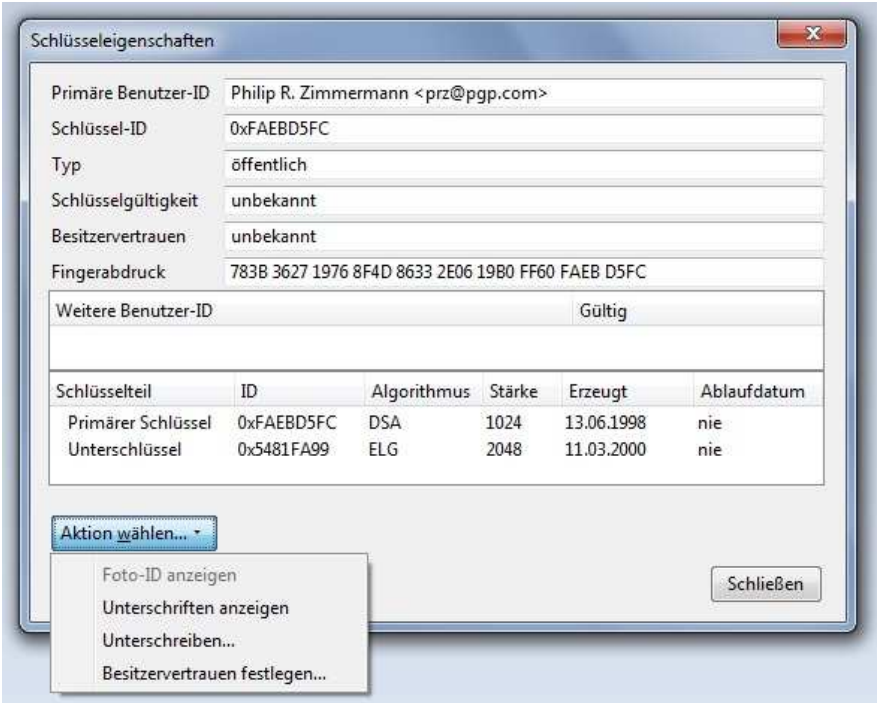

Abbildung 2: Eigenschaften eines (hier: öffentlichen) Schlüssels

Jeder Teilnehmer des *Web of Trust* hat die Möglichkeit, die Authentizität und Integrität des digitalen Zertifikats zu einem fremden öffentlichen Schlüssel zu überprüfen [Jan06]. In diesem Fall kontaktiert er den Schlüsselbesitzer in der Regel persönlich, um Eigenschaften des zu überprüfenden Schlüssels abzufragen und abzugleichen [KAI_A6]. Hierbei sollte der digitale Fingerabdruck wegen seiner identifizierenden Eigenschaft bei einer Überprüfung auf jeden Fall abgeglichen werden, zusätzlich aber noch weitere Attribute, denn solch ein Fingerabdruck kann durch einen Gestaltungsfehler des *Web of Trust* nachgebaut werden, wodurch sich jedoch dann wiederum die Schlüsselstärke ändern würde [AE]. Somit kann ein auf diese Art gefälschter Schlüssel erkannt werden. Durch den Abgleich der Eigenschaften kann festgestellt werden, ob es sich wirklich um die Person handelt, für die sich der Schlüsselinhaber ausgibt [LPKA].
Sicherlich wäre Bob im einführenden Beispiel auf diese Weise aufgefallen, dass es sich bei dem öffentlichen Schlüssel gar nicht um den von Alice handelt, ja vielmehr, dass Alice die Möglichkeit der E-Mail-Verschlüsselung gar nicht wahrnimmt.

3.1.2 Signierung des digitalen Zertifikats

Doch was nützt jede Überprüfung, wenn sie nicht für andere Teilnehmer ersichtlich ist? So bietet das *Web of Trust* außerdem die Möglichkeit, eine von insgesamt drei Optionen auszuwählen [AAR] – je nach Genauigkeit der Überprüfung:

- Keine Überprüfung (*undefined*)
- Einfache Überprüfung (*marginal*)
- Genaue Überprüfung (*complete*)

Dadurch wird das oben angesprochene Zertifikat entsprechend mit dem eigenen privaten Schlüssel (stellvertretend für den eigenen Namen) unterzeichnet. Dieser Prozess wird häufig Schlüsselsignierung (*Key Signing*) genannt und ist mit dem Unterschreiben eines Dokuments vergleichbar. Die Signierung des Zertifikats ist in Abbildung 3 zu erkennen.

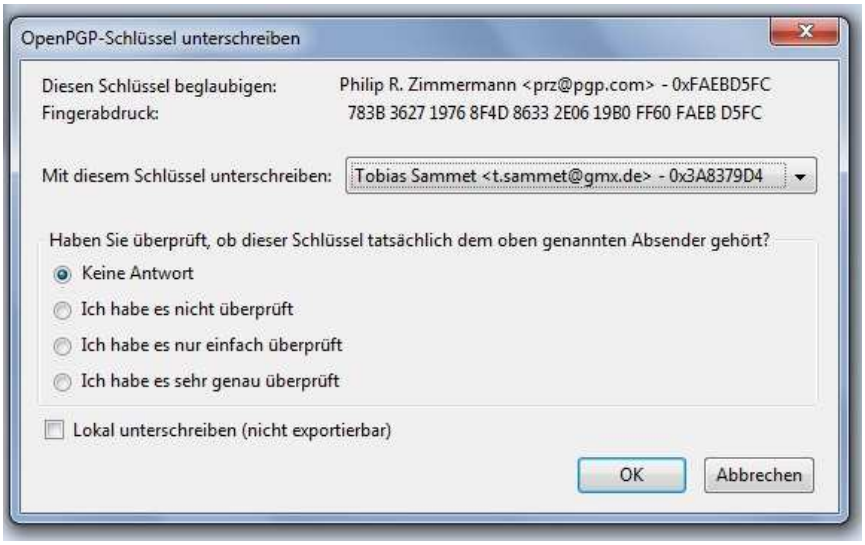

Abbildung 3: Signierung des Zertifikats des zugehörigen öffentlichen Schlüssels

Ein jedes Zertifikat eines öffentlichen Schlüssels, das man selbst sehr genau überprüft und signiert hat, ist nun gültig (*complete*), d.h. mit anderen Worten: Man glaubt an die Echtheit dieses Schlüssels bzw. daran, dass die Person hinter dem Schlüssel diejenige ist, für die sie sich ausgibt. Ein Zertifikat eines öffentlichen Schlüssels hingegen ist teilweise gültig (*marginal*), sofern man selbst es einfach überprüft und signiert hat. Diese Unterschriften sind von anderen Teilnehmern des *Web of Trust* einsehbar, sofern sie auf die Schlüsselserver exportiert werden [Jan06], was jedoch standardmäßig eingestellt ist. Sollte dies nicht erwünscht sein, so hat man die Möglichkeit nur lokal zu unterschreiben (siehe Abbildung 3), wodurch keine Exportierung der Signatur erfolgt.
An dieser Stelle möchte ich erwähnen, dass es so genannte *Key Signing Parties* gibt. Dies sind Veranstaltungen, um sich gegenseitig die Authentizität seines öffentlichen Schlüssels gegen Vorlage des eigenen Personalausweises bestätigen zu lassen [AE].

3.1.3 Aussprechen eines persönlichen Vertrauens

Nun hat die Bestätigung der Identität noch lange nichts mit dem persönlichen Vertrauen in diese Person zu tun. Als Beispiel: Fast jeder Personalausweis ist korrekt, jedoch wird über den Menschen selbst nicht viel ausgesagt [Tob02]. Nur, weil man sich nun durch diesen Ausweis die Identität einer Person bestätigen lassen kann, heißt das noch lange nicht, dass man ihr auch persönlich vertraut.
Das *Web of Trust* trägt diesem Umstand dadurch Rechnung, dass jedem Zertifikat eines öffentlichen Schlüssels eine Angabe, genauer ein persönlicher Vertrauenswert, darüber zugeordnet wird [GPGH_3], inwieweit man persönlich dem Eigentümer des Zertifikats vertrauen kann. Dieses Konzept wird als *Trust Model* [AAR] bzw. O*wnertrust* [GPGH_3] bezeichnet. Hierbei stehen folgende vier Vertrauensstufen zur Auswahl [GPGH_A], welche auch in Abbildung 4 dargestellt werden:

- Kein Vertrauen (*untrusted*)
- Teilweises Vertrauen (*marginally trusted*)
- Volles Vertrauen (*fully trusted*)
- Absolutes Vertrauen (*ultimately trusted*)

Abbildung 4: Vertrauen in den Schlüsselbesitzer

Diese letzte Option sollte nur für den Schlüssel sein, zu dem man auch den entsprechenden privaten Schlüssel besitzt [PF]. Deshalb wird diese Option in den nachfolgenden Fällen außer Acht gelassen.
Zur Erinnerung: Bisher ist es der Fall, dass ein Zertifikat nur dann teilweise gültig (*marginal*) bzw. gültig (*complete*) ist, sofern man es persönlich signiert hat. Dies ist jedoch sehr restriktiv. Daher sieht das Web of Trust die Möglichkeit vor, das persönliche Vertrauen wirken zu lassen. Durch das Aussprechen von persönlichem Vertrauen in den Eigentümer eines Zertifikats verändert sich die Gültigkeit der Zertifikate (noch fremder) öffentlicher Schlüssel. Wenn man einer Person vertraut, setzt man voraus, dass sie nur diejenigen Zertifikate signiert hat, die von ihr zuvor überprüft worden sind. Aufgrund der Transitivität wirkt sich dies auf das gesamte System aus: Das Zertifikat eines Schlüssel ist genau dann gültig, sofern er von einem persönlich oder von einem Schlüssel vollen Vertrauens oder von drei Schlüsseln teilweisen Vertrauens unterzeichnet wurde und wenn zusätzlich der Pfad der unterschriebenen Schlüssel aus maximal fünf Schritten besteht [GPGH_3]. Die maximale Pfadlänge kann nach Wunsch geändert werden, ebenso wie die Anzahl der benötigten Schlüssel teilweisen Vertrauens, um einem weiteren Schlüssel Gültigkeit zu verleihen [AE].
Im Nachfolgenden sollen die Auswirkungen von persönlichem Vertrauen anhand von vier Beispielen gezeigt werden.

3.1.4 Das *Web of Trust* anhand eines Beispiels

Zur Vereinfachung werden in den folgenden Beispielen nur die Standardfälle und jeweils nur ein kleiner Ausschnitt aus einem Vertrauensnetz abgedeckt. Die übrigen Fälle sind daraus ableitbar.
Bob hat die Authentizität von Carols öffentlichem Schlüssel durch Abgleichen einiger Schlüsseleigenschaften einfach überprüft, d.h. das dazugehörige Zertifikat ist für ihn teilweise gültig (*marginal*), während der öffentliche Schlüssel von Dave genau von ihm überprüft wurde, d.h. dass dieses Zertifikat für ihn gültig ist (*complete*). Dies stellt die Ausgangssituation der nächsten drei Grafiken dar.
Im ersten Beispiel (siehe Abbildung 5) räumt Alice Bob keinerlei Vertrauen ein (*untrusted*). Dadurch ändert sich für sie in dieser Situation nichts. Alice traut Bob nicht und damit traut sie auch nicht der Authentizität der von Bob unterzeichneten Zertifikate.

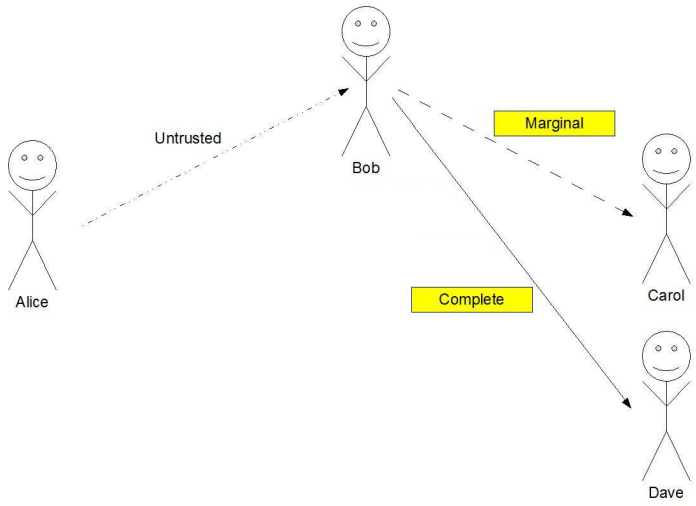

Abbildung 5: Wirken der Transitivität bei keinerlei Vertrauen

Im zweiten Beispiel wird die Situation leicht abgewandelt: Alice vertraut Bob teilweise (*marginally trusted*), d.h. sie vertraut teilweise darauf, dass Bob die Zertifikate anderer öffentlicher Schlüssel prüft, bevor er sie signiert. Durch die schon erwähnte Transitivität ist für Alice deshalb auch das Zertifikat des öffentlichen Schlüssels von Dave teilweise gültig. Dieser Fall wird in Abbildung 6 abgedeckt.

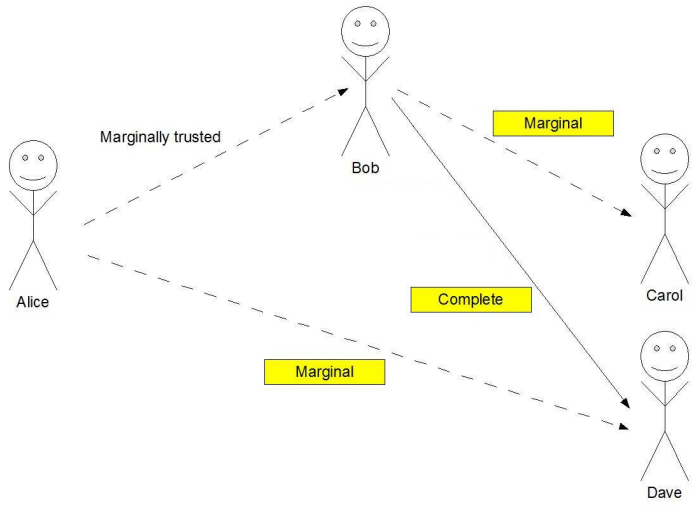

Abbildung 6: Wirken der Transitivität bei teilweisem Vertrauen

Im dritten Beispiel (siehe Abbildung 7) schenkt Alice Bob volles Vertrauen (*fully trusted*), d.h. alle Zertifikate von öffentlichen Schlüsseln, die für Bob teilweise gültig sind, sind nun auch für Alice teilweise gültig. Ebenso sind für sie alle Zertifikate öffentlicher Schlüssel gültig, die auch für Bob gültig sind.

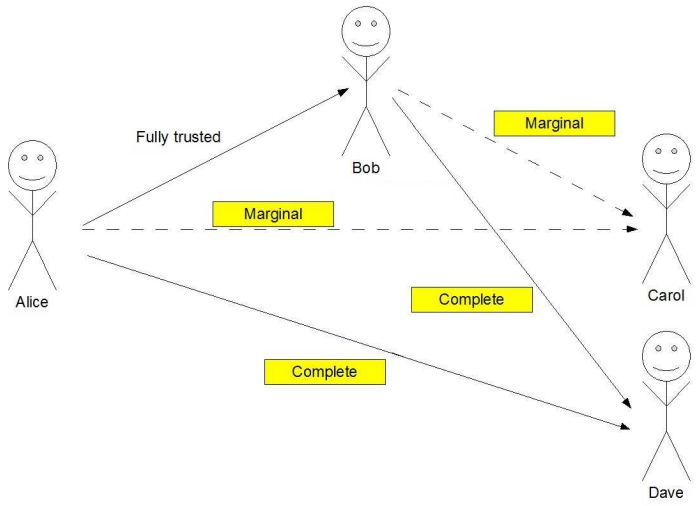

Abbildung 7: Wirken der Transitivität bei vollem Vertrauen

Im letzten der vier Beispiele hat sich die Situation leicht abgewandelt, um eine Besonderheit aufzeigen zu können: Alice hat teilweises Vertrauen *(marginally trusted*) in drei Personen, nämlich Bob, Carol und Dave, die wiederum Eddas öffentlichen Schlüssel sehr genau überprüft und anschließend signiert haben. Dadurch ist für die drei das Zertifikat für Eddas öffentlichen Schlüssel gültig. Nun müsste laut obigem Beispiel für Alice das Zertifikat für den öffentlichen Schlüssel von Edda teilweise gültig sein. Diese Situation hier ist jedoch ein Spezialfall: standardmäßig wird das Zertifikat eines Schlüssel im *Web of Trust* gültig, sofern es für drei Personen teilweisen Vertrauens gültig ist – wie auch in Abbildung 8 zu erkennen ist [GPGH_A].

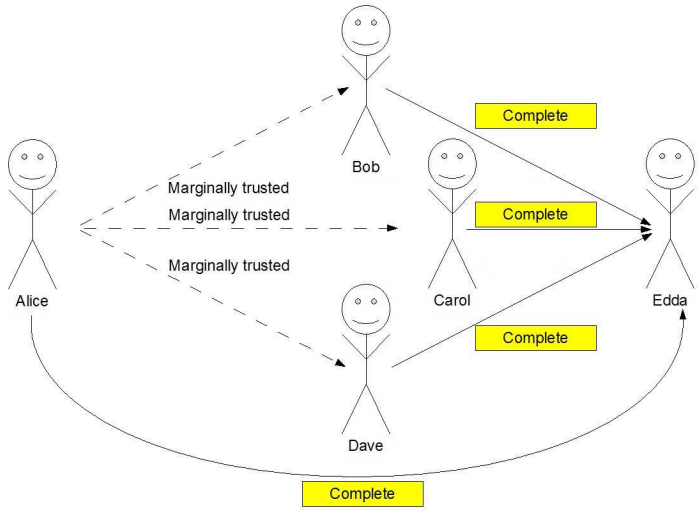

Abbildung 8: Wirken der Transitivität bei dreifachem teilweisem Vertrauen

Die vier behandelten Beispiele zeigten jeweils die Standardfälle im *Web of Trust*, wie sich Vertrauen in andere Personen auswirkt, jedoch können diese Einstellungen – wie schon erwähnt – modifiziert werden. So kann unter anderem die für das letzte Beispiel erforderliche Anzahl an Personen mit teilweisem Vertrauen nach Belieben verändert werden, um dem Zertifikat eines öffentlichen Schlüssels Gültigkeit zu verleihen. Beispielsweise benötigt man dann fünf statt nur drei Personen [AE].

3.2 Fazit

Eine Stärke des *Web of Trust* ist seine dezentrale Auslegung, da jeder Teilnehmer für sich entscheiden kann, wem er vertrauen möchte. Dies hat zur Folge, dass gefälschte Schlüssel oder nicht vertrauenswürdige Personen im Idealfall dadurch erkannt werden, dass ihnen niemand vertraut. Wenn Schlüssel nur signiert werden, nachdem sie (ausführlich) geprüft worden sind, bewirkt dies ein hohes Maß an Sicherheit [Jan06]. Die Dezentralität birgt zugleich aber auch Gefahr: Jeder Teilnehmer kann dadurch in diesem Netz gegenseitigen Vertrauens die Tür für Angreifer öffnen. [Sch05]
Abschließend bleibt noch festzuhalten: Das *Web of Trust* bildet die Grundlage für die Software *Pretty Good Privacy* (PGP) bzw. *Gnu Privacy Guard* (GnuPG) und wird meist in Zusammenhang mit eben jenen Produkten betrachtet.

4 Pretty Good Privacy

Als 1991 dem US-Senat eine Gesetzesvorlage vorgestellt wurde, mit deren Verabschiedung jede Verschlüsselungssoftware eine Hintertür für den staatlichen Zugriff enthalten müsse, wurde Philip R. Zimmermann dazu veranlasst, eine sichere Software, nämlich *Pretty Good Privacy*, zu schreiben [Sch05].
Pretty Good Privacy ist ein so genanntes hybrides Verschlüsselungsverfahren, da es sowohl symmetrische als auch asymmetrische Verschlüsselung miteinander vereint und somit die Vorteile beider Techniken genießt. Wie dies im Einzelnen funktioniert, soll im Folgenden dargestellt werden.

4.1 Funktionsweise

4.1.1 Verschlüsseln

Für jede Verschlüsselung wird ein geheimer zufälliger Sitzungsschlüssel (so genannter *Session Key*) generiert, mit dem der Klartext symmetrisch verschlüsselt wird. Nur dieser zufällige Schlüssel wird anschließend mit dem öffentlichen Schlüssel des Empfängers asymmetrisch verschlüsselt und mitgeschickt [Tob02]. Das Ganze hat folgenden Hintergrund: Die asymmetrische Verschlüsselung von Daten und die weiteren Berechnungen benötigen viel mehr Zeit als eine symmetrische Blockchiffrierung mit einem symmetrischen Schlüssel – nämlich ungefähr Faktor tausend [Tob02]. Der Vorteil der asymmetrischen Verschlüsselung jedoch ist eine hohes Maß an Sicherheit, da der Schlüssel nicht wie bei der symmetrischen Verschlüsselung zum Empfänger transportiert werden muss [Tob02]. Die Funktionsweise der Verschlüsselung wird in Abbildung 9 graphisch dargestellt.

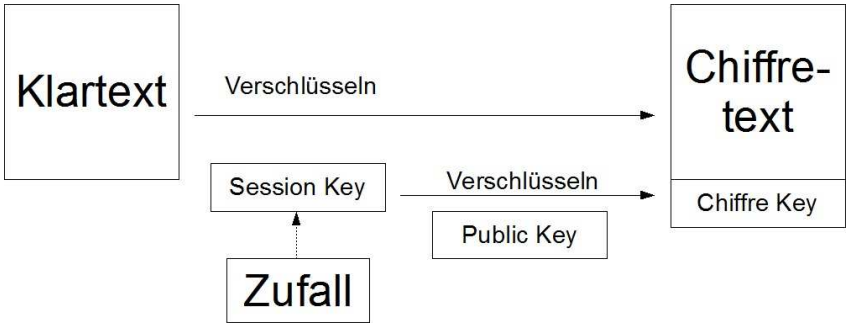

Abbildung 9: Verschlüsseln [Tob02]

4.1.2 Entschlüsseln

Bei der Entschlüsselung einer Nachricht ist genau der umgekehrte Weg der Verschlüsselung zu gehen: Der entsprechende private Schlüssel des Empfängers entschlüsselt den erhaltenen

Sitzungsschlüssel, mit dem der Chiffretext wieder zum Klartext symmetrisch entschlüsselt wird [Tob02]. Nach der Entschlüsselung kann der Empfänger der Nachricht nun genau das lesen, was ihm der Absender geschickt hat. Dieser Vorgang ist in Abbildung 10 auch graphisch dargestellt.
Hier sei noch erwähnt, dass bei Verwendung des privaten Schlüssels eine eigens gewählte Passphrase – z.B. ein komplexes Passwort, entweder ein langer Text oder eine Zeichenfolge mit vielen Sonderzeichen [Sch05] – als zusätzliche Sicherheitshürde eingegeben werden muss. Dadurch soll Personen mit physikalischem Zugriff auf denselben PC (z.B. Mehrbenutzersysteme) der Missbrauch des privaten Schlüssels verwehrt bleiben.

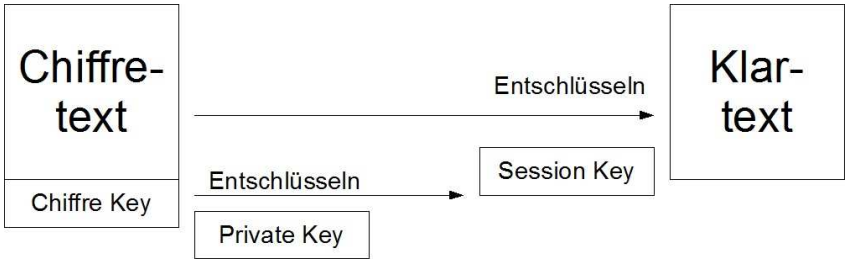

Abbildung 10: Entschlüsseln [Tob02]

4.1.3 Signieren

Wie am Anfang des Kapitels des *Das Web of Trust* schon erläutert: Eine digitale Signatur erfüllt denselben Zweck wie eine handschriftliche Unterschrift [PGPi]. Daher interessiert die digitale Unterschrift oder Signatur aufgrund zweier Gesichtspunkte besonders, da sie – im Gegensatz zu einer handschriftlichen Unterschrift – beinahe unmöglich gefälscht werden kann und da sie außerdem für den Inhalt der Nachricht Schutz vor Änderungen bietet [PGPi]. Sie verschafft dem Empfänger also die Möglichkeit, zu überprüfen, ob eine Nachricht zum einen wirklich vom Absender stammt und zum anderen, ob sie unterwegs eventuell manipuliert wurde [GPGH_2].
Bei der Erzeugung von digitalen Unterschriften benutzt man das Schlüsselpaar, bestehend aus öffentlichem und privatem Schlüssel, anders als bei der Ver- und Entschlüsselung [GPGH_2]. Zunächst wird vom Klartext mit Hilfe eines Hash-Algorithmus die (Text-)Prüfsumme (auch *Hashwert* oder *Message Digest* genannt) gebildet, welcher anschließend mit dem privaten Schlüssel des Absenders verschlüsselt wird, weshalb auch hier wieder die Passphrase eingegeben werden muss. (Siehe Abbildung 11)

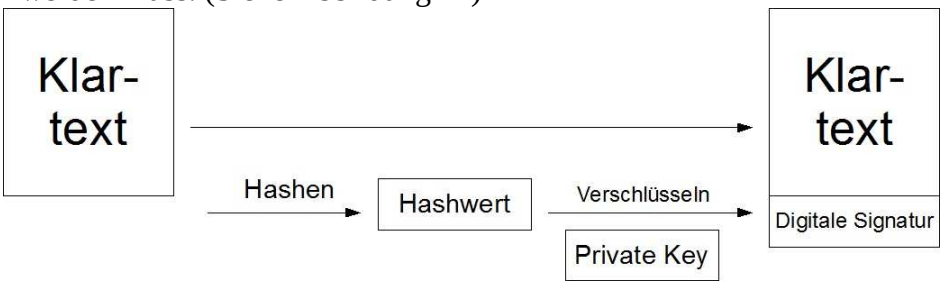

Abbildung 11: Signieren [Tob02]

Da die Prüfsumme in der Regel klein ist, kann sie ohne große Zeitverluste asymmetrisch verschlüsselt werden [Tob02]. Sobald der Klartext digital signiert wurde, darf er nicht mehr verändert werden, denn bereits eine winzige Modifikation des Textes (z.B. ein Leerzeichen oder ein Zeilenumbruch) reichen aus, um die Signatur ungültig zu machen [Jan06]. Dadurch würde sich nämlich die Prüfsumme der Nachricht drastisch verändern und der anschließende Vergleich fehlschlagen [Tob02]. Dies wird in Abbildung 12 und Abbildung 13 demonstriert.

Hier wurde derselbe Text einmal mit einem Punkt am Ende und einmal ohne diesen Punkt am Ende signiert.

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

Ein sinnloser Text
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.10 (MingW32)
Comment: Using GnuPG with Mozilla - http://enigmail.mozdev.org/

iQEcBAEBAgAGBQJMPzhmAAoJEBWCOPs6g3nU2MkH/1Z8na2bx04y0zzA/Gh7mtmv
mJO6YtMC1rB22OqSYY5sP8CylKjj4Y2asflalgj1D/pIiE9dmFSgXdxAxz0CyDRs
5FTUnYLFWASRJ71y/fUTPN4ApI2lOcj4iJlbMa911xTg6qs9sMXw2GsTHgilWjx4
gchujZwGsYRQIQ7a4jxMZdVAb/lV09NeRxLTrJE90q5VfA8C+CzRzzvbwpc00heG
HErbrs/jxwor2S3xO/RivAlA/ZU2GOPOfZzUQgoW0YCXLbZhATn61B7tusBSiLRa
rOL+1BVTXE+sAhw8KKhW5tNrHUMGC8lo/lti7MGyQTOJ82txfVgS6b6CcOxZHj0=
=5RFL
-----END PGP SIGNATURE-----
```

Abbildung 12: Prüfsumme von "Ein sinnloser Text" (ohne Punkt)

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

Ein sinnloser Text.
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.10 (MingW32)
Comment: Using GnuPG with Mozilla - http://enigmail.mozdev.org/

iQEcBAEBAgAGBQJMPw8FAAoJEBWCOPs6g3nUHggIAKNxXI80zgaqwvgg8k9Gg0ZP
ttdFOfyyEJZ4BLML//ZKuYJdFsNZ4LyT0fd4IhcJcc66zIWisqBwGY1f3HueUxh8
wu9sL4evHJgOsK8FqEqxfB/Gb6vr2T4H581t9B2hswLhdFGJGmZ3JQyB3XzDmd0n
R3SsGb24aJF7i0mPTgoNaW+63LRV8OS7EA1mV7ytLq+YVLaviazIiv2zn7vMIDBn
SrpIR3H89lj9GLGTxmV6c8otfKxJUwy6nksKkhVLXxHEnOU+KlbYXZL17nF6wyNg
XHV7sYTQ+ugHL97VmKlrEnkEpE6XEt/oBywrYd0bnJOmWXXztyeDPsxiFQPW3bc=
=KaFK
-----END PGP SIGNATURE-----
```

Abbildung 13: Prüfsumme von "Ein sinnloser Text." (mit Punkt)

4.1.4 Verifizieren

Unter Verifikation versteht man die Überprüfung der Signatur. Dazu wird zunächst die verschlüsselte Prüfsumme mit dem öffentlichen Schlüssel des Absenders entschlüsselt. Anschließend wird vom übertragenen Klartext mit demselben Verfahren erneut die Prüfsumme ermittelt [Tob02], die im Anschluss noch mit der mitgelieferten Prüfsumme verglichen werden muss. Dieser Vorgang wurde in Abbildung 14 graphisch dargestellt. Sind sie gleich, so ist alles in Ordnung, stimmen sie jedoch nicht überein, weiß der Empfänger, dass der Text unterwegs manipuliert wurde [Tob02].

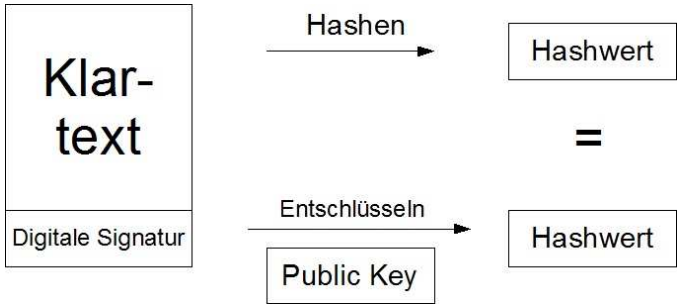

Abbildung 14: Verifizieren [Tob02]

4.2 Gemeinsamkeiten mit GNU Privacy Guard

Sowohl *GNU Privacy Guard* (GnuPG) als auch *Pretty Good Privacy* (PGP) sind weit verbreitete Programme, mit denen man Nachrichten und Dateien ver- und entschlüsseln sowie digital signieren kann. Beide beruhen auf dem OpenPGP-Standard [Zoc02], welcher die Struktur der PGP-Nachrichten und auch die Prozeduren zu ihrer Generierung [Sch05] beschreibt. Und gerade weil beide Softwareprodukte auf demselben Standard basieren – anfangs auf der vom Internet Standardisierungsgremium *IETF* (*Internet Engineering Task Force*) definierten Spezifikation *RFC 2440*, November 1997 [RFC2440], mittlerweile jedoch auf der aktuelleren Spezifikation *RFC 4880*, November 2007 [RFC4880] –, unterscheiden sie sich in ihrer Funktionsweise im Grunde nur unwesentlich. Unter anderem deshalb sind die verschiedenen Versionen von PGP mit denen von GnuPG kompatibel [Jan06].

4.3 Unterschiede zu GNU Privacy Guard

Trotz des gemeinsamen Standards unterscheiden sich beide Programme in gewissen Punkten. Ein Unterschied findet sich in der Offenlegung des Quellcodes. Während PGP mittlerweile *Closed Source* ist, legt GnuPG den Quellcode völlig offen, denn das schafft Transparenz. Die Tatsache, dass der Quellcode von jedermann auf Fehler überprüft werden kann, wird durch die Vergangenheit bestätigt (z.B. Klima-Rosa Angriff [Jan06]). Ein weiterer Vorteil für die Nutzer von *Open Source* Software: Es ist für jeden kostenfrei, womit auch der nächste Unterschied angesprochen wurde: Während PGP seit Version 9.0 kommerziell vertrieben wird [Jan06], ist GnuPG frei erhältlich (so genannte *Freeware*) [GPG].
Aufgrund dessen verwendet GnuPG nur patentfreie, PGP hingegen zusätzlich auch patentierte Algorithmen. Dies ist aber bei den Algorithmen für symmetrische Verschlüsselung nur zum Teil ersichtlich: Während beide Programme sowohl *AES* (*Advanced Encryption Standard*, der eigentliche Algorithmus ist unter *Rijndael* bekannt), *BLOWFISH*, seine Weiterentwicklung *TWOFISH*, *CAST* und *3DES* (auch *Triple-DES* genannt) verwenden, findet der patentierte Algorithmus *IDEA* nur bei PGP Verwendung [PGPFAQ] [BK]. Auch bei den Algorithmen für asymmetrische Verschlüsselung gibt es Unterschiede: PGP bietet *RSA* und den *Diffie-Hellman*-Algorithmus zur Auswahl an, während GnuPG ebenso auf *RSA* und den *ElGamal*-Algorithmus zurückgreift, wobei der letztgenannte aus der Diffie-Hellman Verschlüsselung hervorgeht [KAI_2]. Somit sind die Unterschiede in dieser Hinsicht auch nur marginal. Als Hash-Algorithmen werden bei beiden Programmen sowohl *MD5* als auch *SHA1* und deren Varianten angeboten. Die in meiner Version von GnuPG verwendeten Algorithmen sind in Abbildung 15 zu sehen.

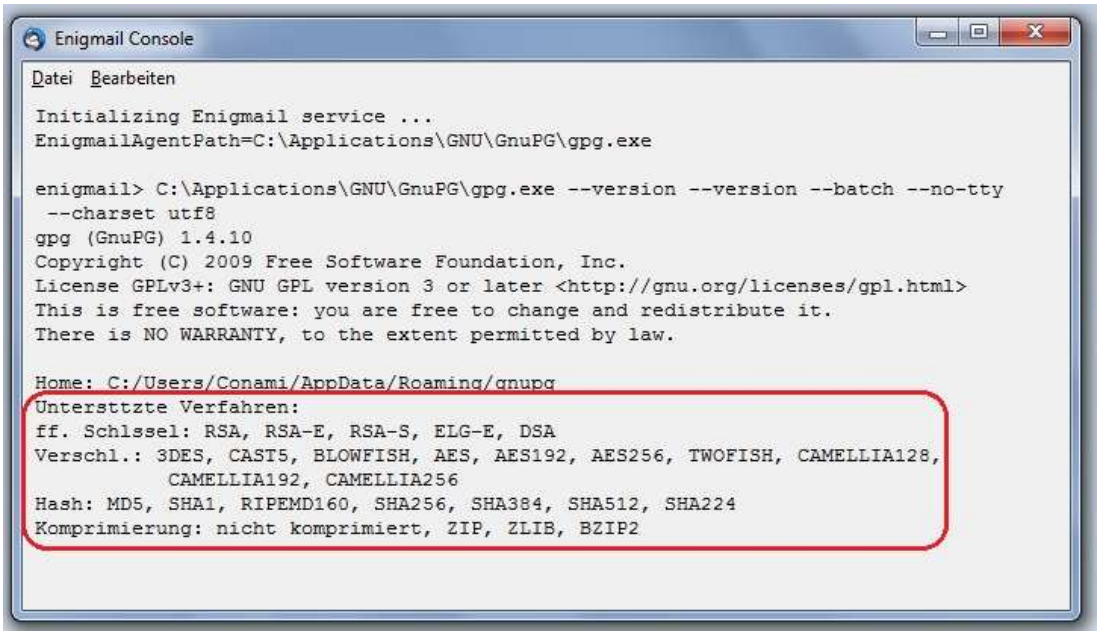

Abbildung 15: Verwendete Algorithmen bei GnuPG [BK]

Insgesamt betrachtet bieten *Pretty Good Privacy* wie auch *Gnu Privacy Guard* im Bereich der E-Mail-Verschlüsselung dieselben Funktionen an, einmal als kommerzielles Softwarepaket, einmal als Freeware.

4.4 Fazit

„Abhören an sich ist wahrlich nichts Neues – doch angesichts des Internets gewinnt es eine bisher unbekannte Dimension und Qualität (…)“ [Tob02]. Grundsätzlich ist die Möglichkeit der E-Mail-Verschlüsselung und -Signierung mittels *Pretty Good Privacy* oder *Gnu Privacy Guard* jedem zu empfehlen, um die eigenen E-Mails dadurch vor ungewolltem Abhören bzw. vor Manipulation zu schützen. Sofern bei dem Signieren von fremden Schlüsseln und anschließendem Zuweisen eines Vertrauensgrades genauso gewissenhaft vorgegangen wird wie beispielsweise im Umgang mit seinen Kontodaten, lässt sich das im einführenden Beispiel genannte Szenario problemlos vermeiden. Gerade heute ist das Verschlüsseln so sicher wie noch nie zuvor, da sich die Schlüssellängen immer schneller verdoppeln lassen – grob vereinfacht – heißt das für Kryptoanalytiker, dass sich ein doppelt so langer Schlüssel nicht ohne Weiteres knacken lässt, denn die Arbeit hierfür verdoppelt sich nicht nur, sondern wächst exponentiell [Tob02]. Das heißt allerdings nicht, dass es niemals möglich sein wird, diese Nachrichten zu entschlüsseln. Da die Sicherheit auf mathematischen Verschlüsselungsverfahren beruht, wird dies in wenigen Jahren mit entsprechend schnellerer Hardware vielleicht schon möglich sein [FAU_QL]. Bisher kann keines dieser Verfahren – trotz vielfältiger, aufwändiger und oft geheimer Entwicklung in Militär und Forschung – in der praktischen Anwendung absolute Sicherheit garantieren [FAU_QL]. Einen Ausweg könnte jedoch die Quantenphysik bieten, auf der die Quantencomputer aufbauen. Hierbei beruht die Sicherheit nicht auf mathematischen Verfahren, sondern auf den Grundgesetzen der Quantenphysik [FAU_QL]. Bereits heute gibt es erste Quantenkryptographiesysteme (z.B. in der Schweiz), um wichtige Informationen zu übertragen [FAU_QL].

5 Quellenverzeichnis

5.1 Literatur

[Jan06] „*Sicherheit im Internet*“, Krzysztof Janowicz, 2. Auflage, 2006, O‘Reilly Verlag, Köln

[Sch05] „*Sicherheit und Kryptographie im Internet*“, Jörg Schwenk, 2. Auflage, 2005, Friedr. Vieweg & Sohn Verlag/GWV Fachverlage GmbH, Wiesbaden

[Tob02] „*Verschlüsseln & Signieren*“, Mick Tobor , 2002, Markt-und-Technik-Verlag, München

[Zoc02] „*Internet Sicherheit*“, Michaela Zocholl, 2002, Microsoft Press Deutschland, Unterschleißheim

5.2 Internet

[AAR] Alfarez Abdul-Rahman, “*The PGP Trust Model*” (Stand: 13.07.2010)
http://www.wim.uni-koeln.de/uploads/media/The_PGP_Trust_Model.pdf

[AE] Arnoud Engelfriet, “*Die Comp.security.pgp FAQ*” – eine FAQ zu PGP
http://www.iks-jena.de/mitarb/lutz/security/pgpfaq.html (Stand: 13.07.2010)

[BK] Brendan Kidwell,
„*A Practical Introduction to Gnu Privacy Guard in Windows*”
http://www.glump.net/howto/gpg_intro (Stand: 13.07.2010)

[FAU_QL] FAU Erlangen-Nürnberg, Quantenkryptographie (Stand: 13.07.2010)
http://www.didaktik.physik.uni-erlangen.de/quantumlab/index.html?/quantumlab/Kryptographie/index.html

[GPG] Gnu Privacy Guard, Offizielle Homepage
http://www.gnupg.org/ (Stand: 13.07.2010)

[GPGH] Gnu Privacy Guard, Handbuch zum Schutz der Privatsphäre, Kapitelübersicht
http://www.gnupg.org/gph/de/manual/ (Stand: 13.07.2010)

[GPGH_2] Gnu Privacy Guard, Handbuch, Kapitel 2 – Digitale Signaturen
http://www.gnupg.org/gph/de/manual/x275.html (Stand: 13.07.2010)

[GPGH_3] Gnu Privacy Guard, Handbuch, Kapitel 3 – Authentifizieren anderer Schlüssel
http://www.gnupg.org/gph/de/manual/x420.html (Stand: 13.07.2010)

[GPGH_A] Gnu Privacy Guard, Handbuch, Anhang A
http://www.gnupg.org/gph/de/manual/r1023.html (Stand: 13.07.2010)

[KAI] Kai Billen, Anleitung für GnuPG, Kapitelübersicht
http://hp.kairaven.de/pgp/gpg/index.html (Stand: 13.07.2010)

[KAI_2] Kai Billen, Anleitung, Kapitel 2
http://hp.kairaven.de/pgp/gpg/gpg2.html (Stand: 13.07.2010)
[KAI_A6] Kai Billen, Anleitung, Anhang 6
http://hp.kairaven.de/pgp/gpg/weboftrust.html (Stand: 13.07.2010)
[LPKA] Lars Packschies, Kristina Auerswald,
GnuPG, PGP und sichere Mails unter Unix, Linux und Windows
http://www.uni-koeln.de/rrzk/sicherheit/pgp/gpg_allg.html (Stand: 13.07.2010)
[PGPFAQ] Wouter Slegers, *Frequently Asked Questions* (FAQ) zu PGP
http://www.pgp.net/pgpnet/pgp-faq/ (Stand: 13.07.2010)
[PGPi] PGP International Homepage, Dokumentation "*How PGP works*"
http://www.pgpi.org/doc/pgpintro/ (Stand: 13.07.2010)
[RFC2440] Internet Engineering Task Force, alte Spezifikation des OpenPGP-Standards
http://www.ietf.org/rfc/rfc2440.txt (Stand: 13.07.2010)
[RFC4880] Internet Engineering Task Force, neue Spezifikation des OpenPGP-Standards
http://www.ietf.org/rfc/rfc4880.txt (Stand: 13.07.2010)
[PF] Patrick Feisthammel, "*Das Web of Trust (Vertrauensnetz)*"
http://www.rubin.ch/pgp/weboftrust.de.html (Stand: 13.07.2010)

6 Abbildungsverzeichnis